

ИЗДАТЕЛЬСТВО
РАНОК

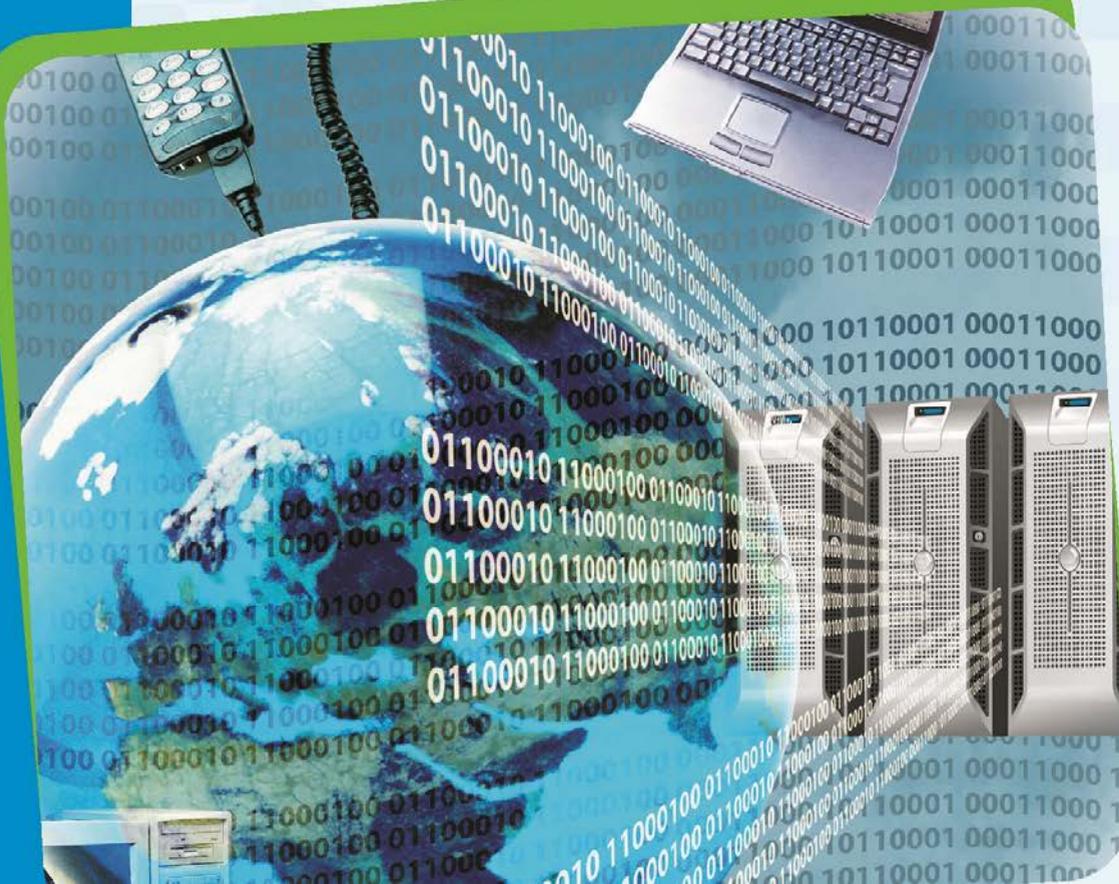


Интернет-
поддержка

8

Е. А. Бондаренко, В. В. Ластовецкий,
А. П. Пилипчук, Е. А. Шестопапов

ИНФОРМАТИКА



Е. А. Бондаренко, В. В. Ластовецкий,
А. П. Пилипчук, Е. А. Шестопапов

ИНФОРМАТИКА

Учебник для 8 класса
общеобразовательных учебных заведений
с обучением на русском языке

Рекомендовано
Министерством образования и науки Украины

Харьков
Издательство «Ранок»
2016

УДК [004:37.016](075.3)

ББК 32.973+74.263.2

Б 81

Рекомендовано Министерством образования и науки Украины
(приказ Министерства образования и науки Украины от 10.05.2016 г. № 491)

Издано за счет государственных средств. Продажа запрещена

Эксперты, осуществившие экспертизу данного учебника в ходе проведения конкурсного отбора проектов учебников для учащихся 8 класса общеобразовательных учебных заведений и давшие заключение о целесообразности присвоения учебнику грифа «Рекомендовано Министерством образования и науки Украины»:

В. Г. Боднар, учитель коммунального заведения «Плисковская общеобразовательная школа I–III ступеней» Погребищенского районного совета Винницкой области,
учитель-методист;

С. В. Гуреева, руководитель районного методического центра учителей информатики
Ленинского района г. Запорожье, ст. учитель;

М. В. Лантева, доцент кафедры информатики Харьковского национального педагогического университета имени Г. С. Сковороды, канд. пед. наук

Переведено по изданию: Информатика : підруч. для 8 кл. загальноосвіт. навч. закл. / О. О. Бондаренко, В. В. Ластовецький, О. П. Пилипчук, Є. А. Шестопалов. — Х. : Вид-во «Ранок», 2016. — 256 с. : іл.

Перевод с украинского Н. Г. Сидоровой

Бондаренко Е. А.

Б 81 Информатика : учебник для 8 кл. общеобразоват. учеб. заведений с обучением на рус. яз. : [пер. с укр.] / Е. А. Бондаренко, В. В. Ластовецький, А. П. Пилипчук, Е. А. Шестопалов. — Харьков : Изд-во «Ранок», 2016. — 256 с. : ил.

ISBN 978-617-09-2942-6

УДК [004:37.016](075.3)

ББК 32.973+74.263.2

Интернет-поддержка
Чтобы воспользоваться
электронными материалами
к учебнику зайдите на сайт
interactive.ranok.com.ua



Служба технической поддержки:
тел. (057) 719-48-65, (098) 037-54-68
(понедельник–пятница с 10:00 до 18:00)
E-mail: interactive@ranok.com.ua

ISBN 978-617-09-2942-6 (рус.)

ISBN 978-617-09-2852-8 (укр.)

© Бондаренко Е. А., Ластовецький В. В.,
Пилипчук А. П., Шестопалов Е. А., 2016

© Хорошенко В. Д., иллюстрации, 2016

© Юхтман О. С., художественное оформление, 2016

© ООО Издательство «Ранок», 2016

ДОРОГИЕ ВОСЬМИКЛАСНИКИ И ВОСЬМИКЛАСНИЦЫ!

Учебник по информатике, который вы держите в руках, предназначен именно для вас, учащихся 8 класса. В 5–7 классах вы уже ознакомились с базовыми понятиями информатики и основами работы с компьютером. Научились создавать графические, текстовые и мультимедийные документы, работать с электронными таблицами, искать в Интернете информационные материалы, составлять простые алгоритмы и создавать по ним программы.

В новом учебном году вы узнаете много интересного и полезного, и мы надеемся, что этот учебник станет вашим надежным помощником. Как же с ним работать?

Учебник состоит из десяти разделов. В начале каждого раздела вы найдете рубрику «Повторяем». Она поможет вам вспомнить материал, который вы уже изучали по данной теме и который пригодится вам при изучении нового.

Раздел состоит из параграфов и практических работ по темам раздела. Каждый параграф содержит *теоретические сведения* по теме урока, *примеры практического применения* полученных знаний, рубрики «Вопросы для самопроверки», «Упражнение», «Компьютерное тестирование».

С помощью «Вопросов для самопроверки» вы сможете проверить, как усвоили учебный материал, а также подготовиться к выполнению упражнений и работе за компьютером.

Рубрика «Упражнение» состоит из теоретических и практических заданий. Упражнения с обозначением ► включают отдельные задания, упражнения с обозначением ►► — задания с пошаговым описанием хода выполнения. Работая над упражнениями, вы получите необходимые практические навыки работы за компьютером.

Оценить свои знания, умения и навыки вам поможет рубрика «Компьютерное тестирование». Предложенные в ней задания можно выполнить в онлайн-режиме на сайте interactive.ranok.com.ua.

В учебнике вы найдете описания всех *практических работ*, предусмотренных программой по информатике. Чтобы выполнить каждую из них, вам нужно повторить материал, изученный на предыдущих уроках, — тогда вы сможете успешно применить свои знания, выполняя задания за компьютером.

В тексте учебника использованы следующие обозначения:



Запомните



Ознакомьтесь с примером



Обратите внимание



Найдите ответ в учебнике



Найдите ответ в Интернете



Выполните практическое задание за компьютером



Выполните тестовые задания за компьютером, используя материалы сайта



Выполните задание повышенной сложности

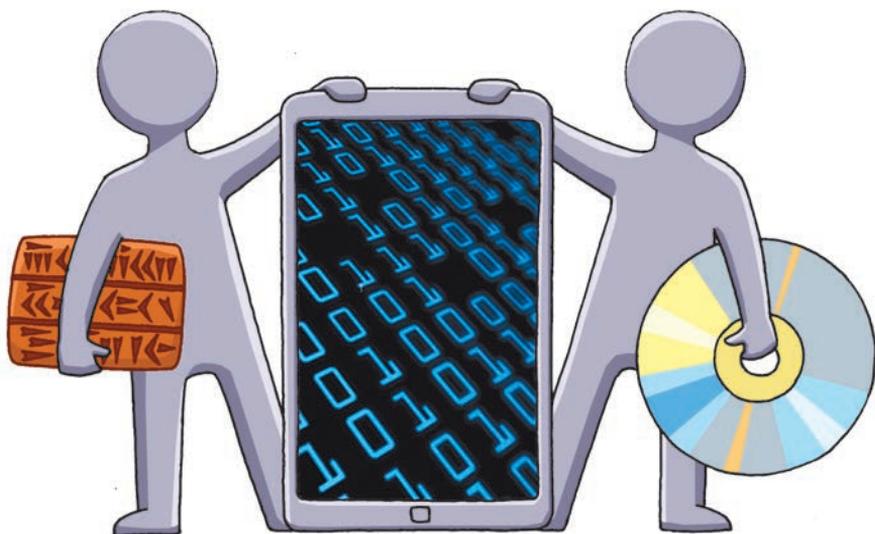


Выполните задание в парах

Желаем вдохновения и успехов!

РАЗДЕЛ 1

КОДИРОВАНИЕ ДАННЫХ



§ 1. Кодирование и декодирование сообщений

§ 2. Двоичное кодирование

Практическая работа 1. Решение задач на определение длины двоичного кода текстовых данных

ПОВТОРЯЕМ



Из курса 5 класса вы знаете, что сведения об окружающем мире и происходящих в нем процессах называют *информацией*. Человек получает информацию из окружающей среды с помощью сообщений, которые воспринимает *органами чувств*. Поэтому по способу восприятия сообщения разделяют на визуальные, вкусовые, обонятельные, звуковые и тактильные. По способу представления сообщения делят на текстовые, числовые, графические, звуковые, видео и комбинированные.

Различные устройства тоже могут принимать сообщения — с помощью датчиков или сенсоров.

Действия, которые можно выполнять с информацией, содержащейся в сообщении, называют *информационными процессами*. К ним относятся хранение, обработка, передача, получение, поиск, защита и т. п.

1. Что такое информация?
2. Кто или что может быть приемником информации?
3. Как человек воспринимает сообщения? Как устройства воспринимают сообщения?
4. Что такое сообщение?
5. Какими бывают сообщения по способу восприятия; по способу представления?
6. Что называют информационными процессами?



В этом разделе вы узнаете о таких информационных процессах, как кодирование и декодирование.

§ 1. Кодирование и декодирование сообщений

В течение жизни человек воспринимает множество сообщений, обрабатывает информацию, сохраняет полученные знания и передает их другим. Как это происходит?

Обработка данных как информационный процесс

Еще в древние времена люди старались хранить полученную из окружающей среды информацию, в частности, в виде наскальных рисунков (рис. 1.1). Разумеется, передавать сохраненную таким образом информацию от одного человека другому было очень неудобно. Однако мы получили данные о жизни наших предков.



Рис. 1.1

Под обработкой данных понимают такой информационный процесс, в результате которого мы получаем новые данные или данные, представленные другим способом.

Кодирование и декодирование сообщений

Сообщения передаются от источника к приемнику по каналам связи.

Во время разговора по телефону звуковые сигналы (голос) преобразуются (кодируются) в электрические сигналы, которые по сети передаются к собеседнику и снова преобразуются (декодируются) в звуковые сигналы (рис. 1.2).



Рис. 1.2

Чтобы передать сообщение с помощью технических устройств, его необходимо закодировать в соответствии с типом канала связи. Сообщение после передачи приобретает понятный для приемника вид.



Кодирование — это преобразование сообщения в удобную для передачи, хранения, обработки форму.

Код — это набор символов или сигналов и правил их использования для кодирования сообщений.

Декодирование — это преобразование закодированного сообщения в форму, понятную для приемника.

После изобретения письменности как способа кодирования естественного языка наиболее распространенным стал способ представления сообщений с помощью символов, например иероглифов, букв и т. п.



Мечта

Рис. 1.3

» Отдельными иероглифами обозначают определенные объекты или понятия (рис. 1.3), буквами — отдельные звуки устной речи, из которых состоят слова, предложения и т. д.

Одни и те же сообщения можно кодировать разными способами. Для кодирования сообщений также используют цифры, знаки, жесты, световые, звуковые и электрические сигналы и т. п.

Способы кодирования звука [а]:



Русский алфавит



Семафорная азбука



Азбука Морзе

» Звук [а] можно закодировать и буквой «А», и определенным жестом, и в виде короткого и длинного звуковых сигналов, которые на бумаге обозначают точкой и тире (рис. 1.4).

Вы можете придумать собственный код для передачи текстовых сообщений, например, заменяя буквы другими символами, рисунками или цифрами. Для декодирования сообщения нужно знать, как оно закодировано.

Обратите внимание на то, что один и тот же символ имеет разные значения в зависимости от сферы его использования.

» Большая латинская буква V может обозначать объем тела и римскую цифру 5, буква P — звук, периметр, вес тела.

Принимать, декодировать и обрабатывать сообщения может человек, созданный им прибор, животное.

Рис. 1.4

Обученная служебная собака выполняет команды, закодированные человеком в виде жестов (рис. 1.5).



Рис. 1.5

Способы кодирования постоянно совершенствуются, ведь удобно закодированные данные способствуют более эффективному их использованию в различных сферах человеческой деятельности.

Вопросы для самопроверки



1. Как первобытные люди кодировали сообщения?
2. Что такое кодирование; код; декодирование?
3. Как сообщение передается от источника к приемнику?
4. Как кодируются звуки естественного языка?
5. Можно ли закодировать понятие?
6. Приведите несколько способов кодирования звука [а].

Упражнение 1



1. Рассмотрите наскальный рисунок (см. рис. 1.1) и определите, какая, на ваш взгляд, информация в нем содержится.
2. Найдите в Интернете азбуку Морзе и закодируйте слово «Информатика».
3. Придумайте собственный код и с его помощью запишите какую-нибудь фразу в тетради. Обменяйтесь тетрадями с одноклассником (одноклассницей) и, пользуясь кодом (ключом), декодируйте написанное им (ею).



Компьютерное тестирование



Выполните тестовое задание 1 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 2. Двоичное кодирование

Компьютер — универсальное устройство для обработки данных разных типов. Но как же он воспринимает, распознает, сохраняет данные? Рассмотрим это на примере текстовых данных.

Двоичное кодирование

В компьютере данные представлены электрическими или магнитными сигналами, которые принимают только два значения: 0 — выключено (нет тока, размагничено) или 1 — включено (есть ток, намагничено).

Любые данные, которые обрабатывает компьютер: числа, текст, рисунки, фотографии, звуки, музыка, видео и т. п. — кодируют с помощью нуля и единицы. Такой способ кодирования называют **двоичным**. Именно двоичное кодирование реализуется в электронных устройствах.

Кодирование символов

Для кодирования данных в компьютере используют многозначные двоичные коды. В таких кодах цифру 0 или 1 называют **битом** (сокр. от англ. *binary digit* — двоичное число).

Для кодирования символов в компьютерах обычно используют 8-битный двоичный код.



Последовательность из 8 двоичных разрядов (битов) называют **байтом**.

Существует $2^8=256$ разных комбинаций из 8 битов. Этого количества вариантов достаточно для кодирования прописных и строчных букв английского и русского алфавитов, цифр, знаков препинания, а также графических элементов.

Все символы, используемые в текстах, для удобства кодирования (декодирования) сводят в специальные таблицы двоичных кодов.

Существует таблица кодов ASCII (*American Standard Code for Information Interchange* — американский стандартный код для обмена информацией), которая содержит 128 символов, в том числе буквы латинского алфавита.

Для кодирования букв других алфавитов эту таблицу дополняют до 256 символов. Так, для букв русского алфавита используют таблицу кодов KOI8-R (*Код Обмена Информацией 8-битный Русский*), фрагмент которой представлен ниже.

Обычно в таблицах кодов возле или вместо двоичного кода указывают десятичный. Так, строчная русская буква «а» в таблице кодов KOI8-R имеет десятичный код 193, который соответствует двоичному 11000001.

Фрагмент таблицы кодов KOI8-R

192	11000000	ю	198	11000110	ф	204	11001100	л	210	11010010	р
193	11000001	а	199	11000111	г	205	11001101	м	211	11010011	с
194	11000010	б	200	11001000	х	206	11001110	н	212	11010100	т
195	11000011	ц	201	11001001	и	207	11001111	о	213	11010101	у
196	11000100	д	202	11001010	й	208	11010000	п	214	11010110	ж
197	11000101	е	203	11001011	к	209	11010001	я	215	11010111	в

В современных компьютерных текстовых документах распространено также кодирование символов 16-битным кодом Юникод (*Unicode* — унифицированное кодирование), содержащий алфавиты практически всех языков мира.

Единицы измерения длины двоичного кода

Объем данных измеряется длиной двоичного кода.

При стандартном 8-битном кодировании текста каждый символ кодируется одним байтом.

Знаки препинания и пробел также являются символами, которые имеют собственные коды.

Текст Андрей играет в футбол состоит из 22 символов. Поэтому при 8-битном кодировании символов объем этого текста равен 22 байтам.

На практике удобнее пользоваться большими единицами измерения: килобайт (Кбайт), мегабайт (Мбайт), гигабайт (Гбайт), терабайт (Тбайт) и др. В информатике префиксы *кило-*, *мега-*, *гига-*, *тера-* имеют несколько иное значение, чем в других науках:

$$1 \text{ Кбайт} = 2^{10} \text{ байтов} = 1024 \text{ байта} \approx 10^3 \text{ байтов}$$

$$1 \text{ Мбайт} = 2^{10} \text{ Кбайтов} = 1024 \text{ Кбайта} \approx 10^6 \text{ байтов}$$

$$1 \text{ Гбайт} = 2^{10} \text{ Мбайтов} = 1024 \text{ Мбайта} \approx 10^9 \text{ байтов}$$

$$1 \text{ Тбайт} = 2^{10} \text{ Гбайтов} = 1024 \text{ Гбайта} \approx 10^{12} \text{ байтов}$$

В начале XXI в. был введен стандарт именованя единиц измерения объемов данных: кибибайт, мебибайт, гибибайт и т. д. Но эти наименования практически не используют.

» Пусть на странице содержится 56 строк по 64 символа в каждой строке. Вычислим объем данных (длину двоичного кода) на странице: $56 \cdot 64 = 3584$ (байта).

Чтобы вычислить объем данных в книге, нужно количество символов на одной странице умножить на количество страниц. Таким образом, объем книги, содержащей 256 подобных страниц: $3584 \cdot 256 = 917\,504$ (байта) = 896 (Кбайтов) = 0,875 (Мбайта).

Не следует отождествлять длину двоичного кода текстового сообщения с объемом информации, которая в нем содержится.

» Двоичный код буквы «Р» имеет длину 1 байт; эта же буква, изображенная на дорожном знаке, содержит информацию для водителя о том, что рядом со знаком обустроено место для стоянки.

Теперь вы понимаете, в каких единицах и каким способом можно измерять объем текстовых данных. С измерением объемов видео- и аудиоданных, которыми оперирует компьютер, вы ознакомитесь позже.

Вопросы для самопроверки



1. Как кодируются данные в компьютере?
2. Почему в компьютере используется двоичное кодирование?
3. Каким образом кодируются буквы в компьютере?
4. Что такое бит; байт?

5. В каких единицах измеряют длину двоичного кода?
6. Сколько байтов содержит 1 килобайт; 1 мегабайт?

Упражнение 2



- ▶ 1. Найдите в Интернете таблицу кодов символов ASCII. Пользуясь этой таблицей, декодируйте сообщение:
01001000 01100101 01101100 01101100 01101111 00100001.



- ▶ 2. Определите, содержание скольких книг (см. пример на с. 12) можно сохранить в электронном виде на DVD-диске объемом 4,7 Гбайта.



- ▶ 3. Определите приблизительный объем данных на странице учебника.



Компьютерное тестирование



Выполните тестовое задание 2 с автоматической проверкой на сайте interactive.ranok.com.ua.

Практическая работа 1



Решение задач на определение длины двоичного кода текстовых данных

Задание: вычислить длину двоичного кода текстовых данных.

Оборудование: компьютер с операционной системой Windows; учебник; тетрадь.

Ход работы

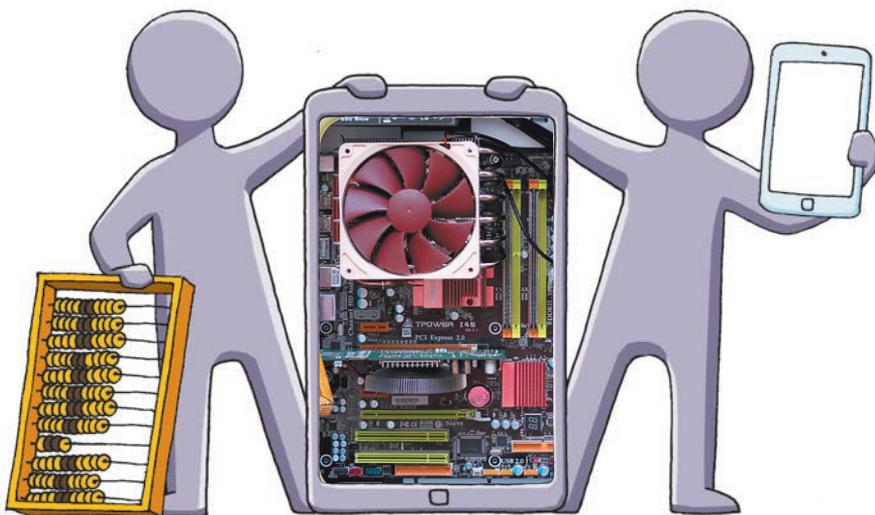
Во время работы за компьютером соблюдайте правила безопасности.

- ▶ 1. Запишите в тетради свое имя и фамилию. Определите длину двоичного кода записанного текста при однобайтном кодировании символов.
- ▶ 2. Запустите на выполнение программу Блокнот. Наберите свое имя и фамилию через пробел и сохраните файл с названием Практическая работа 1 в соответствующей папке. С помощью контекстного меню файла (пункт Свойства) определите размер файла в байтах и сравните с данными, полученными в п. 1.
- ▶ 3. С помощью фрагмента таблицы кодов KOI8-R на с. 11 закодируйте слово «байт» и запишите его в тетради с помощью двоичных и десятичных кодов.
- ▶ 4. Используя программу Калькулятор, определите приблизительный объем памяти, необходимый для хранения в электронном виде одной страницы данного учебника с текстом (без рисунков) и всего учебника.
- ▶ 5. Вычислите, сколько таких учебников может поместиться на флеш-носителе объемом 8 Гбайтов.
- ▶ 6. Все расчеты, полученные в пп. 1–5, запишите в текстовый документ Практическая работа 1 и сохраните файл. Завершите работу за компьютером.

Сделайте вывод: как определять длину двоичного кода текстовых данных.

РАЗДЕЛ 2

АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА



§ 3. История обработки информационных объектов

§ 4. Архитектура компьютера. Технические характеристики устройств компьютера

Практическая работа 2. Выбор конфигурации компьютера исходя из потребностей пользователя

§ 5. Программное обеспечение компьютера

Практическая работа 3. Архивация и разархивация данных

ПОВТОРЯЕМ



Вы знаете, что компьютер состоит из нескольких основных частей: *системного блока*, *монитора*, *мыши* и *клавиатуры*. К нему также могут подключаться другие устройства.

Все устройства компьютера делятся на устройства *ввода* — клавиатура, мышь, микрофон, сканер, камера и т. п.; *вывода* — монитор, принтер, наушники, колонки и т. п.; *обработки* — процессор; *хранения* — внутренняя и внешняя память (жесткие магнитные диски, CD- и DVD-диски, флеш-память и т. п.).

Как известно, компьютер не может работать без программ. *Программа* — это последовательность команд, предназначенных для выполнения компьютером. Наиболее важным комплектом программ является операционная система. Именно она управляет работой компьютера, обеспечивает взаимодействие всех его устройств, а также компьютера и пользователя.

1. Назовите основные устройства компьютера.
2. Какие устройства хранения информации вы знаете?
3. Назовите устройства ввода информации.
4. Назовите устройства вывода информации.
5. Что такое компьютерная программа?
6. Каково назначение операционной системы?



В этом разделе вы подробнее ознакомитесь с аппаратным и программным обеспечением компьютера, а также со способами взаимодействия пользователя и компьютера.

§ 3. История обработки информационных объектов

История человечества неразрывно связана с познанием мира, информационными процессами. С давних времен люди изобретали различные устройства, позволяющие облегчить процесс обработки информации (в частности, таких информационных объектов, как звук, изображение, текст, число).

Этапы развития вычислительных средств

Различают несколько этапов развития вычислительной техники.

На **домеханическом этапе** для счета использовали подручные средства: камешки, узелки, зерна и т. п. Со временем разные народы стали применять для расчетов различные приспособления и устройства. Наибольшее распространение получили абак и его разновидности.

На **механическом этапе** с развитием промышленности и торговли был создан ряд механических вычислительных устройств. У истоков механической вычислительной техники стояли известные ученые, инженеры и исследователи (рис. 3.1).



Леонардо
да Винчи



Блез
Паскаль



Готфрид
Лейбниц



Чарльз
Бэббидж



Ада
Лавлейс

Рис. 3.1

Первый известный нам эскиз механического вычислительного устройства (ок. 1500 г.) был выполнен итальянским изобретателем и художником Леонардо да Винчи. Первое действующее устройство для сложения и вычитания чисел («паскалина») разработал в 1642 г. Блез Паскаль, со временем известный французский ученый. В 1673 г. немецкий математик и физик Готфрид Вильгельм Лейбниц создал механический арифмометр на основе двоичной

системы счисления, которая используется в современных компьютерах. В 1833 г. английский математик Чарльз Беббидж изобрел аналитическую вычислительную машину. Описание и программу для нее составила Ада Лавлейс — ее и считают первым в мире программистом.

Значительный вклад в дальнейшее развитие вычислительных устройств внесли многие ученые, инженеры, исследователи из разных стран мира.

» В 1938 г. немецкий инженер Конрад Цузе построил первую электромеханическую программируемую цифровую машину.

На электронном этапе были созданы электронно-вычислительные машины (компьютеры), развитие которых продолжается и в наше время.

Основные этапы развития компьютеров

Рассмотрим историю развития компьютеров как универсального средства для работы с различными информационными объектами. В зависимости от того, какие основные элементы использовались для изготовления, выделяют несколько поколений компьютеров.



Рис. 3.2

» **Компьютеры первого поколения** были созданы в 40-х годах XX в. на базе *электронных ламп* (рис. 3.2). Быстродействие таких компьютеров не превышало 20 тыс. операций в секунду.



Рис. 3.3

» Первый компьютер под названием «ENIAC» был создан в США в 1945 г. под руководством Джона Моучли и Джона Преспера Эккерта. Компьютер массой 30 т выполнял до 5 тыс. операций в секунду и занимал целое здание с кондиционерами для охлаждения.

Основоположником вычислительной техники в СССР был академик Сергей Алексеевич Лебедев (рис. 3.3). Под его руководством в Киевском электротехническом институте в 1951 г. создан компьютер под названием «МЭСМ» (*малая электронная счетная машина*).



Рис. 3.4

Компьютеры второго поколения появились в 50-х годах XX в. на базе *транзисторов* (рис. 3.4). Размеры компьютеров уменьшились в десятки раз, а быстродействие достигло 100 тыс. операций в секунду.

» В Институте кибернетики Академии наук Украины в 60-х годах XX в. под руководством Виктора Михайловича Глушкова (рис. 3.5) была создана серия машин для инженерных расчетов (сокращенно МИР).



Рис. 3.5

Компьютеры третьего поколения были созданы в 70-е годы XX в. с появлением *интегральных схем* (рис. 3.6). В таких схемах на кристаллической пластине площадью до 1 см² размещались тысячи элементов. Размеры компьютеров снова уменьшились, а быстродействие достигло миллионов операций в секунду.



Рис. 3.6

Компьютеры четвертого поколения появились благодаря использованию *больших интегральных схем* (рис. 3.7). Такие схемы на кристаллической пластине площадью около 1 см² содержат миллионы элементов. Быстродействие компьютеров достигло сотен миллионов операций в секунду. Появились настольные компьютеры.

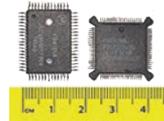


Рис. 3.7

» Один из первых персональных компьютеров был создан в США в 1976 г. основателями фирмы Apple Стивом Джобсом и Стивом Возняком (рис. 3.8).



Рис. 3.8

Уменьшение размеров и повышение продуктивности компьютеров способствует их широкому внедрению во все сферы жизни. На начало 2016 г. количество компьютеров в мире достигло 2 млрд и продолжает расти.

Виды современных компьютеров

В зависимости от вычислительных возможностей современные компьютеры делятся на несколько основных видов.

- **Суперкомпьютеры** — многопроцессорные системы для выполнения сложных вычислений.

» В Украине созданы мощные центры суперкомпьютерных вычислений в Национальном техническом университете «Киевский политехнический институт» и Институте кибернетики им. В. М. Глушкова НАН Украины.

- **Мейнфреймы** — высокопроизводительные надежные серверы для использования в критически важных системах (управление электростанциями, продажа билетов, облачные технологии и т. п.).

- **Микрокомпьютеры** — компьютеры, предназначенные для реализации простых задач управления, например бытовой техникой.
- **Персональные компьютеры (ПК)** — компьютеры универсального назначения, рассчитанные на одного пользователя.

Рассмотрим особенности персональных компьютеров в зависимости от их размеров (см. форзац, рис. 3.9).

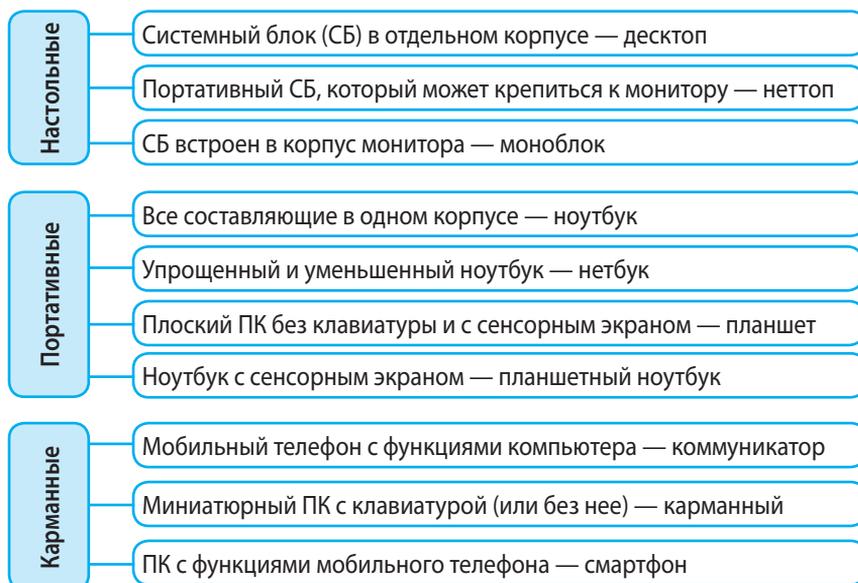


Рис. 3.9



Рис. 3.10

Мощность компьютеров постоянно увеличивается, а сфера применения расширяется. Популярными становятся новые виды компьютеров: роботы-игрушки, роботы — помощники по дому и т. п. (рис. 3.10).

Вопросы для самопроверки



1. Назовите этапы развития вычислительной техники.
2. Кто создал первый механический вычислитель?
3. Кого считают первым в мире программистом?
4. Что является элементной базой персональных компьютеров?
5. Назовите основоположников разработки вычислительной техники в Украине.
6. Назовите виды современных компьютеров.

Упражнение 3



1. Найдите в тексте параграфа данные о быстродействии компьютеров.



2. Запустите текстовый процессор, создайте новый документ. Найдите в Интернете информацию о вкладе Джона Винсента Атанасова и Михаила Филлиповича Кравчука в создание первых компьютеров и скопируйте в документ.



3. Напишите небольшое сочинение на тему «Компьютеры будущего», введите его текст в документ. Сохраните файл с именем Упражнение 3 в своей папке.



Компьютерное тестирование



Выполните тестовое задание 3 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 4. Архитектура компьютера. Технические характеристики устройств компьютера

Большинство современных компьютеров построены по принципам, сформулированным американским ученым Джоном фон Нейманом (рис. 4.1) в 1948 г. С того времени изменился внешний вид компьютеров, появились новые устройства, но не изменилась общая схема работы компьютера (рис. 4.2).



Рис. 4.1

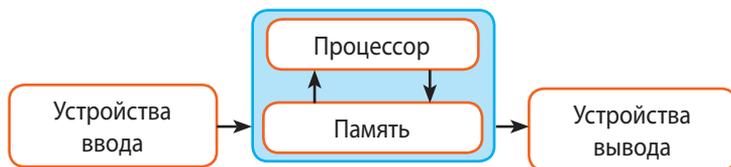


Рис. 4.2

Архитектура компьютера



Архитектура компьютера — это описание устройства компьютера и принципов работы его составляющих.

Основные узлы настольного персонального компьютера расположены в системном блоке.

Системная (материнская) плата — самый большой узел компьютера, на котором установлены *процессор, внутренняя память, системная шина, порты* (разъемы, слоты).

Системная шина (магистраль) — набор проводников и устройств, предназначенных для обмена данными между всеми компонентами компьютера.

Рассмотрим более подробную схему обмена информацией между устройствами компьютера (рис. 4.3).

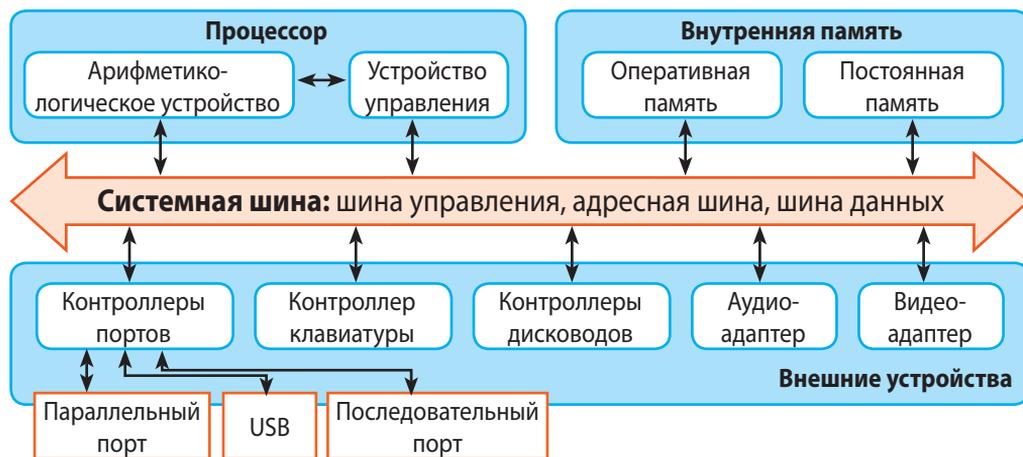


Рис. 4.3

Процессор

Процессор (центральный процессор) — это основная микросхема компьютера, предназначенная для выполнения элементарных арифметических и логических операций и управления работой компьютера.

Процессор состоит из двух основных частей: *устройства управления* и *арифметико-логического устройства*. Процессор имеет свою внутреннюю память.

Конструктивно современный процессор (микропроцессор) — это большая интегральная схема, помещенная в защитный корпус (рис. 4.4).

Быстродействие процессора — основной фактор, влияющий на производительность компьютера.



Рис. 4.4

Основные технические характеристики процессора:

- *тактовая частота* — количество тактов в секунду; измеряется в герцах (Гц), определяет быстродействие процессора;
- *объем кэш-памяти* — объем сверхбыстрой памяти, используемой процессором для обработки наиболее часто применяемых данных;
- *разрядность* — количество бит данных, обрабатываемых процессором одновременно (разрядность современных процессоров — 64 бит);
- *количество ядер* — количество отдельных вычислительных ядер (отдельных процессоров) в одной микросхеме;
- *сокет* — тип разъема для процессора на системной плате.

» Рассмотрим запись краткой характеристики процессора:

Intel Core 2 Duo E8600 3,33 ГГц/6 Мб/LGA775.

Имеем: *Intel* — производитель; *Core 2 DUO* — тип — 64-разрядный двухъядерный; *E8600* — модель; *3,33 ГГц* — тактовая частота; *6 Мб* — объем кэш-памяти; *LGA775* — сокет.

Память компьютера

Память компьютера предназначена для временного или постоянного хранения данных. Выделяют память *внутреннюю* и *внешнюю*; *энергозависимую* и *энергонезависимую* (рис. 4.5).

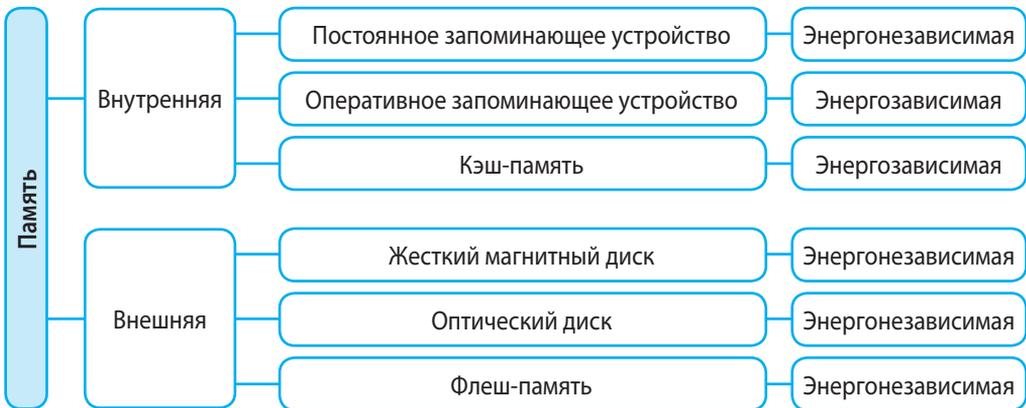


Рис. 4.5

После отключения компьютера от сети данные из энергозависимой памяти исчезают, а в энергонезависимой памяти хранятся постоянно.



Внутренняя память — это быстродействующая память, размещенная на системной плате компьютера в виде микросхем.

Внутренняя память делится на *оперативную* и *постоянную*.

Оперативная память (ОЗУ — оперативное запоминающее устройство, или RAM — Random Access Memory) предназначена для хранения данных и программ во время работы компьютера.

Основной технической характеристикой ОЗУ является его объем.

Постоянная память (ПЗУ — постоянное запоминающее устройство, или ROM — Read-Only Memory) предназначена для хранения программ и данных, обеспечивающих работу компьютера после включения.

В постоянную память записана базовая система ввода-вывода (BIOS — Basic Input-Output System) — программа для управления работой устройств, загрузки операционной системы и др.

Внешние запоминающие устройства



Внешняя память — память, предназначенная для долговременного хранения данных на внешних носителях.

Рассмотрим некоторые устройства внешней памяти:

Устройство	Название устройства	Принцип действия
	Накопитель на жестких магнитных дисках (винчестер)	Данные записываются и считываются специальными магнитными головками
	Оптические CD- и DVD-накопители	Данные записываются или считываются лазерным лучом
	Флеш-накопитель	Данные кодируются электрическими зарядами

Устройства ввода-вывода данных

Вы уже знакомы с основными устройствами ввода-вывода данных — клавиатурой, мышью, монитором. Кроме этих устройств, есть и другие. Работа любых устройств и компьютера требует согласования действий.



Адаптер (контроллер) — устройство, предназначенное для подключения к компьютеру другого устройства и управления его работой.

Существуют адаптеры для работы с видеокамерами и мониторами (видеоадаптеры); для работы с аудиоустройствами (аудиоадаптеры) и др.

Монитор — устройство вывода, предназначенное для визуального отображения текстовой и графической информации. В зависимости от принципа действия различают мониторы на электронно-лучевой трубке (CRT), на жидких кристаллах (LCD), на органических светодиодах (OLED), «электронные чернила» (e-ink), плазменные, проекционные и др.

Размер монитора задается длиной диагонали экрана в дюймах ($1'' = 2,54$ см). Изображение на экране формируется из отдельных точек — пикселей. Важной характеристикой монитора является *разрешающая способность* — количество пикселей на дюйм (ppi — pixels per inch).

Принтер — устройство вывода, предназначенное для вывода информации на твердые носители (бумагу, пластик и т. п.). В зависимости от технологии печати различают следующие типы принтеров:

Тип принтера	Описание технологии печати
Матричный	Изображение формируется из точек от ударов игл по красящей ленте
Струйный	Изображение формируется каплями чернил разного цвета
Лазерный, светодиодный	Изображение формируется из порошкового тонера на наэлектризованной поверхности, а затем переносится на бумагу

Существуют также принтеры с другими технологиями печати.

Основные технические характеристики принтера:

- *разрешающая способность* — наибольшее количество точек на дюйм (dpi — dots per inch);
- *скорость печати* — количество страниц, отпечатанных за минуту;
- *формат бумаги* — наибольший размер бумаги для печати;
- *параметры печати* — возможность печати без полей, двусторонняя печать и т. п.;
- *количество красителя* — на сколько страниц хватает красителя.

Все чаще используются 3D-принтеры, которые формируют объемные предметы методом послойного нанесения вещества.

Мультимедийные устройства

Компьютер обрабатывает данные, представленные разными способами (текстовые, графические, звуковые, комбинированные и др.). Комбинированные данные называют мультимедийными.

Мультимедиа — это сочетание разных способов представления данных и устройств для их воспроизведения. Для ввода мультимедийных данных к компьютеру подключаются такие внешние устройства, как микрофон, сканер, видеокамера и т. п., для вывода — монитор,

проектор, наушники, принтер и т. п. Для согласования действий этих устройств и компьютера используют соответствующие адаптеры.



Аудиоадаптер (звуковая карта) — устройство для воспроизведения и записи звука.

Видеоадаптер (видеокарта) — устройство поддержки преобразования данных в видеосигнал для монитора или проектора.

Видео- или аудиоадаптеры могут быть встроены (интегрированы) в системную плату или установлены в разъем (слот) на ней. Основными характеристиками видеоадаптера является частота и объем видеопамяти. От них зависит качество изображения на экране монитора.

Отображение информации происходит при помощи **видеосистемы** компьютера, которая включает в себя видеоадаптер, монитор и специальные программы для их подключения.

Конфигурация компьютера

Конфигурация компьютера определяется составом и характеристиками его устройств.



Базовой конфигурацией компьютера называется обязательный набор его устройств.

Сегодня к базовой конфигурации относят системный блок (с основными устройствами), монитор, клавиатуру и мышь. Конфигурацию компьютера можно изменить в соответствии с потребностями пользователя.

В описании конфигурации компьютера используют обозначения характеристик монитора, процессора, оперативной памяти, накопителя на жестких магнитных дисках, видеоадаптера, оптического привода, сетевой карты и т. д.



Рассмотрим запись краткой характеристики конфигурации компьютера: 19.5" (1920x1080) LED/AMD Athlon IIx4 750K (3.4 ГГц)/RAM 8 Гб/HDD 1 Тб/nVidia GeForce GT730 2 Гб/DVD+RW/LAN.

Вопросы для самопроверки



1. Что такое процессор?
2. Что определяет быстродействие компьютера?
3. Для чего предназначена оперативная память?
4. Каковы функции постоянной памяти компьютера?
5. Каково назначение устройств внешней памяти?
6. Чем определяется конфигурация компьютера?

Упражнение 4



1. Найти и проанализировать технические характеристики процессора и оперативной памяти (ОЗУ) вашего компьютера.

1) Запустите текстовый процессор и создайте новый документ. В контекстном меню объекта Компьютер откройте вкладку Свойства. Найдите, расшифруйте и введите в документ характеристики процессора и ОЗУ.



2) По каталогам интернет-магазинов найдите описание процессоров. Скопируйте в документ ссылку на страницу и запись краткой характеристики одного из них. По записи определите и запишите разрядность процессора, количество ядер, тактовую частоту, объем кэш-памяти, сокет.

3) Найдите в Интернете и скопируйте в документ описание следующих характеристик ОЗУ: пропускная способность, тип, объем, частота. Сохраните файл с именем Упражнение 4 в соответствующей папке.



2. Запишите краткую характеристику конфигурации своего домашнего компьютера (см. пример на с. 26).

Компьютерное тестирование



Выполните тестовое задание 4 с автоматической проверкой на сайте interactive.ranok.com.ua.



Практическая работа 2

Выбор конфигурации компьютера исходя из потребностей пользователя

Задание: определить конфигурацию компьютера для компьютерной игры с заданными системными требованиями: 30 Гб свободного места на жестком диске; 8 Гб оперативной памяти; процессор i7 4790 3.6 ГГц или AMD FX9 590 4.7 ГГц.

Оборудование: компьютер с установленными браузером и текстовым процессором, подключенный к сети Интернет.

Ход работы

Во время работы за компьютером соблюдайте правила безопасности.



▶ 1. По каталогам интернет-магазинов подберите составляющие для комплектации компьютера необходимой конфигурации.



▶ 2. Запустите текстовый процессор. Создайте новый документ, введите название и стоимость таких составляющих компьютера: корпус системного блока, системная плата, процессор, монитор, оперативная память, накопитель на жестких магнитных дисках, оптический привод, видеоадаптер (если он не встроен в системную плату), сетевая карта, клавиатура, мышь.

▶ 3. Определите составляющие, от которых зависит соответствующая конфигурация компьютера. Скопируйте их названия, краткие технические характеристики и ссылки на страницы.

▶ 4. Запишите, исходя из каких характеристик вы выбирали процессор, материнскую плату.

▶ 5. Сохраните файл с именем Практическая работа 2 в своей папке. Завершите работу за компьютером.

Сделайте вывод: как подобрать конфигурацию компьютера исходя из потребностей; какие комплектующие являются обязательными; какими характеристиками они должны обладать.

§ 5. Программное обеспечение компьютера

Мы рассмотрели основные устройства компьютера, то есть аппаратную составляющую. Но поскольку компьютер не может работать без программ, рассмотрим программную составляющую.



Программное обеспечение — совокупность программ, установленных на компьютере и предназначенных для обеспечения его работы и решения задач пользователя.

Программное обеспечения (ПО) разделяют на *системное, прикладное и служебное.*

Системное программное обеспечение

Системное программное обеспечение (системные программы) предназначено для управления работой устройств компьютера и обмена данными между ними. Среди системных программ особое место занимает операционная система.

Как вы знаете, *операционная система (ОС)* — это комплект программ управления ресурсами, вычислительными процессами компьютера, а также для обеспечения взаимодействия с пользователем.

ОС автоматически загружается при включении компьютера.

Основные функции ОС:

- ввод-вывод данных, пуск-остановка программ и т. п.;
- обеспечение работы устройств ввода-вывода;
- загрузка программ в ОС и их выполнение;
- управление оперативной памятью;
- управление доступом к данным на внешних носителях;
- обеспечение интерфейса пользователя;
- поддержка файловой системы.

Рассмотрим типы ОС в зависимости от основных признаков (рис. 5.1).



Рис. 5.1

» На персональных компьютерах используют ОС Windows, Linux, MacOS. Для мобильных устройств предназначены ОС Android, Windows CE, iOS, PalmOS и др.

Прикладное программное обеспечение

Прикладное программное обеспечение (прикладные программы) предназначено для решения необходимых пользователю задач.

Прикладные программы делятся на группы в зависимости от назначения:

Прикладные программы	Назначение
Программы общего назначения	Различные сферы деятельности (например, работа с текстом, рисунками, таблицами и т. п.)
Программы специального назначения	Отдельные сферы деятельности или предприятия (например, создание видеоэффектов и т. п.)
Инструментальные программные средства	Создание новых программ (системы программирования)

Служебное программное обеспечение

Служебное программное обеспечение (служебные программы) предназначено для диагностики аппаратной и программной составляющих компьютера. К служебным программам относятся программы для архивации данных, форматирования и проверки целостности диска и т. п.



Форматирование носителя данных — процесс разметки области хранения данных.

Форматирование обычно выполняется перед первым использованием диска или в случае появления серьезных сбоев в его работе. Эту операцию можно выполнить с помощью контекстного меню носителя; после завершения форматирования данные на носителе будут удалены. При форматировании можно проверить и целостность носителя.



Архивация данных — процесс упаковки файлов в архив путем сжатия данных, хранящихся в них.

В результате архивации создается *архив* — файл, содержащий один или более файлов со сжатыми данными.

Распространенным методом сжатия данных является нахождение повторяющихся фрагментов и их кодирование. Такими фрагментами могут быть пиксели одинакового цвета растрового изображения, повторяющиеся слова или символы в тексте и т. п.

Степень сжатия данных, то есть отношение объема несжатого файла к объему его архива, зависит от типа файла и метода сжатия.

» В текстах на украинском языке вместе со знаками препинания и цифрами используется около 60 символов. Чтобы сжать такой текст, вместо 8-битного кодирования можно использовать 6-битное и без потери данных сэкономить 25 % памяти.



Разархивация — процесс возврата данных архива в исходное состояние.

Операции архивации и разархивации выполняют с помощью служебных программ — архиваторов. К ним относятся программы 7-Zip (бесплатный), WinRAR и WinZIP (условно-бесплатные).

Программы-архиваторы позволяют создавать архивы разных типов: *обычные, многотомные, самораспаковывающиеся*.

Для создания **обычного архива** нужно:

- 1) вызвать контекстное меню файла (группы файлов, папки);
- 2) выбрать для архиватора пункт меню Добавить в архив...;
- 3) выполнить необходимые настройки в диалоговом окне Имя и параметры архива;
- 4) щелкнуть кнопку ОК.

Архив появится в одной папке с исходным файлом и будет иметь имя файла, указанное на шаге 3.

Если архивный файл имеет очень большой объем, не позволяющий, например, передать его через Интернет или сохранить на одном носителе, его нужно разделить на несколько частей (томов). Такой архив называется **многотомным**.

Архивный файл с расширением .exe после запуска распаковывается автоматически. Таким **самораспаковывающимся (SFX)** архивом пользуются, если на компьютере получателя не установлен нужный архиватор.

Кроме операций архивации и разархивации, современные архиваторы позволяют просматривать содержимое архива, добавлять файлы в архив, удалять отдельные файлы из архива, проверять целостность архивного файла и т. п.

Лицензии на программное обеспечение

Компьютерные программы как результат труда многих людей защищены авторским правом, поэтому их нельзя использовать без разрешения правообладателя.



Лицензия на программное обеспечение — это специальное соглашение, предоставляющее право на его использование.

Закон об авторском праве оговаривает условия использования программ, запрещает выполнять не разрешенные лицензией действия.

Рассмотрим некоторые виды лицензий на программное обеспечение:

Вид лицензии	Описание программы
Коммерческая (commercial)	Собственнические программы (необходимо купить программу)
Некоммерческая (non-commercial use)	Программы, бесплатные для некоммерческого использования (например, упрощенная версия коммерческой программы)
Условно-бесплатная (shareware)	Программы для бесплатного использования на определенных условиях (например, для ознакомления в течение какого-то периода)
Свободная (free software)	Программы, которые разрешается копировать, модифицировать, распространять

Популярной лицензией на свободное программное обеспечение является лицензия GNU GPL.

Инсталляция и деинсталляция программ

Большинство программ перед использованием требуют установки, в ходе которой согласовываются действия программы с ОС.



Инсталляция — процесс установки программ на компьютере пользователя.

При инсталляции происходит копирование всех или отдельных компонентов программы на жесткий магнитный диск, в системные файлы вносятся необходимые записи и т. п.



Деинсталляция — процес удаления с компьютера установленных программ.

При деинсталляции также вносятся изменения в системные файлы, поэтому просто удалить файлы (папки) программы из компьютера нельзя. Нужно воспользоваться программой деинсталляции конкретной программы или системной утилитой для удаления программ: Пуск → Панель управления → Программы и компоненты.

Инсталлировать и деинсталлировать программу может только пользователь с правами администратора.

Совместимость программного обеспечения

Операционные системы могут значительно отличаться друг от друга. В большинстве случаев программа, разработанная для одной ОС (например, Windows 7), не будет работать на компьютере с другой ОС (например, Linux). В таких случаях говорят о *несовместимости программы и ОС*.

» Для настройки совместимости программы с разными версиями ОС Windows иногда достаточно выбрать в контекстном меню команду Свойства → Совместимость и указать нужную версию ОС.

Прежде чем приобрести и установить на компьютер какую-либо программу, следует обратить внимание на требования к нему и ОС.

Вопросы для самопроверки



1. Что такое программное обеспечение компьютера?
2. Каково назначение прикладных программ?
3. Каково назначение служебных программ?
4. Что такое архивация данных; разархивация данных?
5. Для чего форматируют носители данных?
6. В чем заключается процесс инсталляции программ?

Упражнение 5



» 1. Найдите в Интернете ответ на вопрос, позволяет ли лицензия GNU GPL продавать копии программ. Создайте в текстовом процессоре документ, введите в него адрес сайта, где был найден ответ, скопируйте ответ. Сохраните файл с именем Упражнение 5 в свою папку.



» 2. В учетной записи пользователя с правами администратора установите на компьютер программу, предложенную учителем. Запустите программу, продемонстрируйте ее учителю и деинсталлируйте.



» 3. Составьте примерный список программ, необходимых для работы за компьютером учащимся; программистам; фотохудожникам.

Компьютерное тестирование



Выполните тестовое задание 5 с автоматической проверкой на сайте interactive.ranok.com.ua.



Практическая работа 3

Архивация и разархивация данных

Задание: выполнить сжатие данных в архивы разных типов с помощью программы-архиватора, выполнить разархивирование, сравнить степень сжатия данных.

Оборудование: компьютер с установленной программой-архиватором, папка с файлами для архивации.

Ход работы

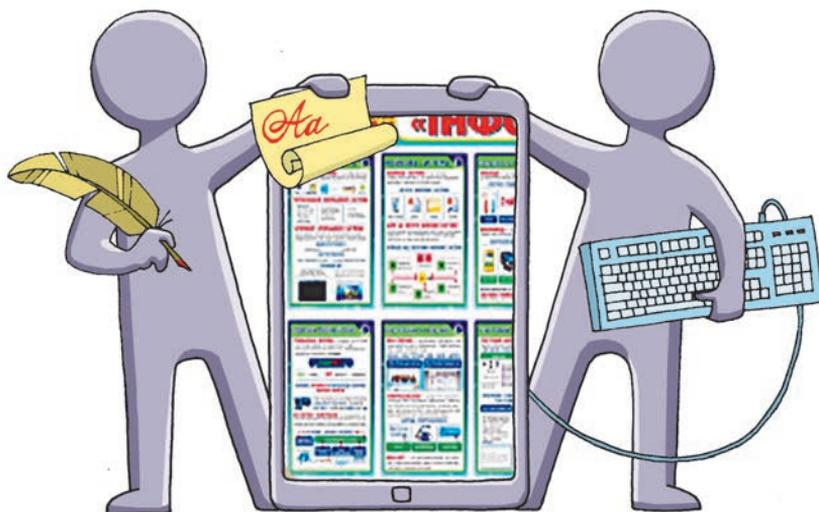
Во время работы за компьютером соблюдайте правила безопасности.

- ▶ **1.** Определите общий объем папки с файлами, подготовленной учителем для архивации. Запустите текстовый процессор. Создайте новый документ. Определите общий объем папки с файлами, подготовленной учителем, и запишите объем папки.
- ▶ **2.** Создайте обычный архив папки. Определите объем архива и введите данные в документ.
- ▶ **3.** Определите степень сжатия архива. Введите данные в документ.
- ▶ **4.** Создайте многотомный архив папки из трех томов примерно одинакового размера. При сжатии задайте размер тома: в три раза меньше объема обычного архива.
- ▶ **5.** Создайте самораспаковывающийся архив папки. Определите объем архива и введите данные в документ. Двойным щелчком извлеките из архива папку, изменив ее имя.
- ▶ **6.** Сохраните файл с именем Практическая работа 3 в своей папке. Завершите работу за компьютером.

Сделайте вывод: как с помощью программы-архиватора создать обычный, многотомный и самораспаковывающийся архивы; какой тип архива вы выберете для пересылки по электронной почте и почему.

РАЗДЕЛ 3

ОБРАБОТКА ТЕКСТОВЫХ ДАННЫХ



§ 6. Списки в текстовом документе

§ 7. Таблицы в текстовом документе

§ 8. Обработка текстового документа, содержащего различные объекты

§ 9. Оформление документов. Структура сложного текстового документа

Практическая работа 4. Создание текстового документа, содержащего объекты разных типов

Практическая работа 5. Структура документа. Автоматическое создание оглавления и предметных указателей

ПОВТОРЯЕМ



Вы уже работали с документом в среде текстового редактора Блокнот и текстового процессора Word. Знаете, что документ может содержать различные *текстовые объекты* и что основными среди них являются символ и абзац.

Каждый объект обладает определенными свойствами. Символ имеет такие свойства, как шрифт, начертание, размер и цвет, абзац — выравнивание, межстрочный интервал, отступы и др. Значения свойств можно изменять.

Вы научились выполнять операции над текстовыми объектами: *редактировать* (удалять, добавлять, перемещать, копировать), *форматировать* (устанавливать шрифт, изменять цвет и начертание символов, расположение абзацев и т. п.); умеете вставлять в текст графические изображения.

1. С какими текстовыми редакторами вы уже работали?
2. Назовите объекты текстового документа.
3. Какие операции можно выполнять над текстовым документом?
4. Какие свойства имеет символ?
5. Какие свойства имеет абзац?
6. Что такое форматирование текстового документа?



В этом разделе вы ознакомитесь с такими объектами текстового документа, как таблица, список, формула, непечатаемые знаки, ссылки, колоннитулы, а также научитесь оформлять сложные текстовые документы.

§ 6. Списки в текстовом документе

Системы обработки текстовых документов различаются по своим возможностям. Текстовые редакторы позволяют выполнять простое форматирование, текстовые процессоры — сложное, включая работу с графическими объектами, содержащимися в документе.

Форматы файлов текстовых документов

Имена файлов текстовых документов, созданных в разных текстовых редакторах, имеют разные расширения. Расширение указывает на формат файла, в котором хранятся данные.



Расширение файла — буквенный код в имени файла, указывающий операционной системе, с помощью какой программы был создан и может быть открыт файл.

Рассмотрим популярные форматы файлов текстовых документов и их характеристики.

Формат `.txt` (*Text* — текст) — сохраняет текст с делением на абзацы без форматирования.

Форматы `.doc` (*Document* — документ) и `.docx` (*Document OpenXML*) — сохраняют текст, рисунки и т. п.; форматы текстового процессора Word.

Формат `.rtf` (*Rich Text Format* — формат обогащенного текста) — сохраняет текст, рисунки, вставленные объекты; позволяет обрабатывать документ в разных программах.

Формат `.pdf` (*Portable Document Format* — формат портативного документа) — сохраненные в этом формате документы не предназначены для редактирования, а их вид не зависит от устройства вывода.

Создание списков

Упорядоченную информацию, например инструкции, перечни, планы мероприятий, удобно представлять в виде списков.



Список — это последовательность абзацев, имеющих общую метку (маркер или номер).

Программа Word позволяет создавать списки разных видов (рис. 6.1).

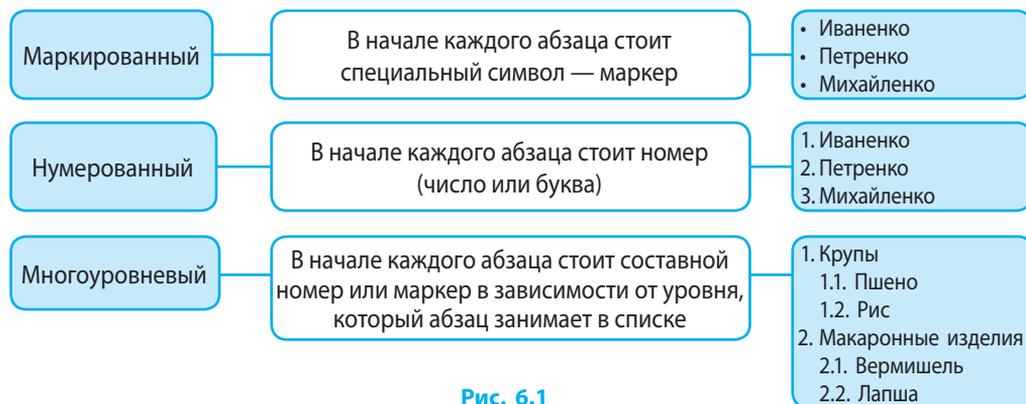


Рис. 6.1

Существует несколько способов создания списков.

Для преобразования введенного текста в список нужно:

- 1) выделить абзацы;
- 2) выбрать на вкладке Главная в группе Абзац значок Маркеры (рис. 6.2, а) или Нумерация (рис. 6.2, б);
- 3) открыть список кнопкой ▼, выбрать значок маркера или формат номера;
- 4) щелкнуть кнопку ОК.

Для создания списка при вводе текста сначала нужно выбрать значок маркера или формат номера — появится метка первого абзаца.

Для автоматического создания списка нужно в начале абзаца ввести символы: звездочку «*» (для маркированного списка) или единицу с точкой либо скобкой: «1.», «1)» (для нумерованного списка). Автоматическое создание (продолжение) списка можно отменить, дважды нажав клавишу Enter.

Для создания многоуровневых списков пользуются инструментом Многоуровневый список (рис. 6.2, в). Кроме номеров и букв, многоуровневая нумерация может содержать различные маркеры, обозначающие разные уровни списка. На рис. 6.3 первый, второй, третий уровни списка выделены красным, зеленым и синим цветом соответственно.

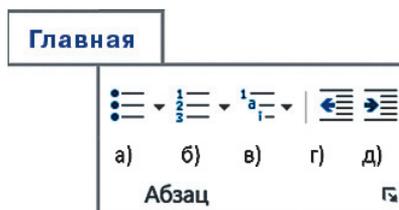


Рис. 6.2

1. Первый уровень	1. Первый уровень
1.1. Второй уровень	А. Второй уровень
1.1.1. Третий уровень	• Третий уровень
1.2. Второй уровень	Б. Второй уровень
1.2.1. Третий уровень	• Третий уровень
2. Первый уровень	2. Первый уровень
2.1. Второй уровень	А. Второй уровень
2.2.1. Третий уровень	• Третий уровень

Рис. 6.3

Программа Word во многоуровневом списке поддерживает девять уровней нумерации. Чтобы перевести абзац на более высокий уровень списка (например, от 1.1. до 1.), следует воспользоваться инструментом Уменьшить отступ (рис. 6.2, з) или комбинацией клавиш Shift + Tab, на более низкий — Увеличить отступ (рис. 6.2, д) или клавишей Tab.

Редактирование списков

В программе Word в списке удобно добавлять или удалять абзацы.

Для **добавления абзаца в список** нужно установить курсор в начале или в конце нужного абзаца и нажать клавишу Enter. Полученный абзац автоматически станет элементом списка, а нумерация соответственно изменится.

Чтобы **отменить нумерацию или маркировку абзаца**, нужно щелкнуть мышью в его пределах и выключить кнопку списка (рис. 6.2, а–в) или удалить номер (маркер). Для этого нужно поместить курсор в начало абзаца и нажать клавишу Backspace. Оставшиеся абзацы списка автоматически перенумеруются. Чтобы **отменить нумерацию нескольких абзацев**, их следует выделить и выключить кнопку списка.

Если нумерация, созданная, например, при копировании списка, не соответствует требованиям пользователя, ее можно изменить, выбрав в контекстном меню номера команду Начать сначала или Продолжить нумерацию.

Форматирование списков

При форматировании списка можно заменить уже существующий маркер другим. Для этого следует щелкнуть по маркеру, открыть список маркировки (рис. 6.4, а) и выбрать нужный. Если такого маркера в стандартной коллекции нет, его можно установить с помощью команды Определить новый маркер.

Чтобы установить **новый формат номеров**, следует выделить нужные абзацы списка или щелкнуть какой-либо номер (если необходимо изменить все номера конкретного уровня) и изменить формат, выбрав команду Определить новый формат номера (рис. 6.4, б) или Определить новый многоуровневый список.

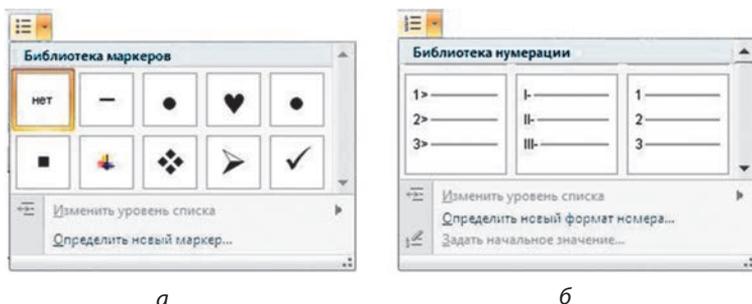


Рис. 6.4

Если при назначении нового формата номера добавить слово *Раздел*, то номера списка 1., 2., 3. будут выглядеть следующим образом: *Раздел 1., Раздел 2., Раздел 3.*

Вопросы для самопроверки



1. Какие списки можно создавать в текстовом процессоре Word?
2. Назовите способы создания списков.
3. Какой список называют многоуровневым?
4. Как добавить в список новый абзац?
5. Как в списке заменить один маркер другим?
6. Как изменить уровень нумерации абзаца в многоуровневом списке?

Упражнение 6



1. Запустите текстовый процессор и создайте новый документ. Установите: размер бумаги — А5, поля — по 2 см, шрифт — *Arial*, размер — 12.
 - 1) Создайте нумерованный список по образцу.
 1. Январь
 2. Февраль
 3. Март
 4. Апрель

2) Создайте маркированный список по образцу.

-  Компьютеры
-  Ноутбуки
-  Смартфоны

3) Создайте многоуровневый список по образцу.

1. Первые блюда
 - 1.1. Борщи
 - 1.1.1. Красный
 - 1.2. Супы
 - 1.2.1. Гороховый
2. Вторые блюда
 - 2.1. Мясные
 - 2.1.1. Вареники с мясом
 - 2.2. Рыбные
 - 2.2.1. Судак в белом соусе

► 2. Сохраните файл с именем Упражнение 6 в своей папке.



Компьютерное тестирование

Выполните тестовое задание 6 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 7. Таблицы в текстовом документе

Для упорядочения и наглядного представления данных в текстовом документе часто используют таблицы. Данные, оформленные в виде таблицы, более удобны для восприятия.

Создание таблицы

Таблица в текстовом документе — это совокупность ячеек, которые могут содержать текст, числа, графические объекты и т.п.

Рассмотрим способы создания таблицы.

Для создания таблицы простой структуры нужно на вкладке Вставка в группе Таблицы выбрать Таблица. Затем на схеме таблицы

(рис. 7.1, а) следует указателем выделить нужное количество строк и столбцов и щелкнуть левой кнопкой мыши. Кроме того, можно выбрать команду Вставить таблицу и в окне Вставка таблицы задать нужное количество столбцов и строк (рис. 7.1, б).

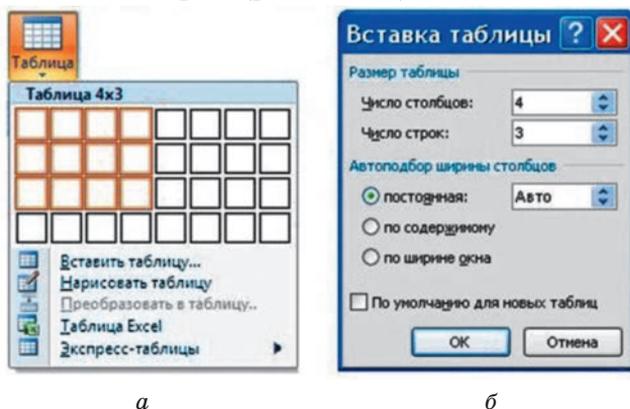


Рис. 7.1

Для создания таблицы произвольной структуры можно выбрать команду Нарисовать таблицу (рис. 7.1, а) и, когда указатель мыши примет вид карандаша, нарисовать таблицу. Лишние границы можно удалить с помощью инструмента Ластик на вкладке Конструктор.

Для преобразования выделенного фрагмента текста в таблицу следует выбрать команду Преобразовать в таблицу (рис. 7.1, а).

Перед началом работы с таблицей ее нужно выделить, например, щелкнув маркер таблицы  над левым верхним углом.

Объекты таблицы удобно выделять с помощью мыши:

Объект	Описание действия
Ячейка	Установить указатель у левой границы ячейки и щелкнуть, когда он примет вид 
Диапазон ячеек	Протянуть указатель с нажатой левой кнопкой мыши от первой ячейки к диагонально противоположной в прямоугольном диапазоне
Столбец	Установить указатель над столбцом и щелкнуть, когда он примет вид 
Строка	Установить указатель слева от строки и щелкнуть, когда он примет вид 
Смежные объекты	Протянуть указатель с нажатой левой кнопкой мыши по нужным объектам
Несмежные объекты	Удерживая клавишу Ctrl, последовательно выделить с помощью мыши нужные объекты

После того как выбранные объекты таблицы будут выделены цветом, их можно удалять, копировать, перемещать и форматировать. Чтобы снять выделение, нужно щелкнуть в любом месте страницы.

Редактирование и форматирование таблицы

Готовую таблицу в Word можно редактировать: изменять содержимое ячеек, их размеры, вставлять, удалять строки и столбцы, перемещать границы ячейки и т. д. По умолчанию ячейки имеют стандартную ширину и высоту. Для изменения высоты строки или ширины столбца можно перетянуть соответствующую границу.

Для **перемещения** нужной **границы** следует навести на нее указатель, затем, когда он примет вид двусторонней стрелки (рис. 7.2), нажать левую кнопку мыши и переместить границу (она отмечается пунктирной линией).

Если перетянуть границу выделенной ячейки, то изменятся размеры только данной ячейки и смежной с ней по этой границе.

Для **изменения размера одного столбца или строки** (без изменения размеров других) нужно перетянуть соответствующий маркер на линейке (рис. 7.3). Чтобы эти маркеры появились, нужно щелкнуть внутри таблицы.

Для **перемещения курсора по ячейкам** нужно щелкнуть ячейку и воспользоваться клавишами управления курсором или клавишей Tab.

Для **добавления новой строки** курсор следует установить в нижнюю правую ячейку и нажать клавишу Tab.

Выбрать общий стиль таблицы и выполнить форматирование ее объектов (изменить направление текста, объединить или разделить ячейки) можно с помощью контекстного меню ячейки или вкладки Конструктор.

Для форматирования содержимого ячейки используются стандартные средства Word.

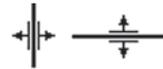


Рис. 7.2

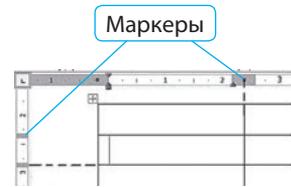


Рис. 7.3

Вопросы для самопроверки



1. Назовите способы вставки таблицы в документ.
2. Как изменить размеры столбца или строки таблицы?
3. Какие объекты таблицы можно выделить с помощью мыши?

4. Как разделить одну ячейку таблицы на несколько?
5. Как объединить несколько ячеек таблицы?
6. Как выбрать стиль таблицы?

Упражнение 7



- ▶ Создать таблицу успеваемости учащихся по нескольким предметам.
- 1) Загрузите текстовый процессор Word. Установите: размер бумаги — А5, поля — по 2 см, шрифт — *Arial*, размер — 12.
 - 2) Создайте таблицу из семи столбцов и шести строк. Объедините ячейки, измените высоту строк по образцу.

№ п/п	Фамилия и имя	Предметы				
		Биоло- гия	Геогра- фия	Инфор- матика	Исто- рия	Физика

- 3) Введите заголовки и отформатируйте по приведенному выше образцу.
- 4) Введите в таблицу свои данные и данные еще троих одноклассников. Пронумеруйте строки, заполните таблицу произвольными оценками. Сохраните файл с именем Упражнение 7 в своей папке.

Компьютерное тестирование



Выполните тестовое задание 7 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 8. Обработка текстового документа, содержащего различные объекты

Документ Word, помимо списков и таблиц, может содержать другие текстовые объекты. Рассмотрим работу с этими объектами подробнее.

Работа с колонками

В научно-популярных статьях, изданиях технического и справочного характера текст обычно размещают в колонках, что придает материалу более компактную форму.

Для размещения текста в колонках следует:

- 1) выделить фрагмент текста;
- 2) на вкладке Разметка страницы в группе Параметры страницы развернуть список Колонки;
- 3) выбрать нужное количество колонок.

Свойства колонок можно настроить с помощью команды Другие колонки.

Работа с символами и формулами

При редактировании и форматировании документов можно использовать символы, отсутствующие на клавиатуре, и непечатаемые знаки.

Чтобы вставить символы, которых нет на клавиатуре (например, символ «§»), нужно на вкладке Вставка в группе Символы выбрать вкладку Символ (рис. 8.1). Если нужного символа в коллекции нет, его следует выбрать в окне Другие символы и вставить. В этом окне также можно определить код символа или вставить символ по значению кода.

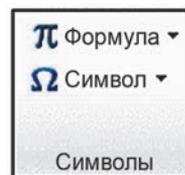


Рис. 8.1

Непечатаемыми знаками называют специальные служебные знаки разметки страницы, которые обозначают места нажатия клавиш Пробел, Tab, Enter и т.п. и не выводятся на бумаге при печати.

Для отображения или скрытия непечатаемых знаков служит кнопка ¶ на вкладке Главная в поле Абзац.

Режим отображения непечатаемых знаков позволяет быстро устранить недостатки форматирования с помощью клавиш:

Непечатаемый знак	Название знака	Клавиша / комбинация клавиш
·	Пробел	Пробел
¶	Конец абзаца	Enter
↵	Разрыв строки	Shift + Enter
⏏	Мягкий перенос	Ctrl + «-»
°	Неразрывный пробел	Ctrl + Shift + Пробел

➤ Если включить режим отображения непечатаемых знаков, текст будет выглядеть так:

Между · словами · появляются · точки, · а · также · знаки · абзаца · и · другие · символы, · которые · ранее · не · отображались. ¶

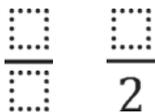


Рис. 8.2

В Word можно работать с формулами. Чтобы вставить **готовую формулу**, следует установить курсор в нужное место, на вкладке Вставка в группе Символы щелкнуть стрелку ▼ (рядом с надписью Формула) и выбрать нужную формулу.

Для **создания новой формулы** следует выбрать команду Вставить новую формулу, а на вкладке Конструктор — нужную структуру (например, дробь) или символ (рис. 8.2).

Графические объекты в текстовом документе

Как вам известно, в документ можно вставлять графические объекты — рисунки, фотографии, схемы и т.п. Объекты могут быть созданы как средствами Word, так и в других программах.

Готовые изображения можно вставить в документ с помощью команд контекстного меню или через буфер обмена.

Для **вставки рисунка** нужно:

- 1) на вкладке Вставка в группе Иллюстрации выбрать Рисунок;
- 2) в диалоговом окне выбрать нужный рисунок;
- 3) щелкнуть кнопку Вставить.

Если вставленный рисунок требует дальнейшего форматирования (изменения размеров, настройки отображения границ, изменения яркости и т. п.), его нужно выделить, щелкнув левой кнопкой мыши. Если выделить рисунок не удастся, следует выбрать команды меню Главная → Редактирование → Выделить → Выбор объектов.

Вокруг выделенного изображения появится рамка с маркерами. С их помощью рисунок можно уменьшать, увеличивать, сжимать, поворачивать и т. п. Изменить размеры рисунка можно путем перетягивания маркеров границ.

Для **перемещения рисунка** нужно установить на рисунке указатель и перетянуть с нажатой левой кнопкой мыши или щелкнуть на рисунке и воспользоваться клавишами перемещения курсора.

Можно задать **обтекание рисунка текстом** (вокруг рамки, сверху и снизу и т. п.). Для этого нужно выделить рисунок и выбрать способ обтекания на вкладке Формат в списке Обтекание текстом.

Над рамкой выделенного рисунка располагается маркер, с помощью которого рисунок можно **поворачивать вокруг центра**.

Чтобы **обрезать рисунок**, нужно выбрать команду Формат → Обрезка, затем, перемещая маркер, обрезать края рисунка и щелкнуть вне его границ.

Вопросы для самопроверки



1. Как расположить текст в несколько колонок?
2. Как включить режим отображения непечатаемых знаков?
3. Как вставить формулу в текстовый документ?
4. Как вставить рисунок в текстовый документ?
5. Как задать способы обтекания рисунка текстом?
6. Как обрезать рисунок?

Упражнение 8



▶▶ Создать текстовый документ с использованием различных объектов.

1) Запустите текстовый процессор и создайте новый документ. Найдите в Интернете небольшое стихотворение А. С. Пушкина и скопируйте его в документ. Разместите текст в две колонки.

2) Вставьте в текстовый документ формулу $v_1 = \frac{s_1}{t}$.



3) Скопируйте в буфер обмена изображение экрана (страницу с формулой) с помощью клавиши PrintScreen. Вставьте рисунок из буфера обмена в документ и обрежьте так, чтобы осталась только формула. Задайте для рисунка формат обтекания Перед текстом, поверните на произвольный угол и разместите под текстом стихотворения. Сохраните файл с именем Упражнение 8 в своей папке.



Компьютерное тестирование



Выполните тестовое задание 8 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 9. Оформление документов. Структура сложного текстового документа

Форматирование сложных документов занимает много времени. Чтобы ускорить этот процесс, применяют стилевое оформление.

Стилевое оформление документа



Стиль — именованная совокупность параметров форматирования объектов документа, определяющих его внешний вид.

До применения стиля:

Первый в Европе компьютер был создан в Киеве под руководством академика С. А. Лебедева.

ГЛАВНАЯ



AaBbVgГз 3. AaB 3.1. AaB AaBbB
Выделение ¶ Заголов... Заголов... ¶ Заголов...
Стили



После применения стиля
Выделенная цитата

Первый в Европе компьютер был создан в Киеве под руководством академика С. А. Лебедева.

Рис. 9.1

Для того чтобы задать стилевое оформление выделенного объекта, нужно выбрать стиль на вкладке Главная в группе Стили (рис. 9.1).

Для редактирования стиля нужно:

- 1) открыть контекстное меню стиля (щелкнуть окошко правой кнопкой мыши);
- 2) выбрать команду Изменить;
- 3) в окне Изменение стиля нажать кнопку Формат;
- 4) настроить параметры для объектов заданного стиля.

После этого все объекты, для которых задан данный стиль, приобретут новый вид.

Шаблоны документов

Довольно часто приходится создавать документы с похожей структурой. Для этого используют шаблоны.



Шаблон — это документ, хранящийся в файле особого формата и использующийся в качестве основы для создания подобных документов.

В Word файлы шаблонов имеют расширение .dot (.dotx). По умолчанию при создании нового документа открывается шаблон Normal.dotx.

Пользователь может создать новый документ на основе данного шаблона или выбрать другой шаблон с помощью команды Office → Создать.

Если пользователь решил сохранить как шаблон документ, оформленный по своему усмотрению, то нужно щелкнуть Office, указать место сохранения, название и выбрать тип файла — Шаблон Word.

Сохраненный шаблон, в отличие от файла документа, будет защищен от случайных изменений, пока пользователь не сохранит другой документ с таким же именем в формате шаблона.

Разделы

Если для страниц документа требуется установить различные значения параметров, документ нужно разбить на разделы, вставив соответствующие разрывы разделов (рис. 9.2).

Чтобы **вставить разрыв раздела**, нужно:

- 1) щелкнуть на вкладке Разметка страницы в группе Параметры страницы элемент Разрывы;
- 2) выбрать тип разрыва раздела.

Теперь в документе для отдельных разделов можно установить разные параметры поля, размер и ориентацию бумаги, источник бумаги для принтера, границы страницы и т. п. Параметры форматирования страницы распространяются на все страницы раздела.

Если удалить разрыв раздела, удалится и форматирование раздела, который расположен перед разрывом, он станет частью следующего раздела и примет его формат.

Чтобы **удалить разрыв раздела** нужно:

- 1) щелкнуть в правом нижнем углу экрана кнопку Черновик (рис. 9.3) — разрывы разделов будут отображаться в виде двойных пунктирных линий;
- 2) выделить разрыв раздела;
- 3) нажать клавишу Delete.

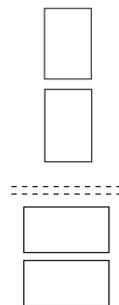


Рис. 9.2

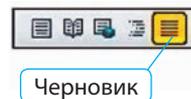


Рис. 9.3

Структура документа

Работая с большим по объему документом, сложно находить в тексте нужные части, если документ не структурирован.



Структура документа — иерархическая схема размещения частей документа.



Рис. 9.4

Для организации документов большого объема, состоящего из оглавления, разделов и подразделов разных уровней (например, учебника), используется **режим структуры** (рис. 9.4).

Режим структуры в программе Word позволяет представлять документ в виде, отображающем заголовки разделов, параграфов и т.п. В этом режиме можно изменять уровни заголовков, копировать и перемещать параграфы и разделы путем перемещения соответствующих заголовков.

Для **включения режима структуры** нужно на вкладке Вид выбрать команду Структура.

Колонтитулы

В документах со сложной структурой можно вставить колонтитулы, в которых указывают название книги, раздела, номера страниц и т. п.



Колонтитулы — сообщения, которые содержатся в верхнем и нижнем полях страницы документа и отображаются на всех его страницах.

Для **установки колонтитулов** необходимо:

- 1) дважды щелкнуть в области верхнего (нижнего) края страницы — откроются поля колонтитулов и вкладка Конструктор (рис. 9.5);
- 2) ввести текст в поле колонтитула.

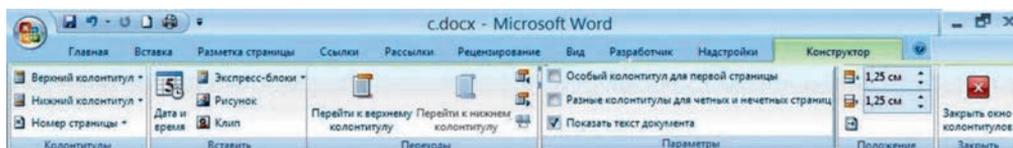


Рис. 9.5

Инструменты вкладки Конструктор позволяют автоматически вставлять в колонтитулы *номера страниц, название документа, данные об авторе, текущей дате* и т. п.

Чтобы после редактирования колонтитула вернуться к основному тексту документа, на нем нужно дважды щелкнуть.

Чтобы удалить колонтитул, нужно на нем дважды щелкнуть, выделить содержимое и нажать клавишу Delete.

Ссылка

В программе Word, наряду с подготовкой документов для печати на бумаге, можно разрабатывать и электронные документы, которые, в частности, содержат ссылки.



Ссылка (гиперссылка) — это часть документа (рисунок, текст и т. п.), щелчком по которому осуществляется переход в определенное место документа или в другой файл, находящийся на компьютере пользователя или в Интернете.

Для **перехода по ссылке** нужно при нажатой клавише Ctrl щелкнуть на ссылке левой кнопкой мыши.

Для **создания ссылки** нужно:

- 1) выделить фрагмент текста или графический объект;
- 2) на вкладке Вставка выбрать группу Связи;
- 3) выбрать элемент Гиперссылка (рис. 9.6) и указать, на какой объект создается ссылка (рис. 9.7).

Прежде чем создать ссылку на какое-либо место в документе, в нем необходимо создать закладку, на которую будет указывать ссылка.

Для **создания закладки** необходимо:

- 1) выделить объект или установить курсор в нужное место документа;
- 2) выбрать команду Вставка → Связи → Закладка;
- 3) ввести название новой закладки, нажать кнопку Добавить в окне со списком закладок.

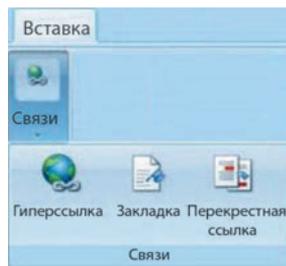


Рис. 9.6

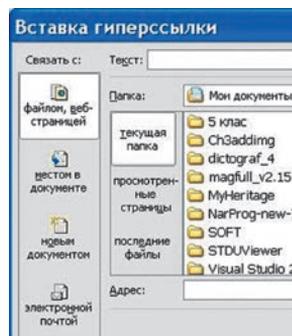


Рис. 9.7

Создание оглавления и предметного указателя

После внесения изменений в текст документа для **автоматического создания оглавления** нужно:

- 1) выбрать место вставки оглавления (в начале или в конце документа);
- 2) щелкнуть на вкладке Ссылки в группе Оглавление элемент Оглавление;
- 3) выбрать стиль оглавления и нажать клавишу Enter — в указанном месте появится оглавление.

Для **обновления оглавления** нужно:

- 1) на вкладке Ссылки в группе Оглавление щелкнуть Обновить таблицу;
- 2) выбрать команду Обновить только номера страниц или Обновить полностью.

Для **удаления оглавления** нужно:

- 1) на вкладке Ссылки в группе Оглавление щелкнуть Обновить таблицу;
- 2) выбрать команду Удалить оглавление.

В конце документа обычно размещают предметный указатель, который удобно использовать для быстрого поиска терминов.



Предметный указатель — список слов с указанием номеров страниц, на которых они упоминаются.

Для **создания предметного указателя** нужно:

- 1) выделить в тексте нужный термин (слово или фразу);
- 2) выбрать команду Ссылки → Предметный указатель → Пометить элемент;
- 3) настроить параметры в окне Определение элемента указателя:
 - при необходимости изменить или заменить текст в поле основной;
 - чтобы элемент указывал на несколько терминов (например, архивный файл, сжатый файл, архив), в поле дополнительный вписать через двоеточие другие термины;
 - нажать кнопку Пометить или Пометить все (чтобы найти и отметить в документе все термины);
- 4) установить курсор в нужное место документа и выбрать команду Ссылки → Предметный указатель.

Для **удаления элемента указателя** нужно отобразить непечатаемые знаки текста (Главная → ¶) и удалить обозначение в фигурных скобках (возле элемента указателя).

Работа со сложным текстовым документом предполагает определенную последовательность действий.

Рассмотрим **алгоритм работы со сложным текстовым документом**.

- 1) *Набор и сохранение* — введение текста с клавиатуры и сохранение в виде файла на носителе.
- 2) *Редактирование* — исправление ошибок, вставка-удаление фрагментов текста, других объектов, изменение их расположения.
- 3) *Форматирование* — придание документу нужного вида: изменение свойств объектов, настройка стилей и т. п.
- 4) *Подготовка к печати* — оформление заголовков, разбиение на страницы, их нумерация, создание оглавления.
- 5) *Печать* — вывод на бумагу всех или выбранных страниц.

В программе Word при работе со сложным документом окно текстового документа можно разделить на несколько областей: вкладка Вид → Разделить (группа Окно).

Документ будет виден в каждой части. Особенно удобно использовать разделение окна при копировании (переносе) фрагментов из одной части документа в другую.

Вопросы для самопроверки



1. Как изменить стиль абзаца?
2. Что такое шаблон документа?
3. Для чего используют разрыв раздела; режим структуры документа?
4. Что такое колонтитул? Как его создать?
5. Как создать ссылку?
6. Как создать автоматическое оглавление документа; указатель?

Упражнение 9



- ▶▶ Создать и оформить текстовый документ с использованием разных стилей.
- 1) Запустите текстовый процессор Word, создайте новый документ. Найдите в Интернете и скопируйте в документ одно из действий пьесы Н. Гоголя «Ревизор». Для каждого явления установите разное стилевое оформление по своему усмотрению.
 - 2) Введите название произведения (если его нет) в начале текста и установите для него стиль Название. В конце документа запишите имя и фамилию автора произведения, задайте стиль Строгий.
 - 3) Установите верхний колонтитул с текущей датой. Добавьте в нижнем колонтитуле номера страниц, к названию произведения — ссылку на веб-страницу с заданным произведением. Сохраните документ в формате шаблона с именем Упражнение 9 в своей папке.

Компьютерное тестирование



Выполните тестовое задание 9 с автоматической проверкой на сайте interactive.ranok.com.ua.

Практическая работа 4



Создание текстового документа, содержащего объекты разных типов

Задание: создать текстовый документ (карточку для урока физики) с использованием объектов разных типов.

Оборудование: компьютер с установленным текстовым процессором Word, подключенный к сети Интернет.

Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

- ▶ 1. Запустите текстовый процессор и создайте новый документ. Установите: формат страницы — А4, поля — по 2 см, ориентация — книжная, интервалы до и после абзацев — 0 см, междустрочный интервал — одинарный, шрифт — *Times New Roman*, размер — 14, начертание — полужирный курсив.
- ▶ 2. Введите в первом абзаце номер параграфа — § 1, во втором абзаце — заголовок «Равномерное прямолинейное движение», задайте выравнивание — по центру.
- ▶ 3. Найдите в Интернете изображение гепарда, скопируйте рисунок и вставьте в документ под заголовком.
- ▶ 4. Задайте для рисунка формат обтекания текстом Сверху и снизу, обрежьте рисунок так, чтобы осталось только изображение гепарда.
- ▶ 5. Пропорционально измените размеры рисунка таким образом, чтобы по ширине он занял всю страницу (от поля до поля).
- ▶ 6. Установите для заголовка размер шрифта — 26, отформатируйте текст, чтобы заголовок поместился в одной строке.
- ▶ 7. Найдите в Интернете данные о скоростях движения пешехода, гепарда и сокола и введите в документ в отдельных абзацах.
- ▶ 8. Оформите введенные данные в виде нумерованного списка.
- ▶ 9. Вставьте в документ под списком таблицу: пять строк, три столбца.
- ▶ 10. Объедините ячейки верхней строки таблицы и введите название таблицы «Основные величины».
- ▶ 11. Заполните таблицу по образцу, введите необходимые формулы.

§1

Равномерное прямолинейное движение



1. Скорость пешехода 000 км/час
2. Скорость гепарда 000 км/час
3. Скорость сокола 000 км/час

Основные величины		
Название	Обозначение	Формула
Пройденный путь	s	$s = v \cdot t$
Скорость движения	v	$v = \frac{s}{t}$
Время движения	t	$t = \frac{s}{v}$

- ▶ 12. Оформите таблицу по образцу, расположите надписи по центру, переместите границы таблицы.
- ▶ 13. Сохраните файл с именем Практическая работа 4 в своей папке. Завершите работу за компьютером.

Сделайте вывод: как создать текстовый документ, содержащий объекты разных типов; какие объекты текстового документа вы использовали при создании карточки.

Практическая работа 5



Структура документа. Автоматическое создание оглавления и указателей

Задание: создать текстовый документ, состоящий из нескольких разделов, выполнить форматирование его объектов, создать автоматическое оглавление и указатель.

Оборудование: компьютер с установленным текстовым процессором Word, подключенный к сети Интернет.

Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

- ▶ 1. Запустите текстовый процессор Word, создайте новый документ. Найдите в Интернете три-четыре стихотворения одного из украинских поэтов и скопируйте их в документ. Если текст был расположен в таблице, выделите всю таблицу и выполните: Макет → Данные → Преобразовать в текст (разделитель — знак абзаца).
- ▶ 2. Отформатируйте текст так, чтобы каждое стихотворение занимало один абзац. Удалите пустые страницы, поместите название стихотворения в отдельном абзаце.
- ▶ 3. Введите в первом абзаце название сборника, например «Стихотворения М. Ю. Лермонтова».
- ▶ 4. Установите для документа стиль Обычный, для названий стихотворений — стиль Заголовки 1, для названия сборника — стиль Название.
- ▶ 5. Найдите в Интернете портрет автора стихов и скопируйте в документ после названия сборника. При необходимости отформатируйте рисунок.
- ▶ 6. Под портретом автора создайте автоматическое оглавление.
- ▶ 7. Переименуйте созданное Оглавление в Оглавление сборника.
- ▶ 8. Разделите документ на разделы: после портрета и последней строки каждого стихотворения вставьте разрывы Текущая страница, а после оглавления — разрыв Следующая страница.
- ▶ 9. Установите для всего документа межстрочный интервал — полупуторный. Выполните обновление оглавления.
- ▶ 10. Установите в нижнем колонтитуле нумерацию страниц.
- ▶ 11. Выберите два-три повторяющихся в стихах слова и создайте в конце документа указатель.
- ▶ 12. Добавьте к названию сборника ссылку на страницу в Интернете, с которой копировались тексты, и проверьте ее работу.
- ▶ 13. Сохраните файл с именем Практическая работа 5 в своей папке. Завершите работу за компьютером.

Сделайте вывод: каковы особенности работы со сложными текстовыми документами; как создать автоматическое оглавление и предметный указатель.

РАЗДЕЛ 4

ОБРАБОТКА ОБЪЕКТОВ МУЛЬТИМЕДИА



§ 10. Форматы видео- и аудиофайлов

§ 11. Программное обеспечение для обработки объектов мультимедиа

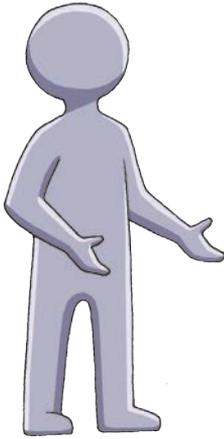
§ 12. Создание аудио- и видеофрагментов

§ 13. Обработка, сохранение и размещение видеоклипа

Практическая работа 6. Создание видеоклипа. Добавление видеоэффектов, настройка временных параметров аудио- и видеоряда

Практическая работа 7. Размещение аудио- и видеоматериалов в Интернете

ПОВТОРЯЕМ



Как вы знаете, под понятием *мультимедиа* понимают совокупность разных по способу представления данных. Объектами мультимедиа являются *текст, графическое изображение, звук, видео, анимация.*

Среди устройств, относящихся к мультимедийному оборудованию, можно назвать мобильные телефоны, мультимедийные доски и проекторы, видеокамеры, акустические системы и т. п. С каждым годом количество таких устройств стремительно растет.

Мультимедиа применяют в различных сферах: кино, телевидении, рекламе, образовании, медицине и т. п.

Мультимедийные данные обрабатываются с помощью различных программ и хранятся в файлах специальных форматов. Вы уже научились пользоваться некоторыми из них, например, программой для создания презентаций, мультимедийным проигрывателем, графическим редактором и т. п.

1. Что такое мультимедиа?
2. Что является объектами мультимедиа?
3. Приведите примеры мультимедийных устройств.
4. Назовите сферы применения мультимедиа.
5. Для чего предназначен графический редактор?
6. Какая программа предназначена для просмотра видео?



В этом разделе вы ознакомитесь с такими объектами мультимедиа, как аудио и видео, научитесь создавать видеоклипы, узнаете способы размещения видеоматериалов в Интернете.

§ 10. Форматы видео- и аудиофайлов

Более 90 % информации человек получает с помощью зрения и слуха, поэтому видео- и аудиоматериалы являются для него наиболее информативными.

Форматы видеофайлов

Технологически фильм — это набор кадров — последовательных изображений движущихся объектов, отснятых с определенной частотой. При просмотре кадров, быстро сменяющихся друг друга, у человека создается впечатление, что изображенные объекты непрерывно движутся. Как известно, на разных этапах развития кинематографа такие кадры фильма размещали на специальной пленке, в эпоху цифровых технологий — в видеофайле.



Видеофайл — это набор статических изображений, которые при просмотре видеофайла поочередно с определенной частотой выводятся на экран.

» В пленочном кино для хранения всех кадров фильма, например, продолжительностью один час и частотой 24 кадра в секунду, потребуются километры пленки. Для хранения такого же количества кадров несжатого видео в видеофайле понадобятся сотни гигабайтов памяти.

Сейчас полнометражный фильм умещается на одном компакт-диске или флеш-накопителе благодаря современным технологиям «сжатия»: объем таких видеофайлов в сотни раз меньше, чем при сохранении в покадровом режиме. Для сжатия и воспроизведения видео используются специальные программы — **кодеки** (от слов *кодировать* — *декодировать*).

Формат видеофайла определяется методом его сжатия. Большинство видеофайлов могут содержать изображения, аудио и видео разных форматов, анимацию и т. п. Такие файлы называют **медиаконтейнерами**.

Рассмотрим форматы некоторых видеофайлов.

AVI — медиаконтейнер, изначально разработанный для ОС Windows. Воспроизводится на смартфонах, коммуникаторах и т. п.

WMV — формат компании Microsoft. Может содержать аудио и видеоданные. Воспроизводится на плеерах MPlayer, Windows Media Player и др.

MOV — формат компании Apple для QuickTime-плеера. Может содержать видео, анимацию, графику. Поддерживает любые аудио- и видеокодеки.

MPG — формат с небольшими потерями при сжатии. Совместим со многими программами и устройствами.

MP4 — формат для хранения видео, аудио и некоторых видов анимации. Поддерживает многоканальный звук. Чаще всего используется в портативных устройствах, для передачи видеофайлов в Интернете.

ASF — формат компании Microsoft. Используется для передачи видео в Интернете. Поддерживает практически все видеокодеки.

MKV (Matroska) — формат, по возможностям превосходящий AVI, может хранить информацию о титрах, навигационных меню и т. п. В контейнер MKV можно упаковывать видео и аудио различных форматов.

WebM — открытый формат компании Google для веб-медиа. Упрощенный аналог MKV с функциями, необходимыми для сети. Поддерживается большинством веб-браузеров и медиаплееров.

Форматы аудиофайлов

Люди получили возможность хранить и воспроизводить звуковые сообщения, когда научились записывать звуковой «след» на грампластинках, магнитофонных лентах и т. п. Такая звукозапись называется **аналоговой**.



Цифровая звукозапись — представление звука в виде двоичного кода.

Для хранения оцифрованного звука в аудиофайлах используют разные технологии сжатия.

Рассмотрим форматы некоторых аудиофайлов:

Пиктограмма	Аудиофайл	Особенности
	WAV, AIFF	Без сжатия
	APE, FLAC	Сжатие без потерь качества
	MP3, Ogg	Сжатие с потерями качества за счет исключения частот, которые не воспринимаются человеком

Существует много других форматов видео- и аудиофайлов. Они создаются для различных устройств и имеют разное назначение.

Вопросы для самопроверки



1. Что такое фильм?
2. Зачем «сжимают» цифровое видео?
3. Назовите известные вам форматы видеофайлов.
4. Что такое цифровая звукозапись?
5. Назовите известные вам форматы аудиофайлов.
6. Чем объясняется большое количество видео- и аудиоформатов?

Упражнение 10



▶ 1. Воспроизведите на компьютере музыкальный аудиофайл. Запишите в тетради, какие данные (кроме звуковых) отображаются медиаплеером при воспроизведении музыки.



▶ 2. Найдите в Интернете видеофильм для скачивания. Запишите в тетради формат и размер видеофайла, продолжительность фильма.



▶ 3. Объясните, почему именно этот формат видеофайла используется для передачи видео в Интернете.

Компьютерное тестирование



Выполните тестовое задание 10 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 11. Программное обеспечение для обработки объектов мультимедиа

Для работы с объектами мультимедиа используют различные программы. Некоторые из них предназначены для импорта или воспроизведения видео и/или аудио, другие — для редактирования мультимедийных данных, изменения форматов видео- и аудиофайлов и т. п.

Программы для обработки аудиоданных

Часто звуковые данные, полученные с внешних устройств, например, микрофона, электрогитары, синтезатора и т. п., нуждаются в обработке на компьютере.



Процесс преобразования звукового сигнала, поступающего с внешних источников, в цифровой формат и записи в аудиофайл называют **захватом аудио**.

Чтобы захватить аудио, нужно подключить к компьютеру аудиоустройство, например микрофон, и запустить программу для записи звука.

» В ОС Windows 7–10 стандартную программу Звукозапись можно запустить с помощью команды Пуск → Все программы → Стандартные → → Звукозапись (рис. 11.1). Если во время записи звука нажать кнопку остановки записи, откроется окно с предложением сохранить файл.

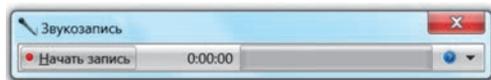


Рис. 11.1

Записанный звук сохранится в формате WAV, и затем процесс звукозаписи можно продолжить с того же места.

» В некоторых программах, например, в стандартной программе Звукозапись ОС Windows XP (рис. 11.2), продолжительность записываемого фрагмента ограничена. Но к открытому звуковому файлу подобные фрагменты можно дописывать много раз.

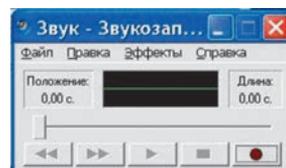


Рис. 11.2



Аудиоредактор — программа для обработки аудиофайлов.

Установив на компьютер аудиоредактор, пользователь может записывать аудио с любых источников, накладывать спецэффекты, обрезать аудиофрагменты, создавать различные звуки и т. п.

» Среди популярных аудиоредакторов можно назвать Audacity, Sony SoundForge, Adobe Audition и т. д. Для воспроизведения аудиоданных (прослушивания звука) используют аудиоплееры, например Winamp, AIMP, JetAudio.

Программы для обработки видеоданных

С помощью компьютера видеоданные можно захватывать, сохранять и т. п. Для этого используют соответствующие устройства и программное обеспечение.



Процесс преобразования видеосигнала с внешнего устройства в цифровой формат и записи в видеофайл называется **захватом видео**.

Рассмотрим захват видео в зависимости от источника и типа видеосигнала.

Захват с аналогового устройства — это запись видео с телевизора, аналоговой видеокамеры, кассетного видеомagnetофона и т. п.

Для оцифровки аналогового видеосигнала используют специальные устройства — видеоадаптеры.

Захват с экрана монитора — это запись всего, что отображается на экране монитора или его части.



Такие программы, как Icecream Screen Recorder, CamStudio и т. п., кроме захвата видео с экрана монитора позволяют произвести захват звука с микрофона или колонок и сохранить видео- и аудиоданные в форматах, поддерживаемых современными медиаплеерами.

Захват с цифровых видеокамер — это запись всего, что фиксирует подключенная к компьютеру цифровая видеокамера.

Большинство цифровых видеоустройств комплектуется специальным программным обеспечением, в том числе и для захвата видео. Но обычно видеоредакторы, среди которых и Windows Movie Maker, снабжены функциями захвата видео с цифровых камер.



Видеоредактор — программа для создания, редактирования и монтажа видеофайлов из отдельных видеофрагментов.



Среди популярных видеоредакторов можно назвать Lightwork, Pinnacle Studio, Sony Vegas Pro и др. (рис. 11.3). В Windows 7–10 можно установить видеоредактор Киностудия, загрузив его из Интернета.

Для воспроизведения аудио и видео, а также для просмотра изображений используют медиаплееры. Один из них — уже знакомый вам Проигрыватель Windows Media.



Рис. 11.3

Средства преобразования аудио- и видеоформатов

Разные устройства и программы выдвигают различные требования к формату данных. Преобразовать аудио- или видеофайлы из одного формата в другой помогают конвертеры.



Конвертер — программа, позволяющая перекодировать медиафайлы в нужный формат.



Чтобы конвертировать видео- или аудиофайл, например, с помощью программы Convertilla, его следует добавить (или перетащить) в рабочее поле конвертера, выбрать нужный формат и нажать кнопку Конвертировать.

Вопросы для самопроверки



1. Что такое захват аудио?
2. Каково назначение аудиоредакторов?
3. Что такое захват видео?
4. С каких устройств можно захватывать видео?
5. Назовите программу для просмотра видеофайлов.
6. Как называются программы для изменения формата видеофайлов?

Упражнение 11



- ▶ Создать аудиофайл с помощью стандартной программы звукозаписи.
- 1) Создайте на рабочем столе папку Медиа. Подключите микрофон (розовое гнездо) и наушники (зеленое гнездо) к компьютеру. Запустите стандартную программу звукозаписи.
 - 2) Нажмите кнопку Запись. Надиктуйте свою фамилию, имя, класс. Остановите запись и сохраните файл с именем Упражнение 11 в папке Медиа. Завершите работу программы, выключите микрофон.
 - 3) Прослушайте запись с помощью медиа- или аудиоплеера.

Компьютерное тестирование



Выполните тестовое задание 11 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 12. Создание аудио- и видеоклипов

Знакомство с Windows Movie Maker

Windows Movie Maker — стандартный видеоредактор некоторых версий ОС Windows для создания, редактирования и монтажа видео. Рассмотрим основные элементы его интерфейса (рис. 12.1).

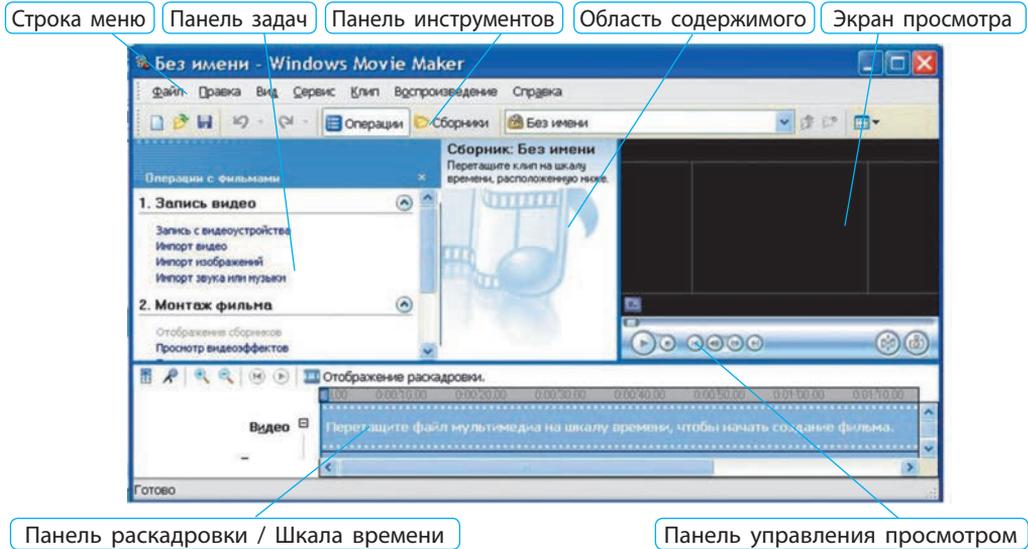


Рис. 12.1

Отображение панелей программы настраивается с помощью меню Вид. Основные действия с объектами выполняются на Панели раскадровки и Шкале времени.

Панель раскадровки используется для просмотра и изменения последовательности фрагментов видеоряда, для просмотра и установки видеоэффектов или видеопереходов (рис. 12.2).



Рис. 12.2

Шкала времени позволяет обрезать фрагменты, настраивать продолжительность переходов между ними, изменять время воспроизведения фрагментов и т. п. (рис. 12.3).

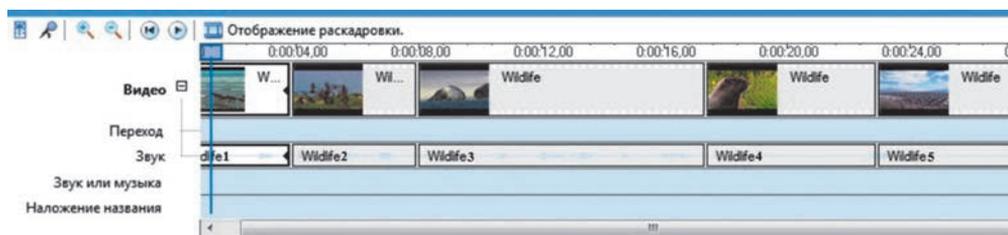


Рис. 12.3

Импорт аудио- и видеоклипов

Процесс создания видеоклипов с помощью Windows Movie Maker состоит из определенной последовательности шагов. Первый шаг работы над созданием клипа — это формирование подборки графических изображений, видео- и аудиоматериалов.



Импорт — добавление к проекту файлов из внешних источников.

Подготовленные файлы можно импортировать путем перетягивания в Область содержимого или с помощью соответствующих команд в меню Файл.

При захвате видео с цифровой видеокамеры (запись с видеоустройства) одновременно происходит сохранение видеофайла в указанном пользователем месте и размещение в Области содержимого.

Графические, видео- и аудиофайлы, импортируемые в Windows Movie Maker, можно увидеть в окне Проводника.

Редактирование видео- и аудиофрагментов

Следующий шаг работы над созданием клипа — редактирование видео- и аудиофрагментов. Для этого видеофайлы нужно перетянуть из Области содержимого на Панель раскладки или Шкалу времени.

Windows Movie Maker позволяет разделять, объединять и обрезать фрагменты.

Для **разделения фрагмента** следует:

- 1) выбрать нужный фрагмент;
- 2) указать место разделения, установив маркер поиска под Окном просмотра (рис. 12.4) или на Шкале времени;
- 3) приостановить воспроизведение и щелкнуть кнопку Разделить.

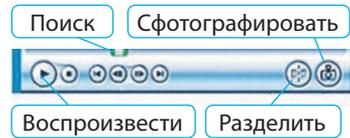


Рис. 12.4

Файл может разделиться на несколько фрагментов. Если фрагменты смежные, их можно объединить на Панели раскадровки или в Области содержимого (см. рис. 12.1). Для этого нужно:

- 1) выделить фрагменты, удерживая клавишу Ctrl;
- 2) в контекстном меню одного из фрагментов выбрать Объединить (рис. 12.5).

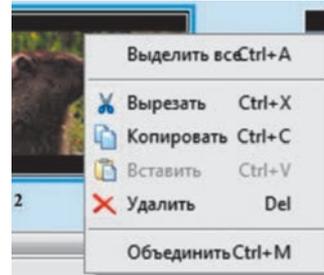


Рис. 12.5

Для **автоматического разделения фрагмента** нужно:

- 1) выбрать фрагмент в Области содержимого;
- 2) выполнить команду Создать клипы в меню Сервис.

Для большинства форматов (кроме WMV и AVI) операция разделения доступна только в ручном режиме.

Чтобы **обрезать начало или конец фрагмента**, следует перетянуть на Шкале времени один из маркеров монтажа, расположенных на краях фрагмента, в нужное место, определив его визуально в Окне просмотра или по временным параметрам на Шкале времени (рис. 12.6).

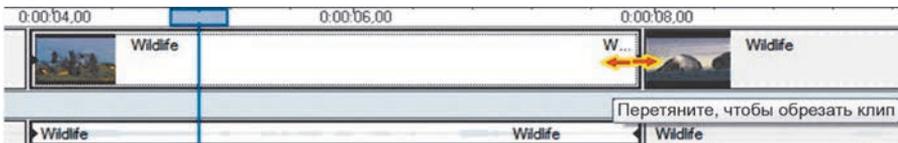


Рис. 12.6

Обрезанный фрагмент можно восстановить до исходного состояния путем перетягивания маркеров монтажа.

Построение аудио- и видеоряда

Следующий шаг работы над созданием клипа — оформление видеоряда и аудиоряда. Шкала времени Windows Movie Maker содержит две звуковые дорожки. Обычно видеосегмент имеет звуковое сопровождение, которое размещается на дорожке Звук.

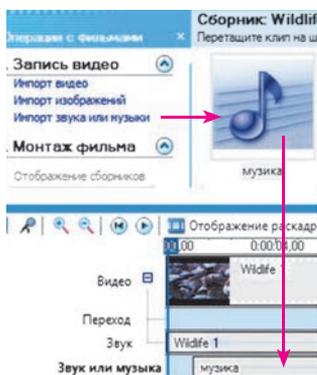


Рис. 12.7

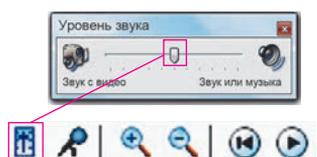


Рис. 12.8

Созданный файл звукового комментария сохранится в автоматически созданной папке или в папке, указанной пользователем.

Видеоряд будущего клипа образуют импортированные, отредактированные, размещенные на Шкале времени или Панели раскадровки объекты.

Текстовые фрагменты (названия, титры и т. п.) обычно размещают в начале или в конце фильма на определенном видефрагменте или изображении, а также перед или после него. Время воспроизведения фрагментов корректируется маркерами на Шкале времени, как и время воспроизведения переходов или видеоэффектов. Текст можно редактировать, менять шрифт, размер, начертание и цвет символов.

Импорт изображений не отличается от импорта видео или звука. Импортировать изображение можно двумя способами: перетащить графический файл из места размещения в Область содержимого; выбрать команду Импорт изображений на Панели задач. Затем следует перетянуть изображение из Области содержимого на Панель раскадровки или Шкалу времени.

Чтобы сохранить изображение одного кадра, нужно осуществить **захват кадра**. Для этого следует приостановить просмотр на определенном кадре или найти его маркером поиска и щелкнуть кнопку

Аудиофрагмент на дорожке Звук можно **обрезать**, **перемещать** и **удалять** только вместе с соответствующим видефрагментом (рис. 12.7).

Импортированные аудиофайлы можно **перетягивать** из Области содержимого на дорожку Звук или музыка. Если перетянуть на звуковую дорожку видефрагмент, то на дорожке останется только аудиосопровождение.

Редактирование звука на дорожке Звук или музыка происходит независимо от видефрагмента.

Относительную громкость обеих звуковых дорожек можно регулировать кнопкой Установить уровень звука (рис. 12.8).

Windows Movie Maker позволяет захватывать **звук с микрофона**. Для этого нужно подключить микрофон, нажать кнопку на Шкала времени комментария и, пользуясь подсказками, добавить звуковой комментарий.

Сфотографировать (см. рис. 12.4). Затем в окне Проводника указать параметры сохранения графического файла, который потом автоматически появится в Области содержимого.

Вопросы для самопроверки



1. Что отображает Панель раскадровки?
2. Какие возможности предоставляет Шкала времени?
3. Как обрезать видеофрагмент?
4. Можно ли восстановить обрезанный видеофрагмент?
5. Как создать собственное звуковое сопровождение видеофильма?
6. Как осуществить захват кадра?

Упражнение 12



▶▶ Создать видеоклип со звуковым сопровождением.



- 1) Найдите в Интернете и скачайте файл для звукового сопровождения видеоклипа в папку, указанную учителем. Скопируйте в нее видеофайлы, подготовленные учителем, или собственные.
- 2) Запустите видеоредактор, импортируйте в него подготовленные аудио- и видеофайлы. Просмотрите видеофайлы.
- 3) Перетащите импортированные видеофайлы на Панель раскадровки, а аудиофайл — на дорожку Звук или музыка, расположенную на Шкале времени. Сохраните проект (Файл → Сохранить проект) с именем Упражнение 12 в своей папке.

Компьютерное тестирование



Выполните тестовое задание 12 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 13. Обработка, сохранение и размещение видеоклипа

Чтобы при просмотре один фрагмент видеоклипа плавно сменялся другим, используют переходы; для настройки определенного стиля показа видеофрагмента добавляют видеоэффекты, которые помогают лучше реализовать замысел автора.

Добавление в видеоклип видеоэффектов и настройка переходов между его фрагментами

Добавление **видеоэффектов** позволяет, например, придать видеофильму вид «старого кино», ускорить его воспроизведение и т. п.

Чтобы **добавить видеоэффект** из коллекции, нужно перетянуть его из Области содержимого на нужный видеофрагмент (или изображение) на Панели раскадровки или Шкале времени. После этого видеоэффект можно редактировать.

Если к одному видеофрагменту добавить сразу несколько видеоэффектов, то происходит наложение эффектов.

» Если к видеофрагменту дважды добавить эффект ускорение в два раза, видео будет воспроизводиться в четыре раза быстрее исходного.

Добавление переходов между видеофрагментами придает видеоклип более профессиональный и эстетичный вид.

Чтобы **добавить переход**, нужно выбрать команду Просмотр видео-переходов на Панели задач, перетянуть нужный переход из Области содержимого на Панель раскадровки или Шкалу времени между видеофрагментами либо статическими изображениями.

Продолжительность перехода определяется временем одновременного воспроизведения двух видеофрагментов.

Для изменения продолжительности переходов нужно на Шкале времени перетянуть один из видеофрагментов, увеличивая или уменьшая площадь перекрытия (переход) с другим видеофрагментом (рис. 13.1). Даже если переход между видеофрагментами не был добавлен специально, автоматически устанавливается Плавный переход.

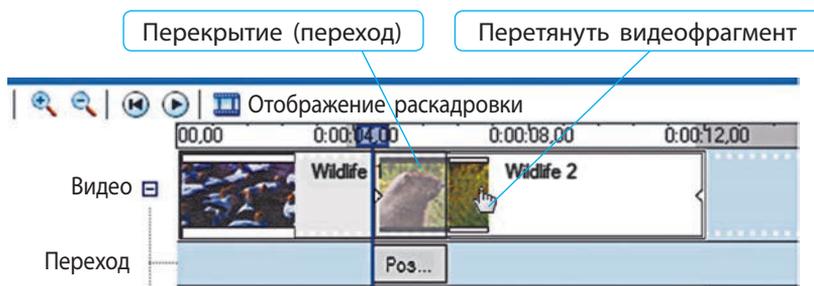


Рис 13.1

Настройка временных параметров аудио- и видеоряда

Видеоклип обычно содержит титры, изображения, аудио- и видеофрагменты, видеоэффекты и т. п. Настройка временных параметров клипа состоит в том, чтобы установить время воспроизведения этих составляющих и синхронизации аудио и видео.

Продолжительность воспроизведения фрагментов, переходов или видеоэффектов на Шкале времени настраивается перемещением соответствующих маркеров монтажа (рис. 13.2).

При этом для графических изображений, переходов, видеоэффектов и текстовых надписей изменяется время их воспроизведения, а для видео- и аудиофрагментов выполняется обрезка (или восстановление).

Для удаления из видеоклипа ненужного объекта достаточно его выделить и нажать клавишу Delete.



Рис. 13.2

Сохранение проекта видеоклипа

Если за определенное время видеоклип создать не удастся, то незавершенный проект нужно сохранить.

При **сохранении проекта** создается файл с расширением .mswmm, который содержит информацию о структуре проекта, но не содержит самих медиафайлов. Поэтому при работе с проектом на другом компьютере возникнут проблемы из-за отсутствия необходимых файлов. Чтобы этого избежать, необходимо сохранить проект и все нужные файлы в отдельной папке и уже из нее импортировать их в Windows Movie Maker. Для работы с проектом на другом компьютере достаточно будет иметь данную папку.

Готовый видеоклип необходимо сохранить. Для этого можно воспользоваться командой Сохранение на Панели задач на компьютере.

Сервисы размещения аудио- и видеофайлов в Интернете

Чтобы просмотреть созданный видеоклип в Интернете, его можно сохранить на «облачном» диске (Yandex-диск, Google-диск и т. п.) и по почте или через социальные сети переслать ссылку на видеоклип.

Другой вариант распространения видео — это размещение на специализированных интернет-сервисах или в социальных сетях.

» Одним из наиболее популярных сейчас интернет-сервисов для просмотра и распространения видео является YouTube. На нем пользователи могут поделиться друг с другом видеоклипами, комментировать и оценивать их. Существуют также другие подобные сервисы: Video Direct, Vimeo и т. п.

Чтобы **разместить видеоклип в Интернете**, нужно:

- 1) войти в аккаунт YouTube (Google);
- 2) щелкнуть кнопку Добавить видео;
- 3) задать тип доступа;
- 4) выбрать видео для загрузки;
- 5) ввести информацию о видео;
- 6) нажать Опубликовать.



После размещения видеоклипа в Интернете пользователь получит электронное письмо со ссылкой на фильм.

Вопросы для самопроверки



1. Каково назначение видеоэффектов; переходов?
2. Как добавить переходы; видеоэффекты?
3. Как обрезать аудио- или видеофрагмент?
4. Как сократить время воспроизведения изображения?
5. Что содержит файл проекта Windows Movie Maker?
6. Как опубликовать видеоклип в Интернете?

Упражнение 13



- Настроить временные параметры, добавить переходы между видеофрагментами.
- 1) Запустите Windows Movie Maker и откройте проект с названием Упражнение 12, созданный на прошлом уроке. Просмотрите видеоряд фильма в области просмотра.
 - 2) Добавьте переходы между фрагментами и видеоэффекты.
 - 3) Переместите аудиофрагмент звукового сопровождения так, чтобы его воспроизведение совпадало с началом фильма. Сохраните проект с именем Упражнение 13 в своей папке.

Компьютерное тестирование



Выполните тестовое задание 13 с автоматической проверкой на сайте interactive.ranok.com.ua.

Практическая работа 6



Создание видеоклипа. Добавление видеоэффектов, настройка временных параметров аудио- и видеоряда

Задание: создать видеоклип со звуковым сопровождением и видеоэффектами на основе ранее подготовленного проекта.

Оборудование: компьютер с установленным браузером, видеоредактором, подключенный к сети Интернет; наушники; подготовленный ранее проект Упражнение 13.

Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

- ▶ 1. Запустите видеоредактор. Откройте проект Упражнение 13.
- ▶ 2. Просмотрите видеофрагменты, в каждом из них захватите по одному кадру, характеризующему содержание фрагментов.
- ▶ 3. Перетяните захваченные кадры на Шкалу времени или Панель раскадрировки, размещая их перед нужными видеофрагментами.
- ▶ 4. Установите переходы между всеми объектами видеоряда.
- ▶ 5. Добавьте видеоэффекты к некоторым объектам видеоряда.
- ▶ 6. На одном из захваченных кадров добавьте текст с кратким описанием соответствующего видеофрагмента.
- ▶ 7. Добавьте анимированное название в начале клипа.
- ▶ 8. Добавьте титры, в которых укажите автора и дату создания видеоклипа.
- ▶ 9. Отключите звучание дорожки Звук.
- ▶ 10. Синхронизируйте время воспроизведения аудио- и видеоряда. Для этого обрежьте звуковой ряд или измените время отображения видеоряда.
- ▶ 11. Просмотрите видеоклип. Проверьте, чтобы он не заканчивался на «полукадре» или «полуслове», при необходимости отредактируйте.
- ▶ 12. Сохраните проект с именем Практическая работа 6 в своей папке. Завершите работу за компьютером.

Сделайте вывод: как создать видеоклип, добавить видеоэффекты, настроить временные параметры аудио- и видеоряда, синхронизировать аудио- и видеоряд клипа.



Практическая работа 7

Размещение аудио- и видеоматериалов в Интернете

Задание: разместить собственный видеоклип в сети Интернет и предоставить к нему доступ.

Оборудование: компьютер с подключенными наушниками, установленными браузером, видеоредактором, видеоконвертером, подключенный к сети Интернет; подготовленный проект Практическая работа 6.

Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

- ▶ 1. Запустите видеоредактор и откройте проект с названием Практическая работа 6.
- ▶ 2. Сохраните видеоклип с именем Практическая работа 7 в папке, указанной учителем. Закройте видеоредактор.
- ▶ 3. Запустите видеоконвертер по указанию учителя.
- ▶ 4. Сконвертируйте файл Практическая работа 7 в формат .mp4 и сохраните файл в своей папке. Закройте видеоконвертер.
- ▶ 5. Запустите браузер, откройте свою электронную почту.
- ▶ 6. Зайдите на свой «облачный» диск. Сохраните на нем созданный видеоклип в формате .mp4.
- ▶ 7. Откройте доступ к видеоклипу всем, кто имеет на него ссылку.
- ▶ 8. Пришлите электронное письмо со ссылкой на видеоклип по адресу, указанному учителем.
- ▶ 9. Зарегистрируйтесь на сервисе YouTube (Google) и (или) войдите в свой аккаунт.
- ▶ 10. Опубликуйте видеоклип на YouTube для общего доступа.
- ▶ 11. Отшлите электронное письмо со ссылкой на видеоклип в YouTube по адресу, указанному учителем.
- ▶ 12. Завершите работу за компьютером.

Сделайте вывод: как разместить аудио- и видеоматериалы в Интернете; какой из способов размещения видеоклипа в Интернете для вас более приемлем и почему.

РАЗДЕЛ 5

ТЕХНОЛОГИИ ОБРАБОТКИ ЧИСЛОВЫХ ДАННЫХ В СРЕДЕ ТЕКСТОВОГО ПРОЦЕССОРА



§ 14. Типы ссылок на ячейки в табличном процессоре Excel

Практическая работа 8. Решение вычислительных задач

§ 15. Логические функции

§ 16. Математические и статистические функции

§ 17. Параметры страницы. Печать таблицы

§ 18. Сортировка данных. Условное форматирование

Практическая работа 9. Использование математических, логических и статистических функций в табличном процессоре. Условное форматирование

§ 19. Расширенный фильтр. Промежуточные итоги

§ 20. Диаграммы и графики

Практическая работа 10. Сортировка данных в таблицах.

Автофильтры и расширенные фильтры

ПОВТОРЯЕМ



Вы уже знакомы с табличным процессором — прикладной программой для обработки данных, представленных в виде электронных таблиц.

Документом Excel является *рабочая книга*, которая состоит из *листов*. Лист содержит электронную таблицу, состоящую из *столбцов* и *строк*, на пересечении которых образуются ячейки. Каждая ячейка имеет свой *адрес*, который состоит из заголовка столбца и номера строки.

Ячейки могут содержать данные разных типов: текстовые, числовые и т. д., а также формулы, предназначенные для выполнения вычислений над данными. Любая *формула* начинается знаком «=» (равно) и может содержать конкретные значения, адреса ячеек, встроенные функции, знаки арифметических и логических операций.

1. Для чего предназначен табличный процессор?
2. Из чего состоит адрес ячейки электронной таблицы?
3. Что представляет собой документ Excel?
4. Что могут содержать ячейки электронной таблицы?
5. Для чего предназначены формулы?
6. Каким знаком начинается запись формулы в табличном процессоре Excel?



В этом разделе вы продолжите знакомиться с функциями для обработки данных в электронных таблицах, рассмотрите способы обработки и сортировки данных.

35	52	70	87	105	122	140	157	175	8	3	6	9	5*	0.0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645														
732	750	767	785	802	819														

§ 14. Типы ссылок на ячейки в табличном процессоре Excel

§ 14. Типы ссылок на ячейки в табличном процессоре Excel

При решении прикладных задач часто возникает необходимость в обработке данных, содержащихся в нескольких ячейках. Например, чтобы найти наибольшее или наименьшее число и т. п., нужно обработать множество чисел, находящихся в определенном диапазоне ячеек.

Диапазоны ячеек



Ссылками называют адреса ячеек и диапазонов ячеек, используемые в формулах.

Диапазон ячеек — это совокупность ячеек, которые можно обрабатывать как единое целое.

Различают *связный* и *несвязный* диапазоны.

Связным является диапазон, который можно выделить путем перемещения указателя при нажатой левой кнопке мыши.

Для **выделения прямоугольного диапазона** нужно протянуть указатель мыши между угловыми диагонально противоположными ячейками диапазона или щелкнуть одну угловую ячейку диапазона и при нажатой клавише Shift щелкнуть диагонально противоположную ячейку.

Для **выделения диапазона столбцов или строк** нужно протянуть указатель мыши по заголовкам столбцов или номерам строк.

Отдельная ячейка также является связным диапазоном и выделяется щелчком левой кнопки мыши.

Адрес *связного диапазона* задается соответственно адресами двух ячеек (верхней слева и нижней справа); номерами двух строк (заголовками двух столбцов), разделенных двоеточием; адресом одной ячейки.

- » A1:C5 — прямоугольный диапазон из 15 ячеек (рис. 14.1, а);
- 2:4 — диапазон из всех ячеек строк 2, 3 и 4;
- A:B — диапазон из всех ячеек столбцов А и В;
- B7 — диапазон из одной ячейки (рис. 14.1, б).

Несвязный диапазон состоит из нескольких связных диапазонов. Адрес *несвязного диапазона* задается адресами связных диапазонов, разделенными точкой с запятой.

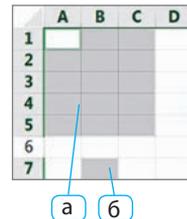


Рис. 14.1



» A1:C5;B7 — несвязный диапазон из 16 ячеек (рис 14.1, а, б)

Для **выделения несвязного диапазона** нужно при нажатой клавише Ctrl последовательно выделить несколько связанных диапазонов.

Для **отмены всех выделений** нужно щелкнуть любую ячейку электронной таблицы.

! Связный диапазон можно копировать и перемещать, что не всегда возможно для несвязных диапазонов.

Типы ссылок

Ячейки, на которые в формулах есть ссылки, называют **адресными**. Различают *относительные, абсолютные, смешанные* ссылки.

Адрес ячейки состоит из заголовка столбца, который обычно обозначается буквами, и номера строки, являющегося числом, например A1. Ссылку, содержащую обычный адрес ячейки, называют **относительной**.

При копировании формул происходит модификация относительных ссылок: номера строк (заголовки столбцов) в адресе ячейки изменяются на разность номеров строк (столбцов) начальной и конечной ячеек с формулой.

» Если формулу =A4+B5 скопировать в ячейку, расположенную на два столбца правее и три строки выше от заданной, то в новой формуле в адресах ячеек все заголовки столбцов увеличатся на 2, а номера строк уменьшатся на 3, и мы получим формулу =C1+D2 (рис. 14.2).

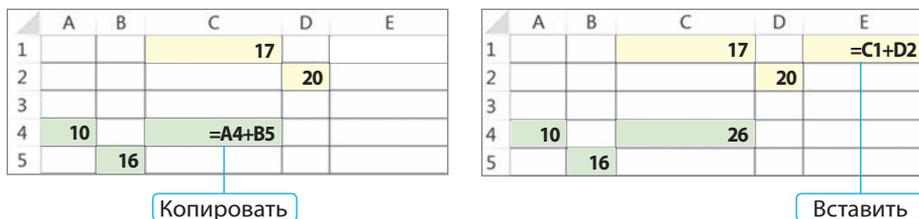


Рис. 14.2

! Если формула содержит адрес ячейки, в которой она же и записана, то возникает ошибка «циклическая ссылка».

Для **копирования формулы в смежные ячейки** можно выделить ячейку с формулой и перетянуть маркер заполнения (рис. 14.3). Полученные таким образом формулы будут модифицированы.



Рис. 14.3

35	52	70	87	105	122	140	157	175	192	210	227	244	262	279	297	314	332	349	367	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	733	750	767	785	802	819	837	854	872	889	906	924	941	958	976	993	1011	1028	1045	1063	1080	1097	1115	1132	1149	1167	1184	1202	1219	1236	1253	1271	1288	1305	1323	1340	1357	1375	1392	1409	1427	1444	1461	1478
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

§ 14. Типы ссылок на ячейки в табличном процессоре Excel

Если формула содержит ссылку на ячейку, адрес которой при копировании не должен изменяться, то используют абсолютную ссылку — вводят знак «\$» перед заголовком столбца и номером строки.

➤ При копировании в любую ячейку формула `=A$1` не изменится и всегда будет содержать ссылку на ячейку с адресом A1 (рис. 14.4).

	A	B	C	D
1	7	<code>=A\$1</code>		
2	8			
3	9		<code>=A\$1</code>	
4				

Копировать Вставить

Рис. 14.4

Если в ссылке на ячейку знак «\$» содержится только перед заголовком столбца или номером строки, то такую ссылку называют **смешанной**.

При копировании формул со смешанными ссылками в адресах ячеек модифицируются только заголовки столбцов или номера строк, перед которыми отсутствует знак «\$».

⚠ Если при редактировании формулы установить курсор на ссылке и нажать клавишу F4, то тип ссылки будет изменяться автоматически.

При копировании данных из адресных ячеек в другие ячейки ссылки на них в формулах не модифицируются.

При перемещении данных из адресных ячеек в другие ячейки все ссылки на них модифицируются в зависимости от их нового местоположения с сохранением типов ссылок.

➤ При перемещении данных из адресной ячейки A1 в ячейку C3 все ссылки A1; \$A\$1; A\$1; \$A1 соответственно модифицируются так: C3; \$C\$3; C\$3; \$C3 (рис. 14.5).

Модифицируются также все ссылки при **перемещении адресной ячейки** в результате вставки либо удаления столбцов, строк, ячеек или диапазонов ячеек, расположенных *выше* или *левее* от адресной.

Удаление адресных ячеек приводит к ошибкам в формулах, содержащих ссылки на удаленные ячейки (рис. 14.6).

	A	B	C
1			
2		<code>=A1</code>	
3			

Вырезать

	A	B	C
1			
2			<code>=C3</code>
3			

Вставить

Рис. 14.5

	A	B
1		
2		<code>=A1</code>

Вырезать
Копировать
Вставить...
Удалить...

	A	B	C
1			
2			#REF!

Ошибка — неправильная ссылка на ячейку

Рис. 14.6

18'	24	
924	941	958
1097	1115	1132
1271	1288	1305
1461	1478	1495

79

Вопросы для самопроверки



1. Как выделить прямоугольный диапазон ячеек?
2. Сколько ячеек в диапазоне A2:B3?
3. Как выделить диапазон, состоящий из столбцов или строк?
4. Что такое относительная ссылка?
5. Как записываются абсолютные ссылки; смешанные ссылки?

Упражнение 14



- Рассчитать массу продуктов для приготовления борща в школьной столовой как произведение нормы их закладки на количество порций.

- 1) Запустите программу Excel. Заполните таблицу по образцу.

	A	B	C	D	E	F	G
1		По норме, г	85	172	155	57	
2	Классы	Порций	Мясо	Картофель	Капуста	Свекла	
3	8	117					
4	9	109					
5	10	107					
6	11	121					
7	всего						

- 2) Введите в ячейку C3 формулу для расчета массы мяса, нужного для приготовления борща для 8-х классов. Установите числовой формат данных с одним десятичным знаком после запятой.
- 3) Введите в ячейки D3, E3, F3 формулы для расчета массы овощей для 8-х классов. Используя маркер заполнения, скопируйте соответствующие формулы в диапазоны C4:C6, D4:D6, E4:E6, F4:F6.
- 4) В ячейку B7 введите формулу для расчета общего количества порций борща для всех классов.
- 5) Используя маркер заполнения, скопируйте формулу из ячейки B7 в ячейки диапазона C7:F7. В ячейку G2 введите текст «Масса продуктов, кг», в ячейку G3 — формулу для расчета общей массы всех продуктов для 8-х классов.
- 6) В ячейках G4:G6 рассчитайте общую массу продуктов для остальных классов, в ячейке G7 — для всех классов.
- 7) Сохраните файл с именем Упражнение 14 в своей папке. Завершите работу за компьютером.

Компьютерное тестирование



Выполните тестовое задание 14 с автоматической проверкой на сайте interactive.ranok.com.ua.



Практическая работа 8

Решение вычислительных задач

Задание: рассчитать зарплату сотрудников за месяц, используя соответствующие формулы.

Оборудование: компьютер с табличным процессором Excel, подключенный к сети Интернет.

Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

- ▶ 1. Запустите программу Excel. Создайте таблицу расчета зарплат по образцу (см. рисунок).

	A	B	C	D	E
1		Рабочих дней в месяце			Курс €
2		22			
3	Фамилия	Отработано дней	Оклад, грн	Зарплата, грн	Зарплата, евро
4	Петренко	20	5 900,00 ₴		
5	Сидоренко	13	4 880,00 ₴		
6	Иванов	12	6 175,00 ₴		
7	Петрова	17	6 266,00 ₴		
8					

- ▶ 2. Найдите в Интернете курсы валют, введите данные в ячейку E2.
- ▶ 3. Установите в нужных ячейках соответствующие денежные форматы с указанием денежной единицы (грн ₴, евро €).
- ▶ 4. В ячейке D4 задайте формулу для расчета зарплат Петренко в гривнях. *Примечание.* Оклад — это зарплата сотрудника за все отработанные рабочие дни текущего месяца, указанные в ячейке B2. Петренко зарабатывает в день $5900 / 22 \approx 268,18$ (грн). Поскольку Петренко работал 20 дней, то его зарплата за этот месяц: $268,18 \cdot 20 \approx 5363,64$ (грн).
- ▶ 5. Скопируйте формулу для расчета зарплат в гривнях для остальных сотрудников в соответствующие ячейки.
- ▶ 6. Введите в ячейку E4 формулу для расчета зарплат Петренко в евро.
- ▶ 7. Скопируйте полученную формулу в соответствующие ячейки для расчета зарплат остальных сотрудников в евро.
- ▶ 8. Введите в ячейку C8 формулу для расчета среднего оклада всех сотрудников.
- ▶ 9. В ячейках D8, E8 рассчитайте сумму зарплат всех сотрудников в гривнях и евро.

- ▶ **10.** Сохраните файл с именем Практическая работа 8 в своей папке. Завершите работу за компьютером.

Сделайте вывод: как записываются формулы, от чего зависит правильность вычислений по скопированным формулам; в каких случаях ссылки в формулах должны быть абсолютными.

§ 15. Логические функции

Классификация функций

В формулах Excel для обработки данных можно использовать различные встроенные функции, каждая из которых имеет уникальное имя. По категориям функции распределяют на *математические, статистические, логические, финансовые* и т. п.



Значения, которые обрабатывает функция, называют ее **аргументами**.

Аргументами функции могут быть другие функции, выражения, ссылки и т. п. Аргументы функции записывают в скобках после имени (знака) функции (рис. 15.1).

Функция в Excel характеризуется не только именем, но и *количеством аргументов, типом аргументов, результатом*.

Рассмотрим известную вам с 7 класса функцию для вычисления наибольшего значения (рис. 15.1):

- имя — МАКС(MAX);
- количество аргументов — не менее одного;
- тип аргумента — числовой;
- результат — наибольшее значение из списка аргументов.

		A	
1	= МАКС(B1:C2)	4	1 2
2			3 4

Имя (подчеркнуто в формуле)
Аргументы (B1:C2)
Результат (4)

Рис. 15.1

Если функция имеет несколько аргументов, их разделяют точкой с запятой.

Функции в формулы вставляют с помощью инструментов вкладки Формулы. Их также удобно вставлять с помощью окна Мастер функций, которое открывается при нажатии кнопки fx, расположенной в строке формул.

Нужную функцию можно найти по описанию действия, категории или имени (рис. 15.2).

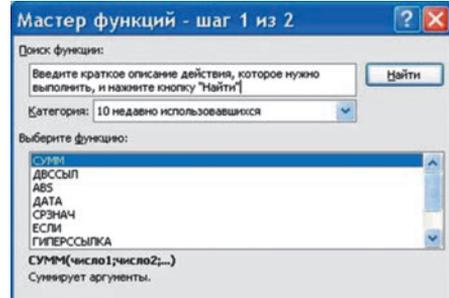


Рис. 15.2

Логические функции

В 7 классе вы уже знакомились с высказываниями, которые могут быть истинными или ложными.



Логическое выражение — это выражение, принимающее одно из двух значений: ИСТИНА (True) или ЛОЖЬ (False).

Рассмотрим **простые логические выражения**, состоящие из двух математических выражений, связанных операцией отношения:

- больше >
- меньше <
- равно =
- не меньше (больше или равно) >=
- не больше (меньше или равно) <=
- не равно <>



$2 > 1$ — истина, $2 < 1$ — ложь.



Логическая функция — это функция, один или несколько аргументов которой являются логическими выражениями или же сама функция принимает значение ИСТИНА или ЛОЖЬ.

Рассмотрим одну из основных логических функций ЕСЛИ(IF).

ЕСЛИ(Условие; Выражение1; Выражение2)

Условие — это логическое выражение.

Выражение1 — становится результатом функции, когда условие истинно или его значение не равно нулю.

Выражение2 — становится результатом функции, когда условие ложно или его значение равно нулю.



В электронной ведомости учебных достижений учащихся около оценки нужно написать слово «хорошо», если учащийся имеет оценку выше 6 баллов, и слово «удовлетворительно», если учащийся имеет другую оценку.

Фрагмент алгоритма решения этой задачи представлен в виде блок-схемы (рис. 15.3).

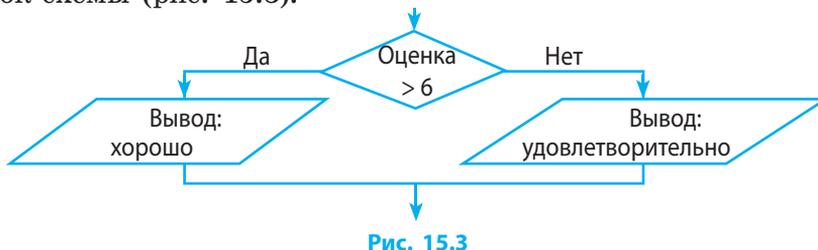


Рис. 15.3

В программе Excel этот алгоритм реализуется с помощью функции ЕСЛИ(IF): ЕСЛИ(B1>6; "хорошо"; "удовлетворительно") (рис. 15.4).

	A	B	C
1	Петренко	5	=ЕСЛИ(B1>6;"хорошо";"удовлетворительно")
2	Сидоренко	8	хорошо
3	Петрова	12	хорошо
4	Сидорова	4	удовлетворительно

Рис. 15.4

В функциях Excel на месте логического выражения может быть имя или адрес ячейки, число или арифметическое выражение. В этом случае нулевое значение аргумента трактуется как ЛОЖЬ, а ненулевое — как ИСТИНА.

- A1 трактуется как ИСТИНА или ЛОЖЬ в зависимости от значения ячейки A1; $2 + 2$ трактуется как ИСТИНА (поскольку $2 + 2 \neq 0$); $2 - 2$ трактуется как ЛОЖЬ (поскольку $2 - 2 = 0$).

Иногда для решения задачи нужно комбинировать несколько условий.



Составное логическое выражение — выражение, в котором несколько простых логических выражений связаны логическими операциями.

Основными логическими операциями являются НЕ (отрицание), И (конъюнкция или логическое умножение), ИЛИ (дизъюнкция или логическое сложение). В Excel они реализуются с помощью функций НЕ(NOT), И(AND), ИЛИ(OR).

- В ведомости учебных достижений (см. пример на с. 83) около оценки нужно написать слово «удовлетворительно» для учащихся, которые имеют оценки выше 6 баллов, но ниже 10, для остальных учащихся — ничего не писать.

Воспользуемся логическими функциями ЕСЛИ(IF) и И(AND).

В данном случае получим: =ЕСЛИ(И(В1>6;В1<10); "удовлетворительно"; ""). Последний аргумент функции ЕСЛИ(IF) — это кавычки без символов между ними, то есть пустая строка.

Иногда целесообразно использовать вложенные логические функции IF(ЕСЛИ).

➤ Пусть нужно оценить учебные достижения учащихся по трем условным уровням: «начальный» (1–4 балла); «средний» (5–8 баллов); «высокий» (9–12 баллов).

В Excel эту задачу можно решить следующим образом (рис. 15.5):

=ЕСЛИ(В1<5;"начальный"; ЕСЛИ(В1<9;"средний";"высокий"))

=IF(В1<5;"начальный"; IF(В1<9;"средний";"высокий"))

В Excel допускается вложение нескольких функций ЕСЛИ(IF).

	A	B	C
1	Петренко	5	=ЕСЛИ(В1<5;"начальный";ЕСЛИ(В1<9;"средний";"высокий"))
2	Сидоренко	8	средний
3	Петрова	12	высокий
4	Сидорова	4	начальный

Рис. 15.5

Рассмотрим еще несколько логических функций:

Имя (рус./англ.)	Описание	Количество аргументов
ЕСЛИОШИБКА (IFERROR)	Если первый аргумент не является ошибкой, то возвращает его значение, в противном случае возвращает значение второго аргумента	Два
ЛОЖЬ(FALSE)	Возвращает значение «ложь»	Без аргументов
ИСТИНА(TRUE)	Возвращает значение «истина»	Без аргументов

Вопросы для самопроверки



1. Что такое функция в Excel?
2. Что такое аргумент функции?
3. Назовите несколько категорий функций в Excel.
4. Существуют ли в Excel функции без аргументов?
5. Как обозначаются операции отношений?
6. Что является первым аргументом логической функции ЕСЛИ(IF)?

Упражнение 15



- ▶ Создать в Excel таблицу учебных достижений учащихся по информатике по приведенному описанию.
- 1) Запустите программу Excel. Создайте и заполните таблицу учебных достижений учащихся вашей группы (класса) по информатике: введите в столбец А фамилии, в столбец В — оценки за семестр.
 - 2) Введите формулу с использованием функции ЕСЛИ(IF) для вывода в столбце С таких уровней учебных достижений: «начальный» (оценка менее 5 баллов) и «высокий» (для остальных оценок) по аналогии с примером на рис. 15.4.
 - 3) Введите формулу для вывода в столбце D таких уровней учебных достижений: «Юниор» (1–4 балла), «Любитель» (5–8 баллов), «Эксперт» (9–12 баллов).
 - 4) Сохраните файл с именем Упражнение 15 в своей папке. Завершите работу за компьютером.

Компьютерное тестирование



Выполните тестовое задание 15 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 16. Математические и статистические функции

Математические функции используют для различных вычислений. С полным перечнем математических функций можно ознакомиться на вкладке Функции в группе Библиотека функций.

Математические функции



Математические функции в Excel — это функции, предназначенные для выполнения вычислений над числами и/или данными ячейек: округление, преобразование чисел и т. п.

Рассмотрим некоторые математические функции:

Имя (рус./англ.)	Описание действия	Количество аргументов
ОКРУГЛ(ROUND)	Округление до указанного количества цифр после запятой	Два
ОСТАТ(MOD)	Остаток от деления	Два
ПРОИЗВЕД(PRODUCT)	Произведение	Не менее одного
СУММ(SUM)	Сумма	Не менее одного
СУММЕСЛИ(SUMIF)	Сумма при условии, если...	Три

➤ Если требуется найти произведение числовых данных в диапазоне ячеек A1:B2 и округлить его до десятых, то формула будет иметь вид: =ОКРУГЛ(ПРОИЗВЕД(A1:B2);1) или =ROUND(PRODUCT(A1:B2);1)

Статистические функции



Статистические функции в Excel — это функции, предназначенные для анализа данных, содержащихся в определенном диапазоне ячеек.

С помощью статистических функций можно найти наибольшее и наименьшее значения, рассчитать среднее значение заданного множества чисел и т. д.

➤ Определить, сколько учащихся учится на 12 баллов, можно с помощью статистической функции СЧЁТЕСЛИ(COUNTIF). Первым аргументом является диапазон ячеек с оценками, а вторым аргументом — критерий выбора:

=СЧЁТЕСЛИ(B3:B4;"=12") или =COUNTIF(B3:B4;"=12")

Знак равенства и кавычки в записи критерия выбора можно опустить: =СЧЁТЕСЛИ(B3:B4;12) или =COUNTIF(B3:B4;12)

Критерием выбора, кроме числа, может быть дата, символ, слово и т. п. По критерию строится условие: содержит ли текущая ячейка указанные данные. Например, можно подсчитать количество уроков, пропущенных учащимися (см. рисунок).

	A	B	C	D	E	F	G	H
1		Дата						
2	Фамилия	2 сен	9 сен	16 сен	23 сен	30 сен	7 окт	Количество пропусков
3	Петренко	6	7	н	8	6	н	=СЧЁТЕСЛИ(B3:G3;"н")
4	Иванова	9	н	н	9			СЧЁТЕСЛИ(диапазон; критерий)

Критерий выбора

Рассмотрим некоторые статистические функции:

Имя (рус./англ.)	Описание	Количество аргументов
МИН(MIN)	Минимальное значение	Не менее одного
МАКС(MAX)	Максимальное значение	Не менее одного
СРЗНАЧ(AVERAGE)	Среднее арифметическое значение	Не менее одного
СЧЁТ(COUNT)	Количество ячеек с числовыми данными	Не менее одного
СЧЁТЕСЛИ (COUNTIF)	Количество ячеек, данные которых удовлетворяют определенному условию	Не менее одного
СЧЁТЕСЛИМН (COUNTIFS)	Количество ячеек, данные которых удовлетворяют нескольким условиям	Не менее одного
СЧЁТЗ(COUNTA)	Количество непустых ячеек	Не менее одного

Вопросы для самопроверки



1. Для чего используют математические функции?
2. Приведите примеры математических функций в Excel.
3. Для чего предназначены статистические функции?
4. Приведите примеры статистических функций в Excel.
5. Приведите пример применения одной из статистических функций.
6. Приведите пример статистической функции, аргументом которой является логическое выражение (условие).

Упражнение 16



- Проанализировать успеваемость учащихся группы (класса) по информатике согласно описанию.
- 1) Запустите программу Excel. Создайте таблицу со столбцами, имеющими следующие заголовки: А — «Фамилия», Е — «Итоговая», В, С, D — текущие оценки (без заголовков). Введите в таблицу фамилии учащихся и текущие оценки.
 - 2) В соответствующих ячейках столбца Е введите формулы для вычисления итоговых оценок учащихся как среднее арифметическое текущих оценок, округленное до целого значения.
 - 3) В произвольной ячейке под таблицей с помощью функции СЧЁТЕСЛИМН(COUNTIFS) рассчитайте, сколько учащихся имеют итоговые оценки 5–8 баллов.

35	52	70	87	105	122	140	157	175	8	3	6	9	5*	0,0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698											
732	750	767	785	802	819	837	854	0,0872											

§ 17. Параметры страницы. Печать таблицы

- 4) В произвольной ячейке под таблицей рассчитайте общее количество текущих оценок всех учащихся.
- 5) Сохраните файл с именем Упражнение 16 в своей папке. Завершите работу за компьютером.



Компьютерное тестирование

Выполните тестовое задание 16 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 17. Параметры страницы. Печать таблицы

Перед печатью электронной таблицы на бумаге следует установить параметры страницы.

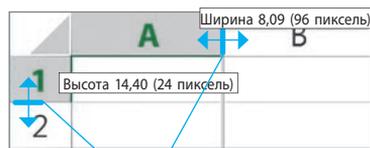
Установка параметров страницы

Параметры страницы в табличном процессоре Excel настраиваются так же, как и в текстовом процессоре Word.

Для установки параметров страницы следует в окне Параметры страницы на вкладке Разметка страницы выбрать нужный параметр (например, Поля) и указать одно из предложенных для него значений или ввести нужное значение с клавиатуры.

Установка размеров столбцов и строк таблицы

Чтобы изменить ширину столбца (высоту строки), достаточно перетянуть правую (нижнюю) границу между заголовками. При этом размеры соседних столбцов (строк) не изменяются, а в информационном окне в режиме просмотра Вид → Обычный выводится текущий размер: в пунктах и пикселях — для строк, в символах и пикселях — для столбцов (рис. 17.1).



Перетягивание границы

Рис. 17.1

18*	24	
924	941	958
1097	1115	1132
1271	1288	1305
1461	1478	1495

Для установки одинаковой ширины нескольких столбцов (рис. 17.2, а) или высоты нескольких строк (рис. 17.2, б) их следует выделить, а затем перетянуть одну из границ или выбрать в контекстном меню Ширина столбца (Высота строки) и установить нужное значение.

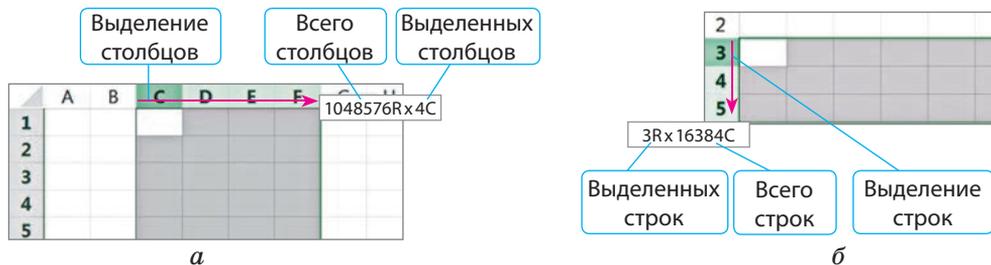


Рис. 17.2

Оптимизация таблицы

Если таблица не помещается на странице (рис. 17.3), можно ее оптимизировать: изменить ориентацию страницы, размер и ориентацию текста, размеры строк и столбцов и т. п. (рис. 17.4).

	A	B	C	D	E	F	G	H	I
1	Фамилия	Информатика	Физика	Химия	Алгебра	Геометрия	Изобразительное искусство	Астрономия	География
2	Петренко	12	10	8	9	11	9	10	10
3	Иванов	7	8	9	8	7	8	9	8
4	Петрова	5	6	5	7	6	5	6	6

Рис. 17.3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Фамилия	Информатика	Физика	Химия	Алгебра	Геометрия	Изобразительное	Астрономия	География					
2	Петренко	12	10	8	9	11	9	10	12					
3	Иванов	7	8	9	8	7	8	9	8					
4	Петрова	5	6	5	7	6	5	6	7					

Рис. 17.4

Для оптимизации таблицы (рис. 17.3) необходимо:

- 1) выделить нужные ячейки верхней строки;
- 2) на вкладке Главная найти группу Выравнивание и развернуть параметры;
- 3) установить в окне Формат ячеек выравнивание и ориентацию текста;
- 4) изменить высоту верхней строки и ширину столбцов.

35	52	70	87	105	122	140	157	175	8	3	6	9	5 ⁴	0,0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88 ⁴	3	6	9	6 ⁴	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87 ⁴	3	6	9	7 ⁴	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698											
732	750	767	785	802	819	837	854	0,0872											

§ 17. Параметры страницы. Печать таблицы

Объединение ячеек

Чтобы сделать, например, общий заголовок для нескольких столбцов таблицы, нужно несколько ячеек объединить в одну. Для этого ячейки следует выделить и выбрать на вкладке Главная в группе Выравнивание кнопку Объединить и поместить в центре . В ячейке останутся только данные ячейки, расположенной сверху слева.

Для отмены объединения ячеек нужно повторно щелкнуть ту же кнопку, при этом данные ячеек не восстанавливаются.

Установка границ ячеек

Таблица, которую мы видим на рабочем листе, разбита на ячейки условно. Если ее распечатать, то на бумаге отобразятся только данные в ячейках (без границ и заголовков столбцов и номеров строк).

Чтобы установить границы ячеек, необходимо выполнить следующие шаги:

- 1) выделить ячейку или диапазон ячеек;
- 2) на вкладке Главная выбрать группу Выравнивание;
- 3) в окне Формат ячеек на вкладке Границы выбрать тип линии, цвет, расположение границ по всей таблице (рис. 17.5, с. 92).

Печать таблицы

Печать электронной таблицы Excel происходит подобно печати текстового документа Word. При этом на бумагу выводятся только те листы книги Excel, в ячейках которых есть данные или имеются границы и/или заливка.

Во время подготовки документа к печати удобно работать в режиме Вид → Разметка страницы. При этом размеры строк и столбцов указываются в сантиметрах и отображается ориентировочный результат печати.

При подготовке электронной таблицы к печати нужно:

- 1) установить параметры страницы;
- 2) оптимизировать таблицу путем форматирования ее объектов;
- 3) объединить необходимые диапазоны ячеек;
- 4) установить внешние и/или внутренние границы ячеек в таблице.

К печати документа на бумаге можно перейти с помощью команды Файл → Печать или комбинации клавиш Ctrl + P. Перед печатью документ желательно предварительно просмотреть.

18 ⁴	24	
924	111	958
1097	1115	1132
1271	1288	1305
	1461	1478

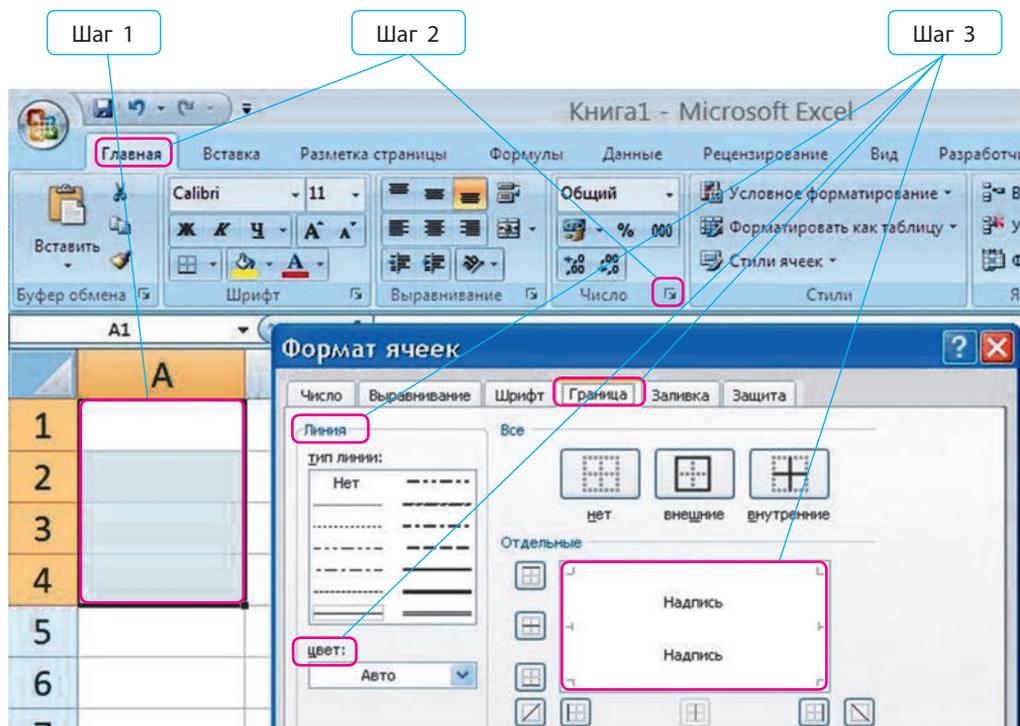


Рис. 17.5

Вопросы для самопроверки



1. Как изменить размер одного столбца (одной строки) таблицы?
2. Как установить одинаковый размер для нескольких столбцов (строк)?
3. Что предполагает оптимизация таблицы?
4. Как объединить несколько ячеек таблицы?
5. Как установить границы таблицы?
6. Как подготовить электронную таблицу к печати?

Упражнение 17



- ▶▶ Создать электронную таблицу по приведенному описанию.
- 1) Запустите программу Excel. Установите параметры страницы: поля — по 1 см, размер — А4 (21×29,7 см), ориентация —

35	52	70	87	105	122	140	157	175	192	210	227	244	262	279	297	314	332	349	366	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	732	750	767	785	802	819	837	854	872	889	906	924	941	958	975	992	1009	1026	1044	1061	1078	1095	1112	1129	1146	1163	1180	1197	1214	1231	1248	1265	1282	1299	1316	1333	1350	1367	1384	1401	1418	1435	1452	1469	1486	1503	1520	1537	1554	1571	1588	1605	1622	1639	1656	1673	1690	1707	1724	1741	1758	1775	1792	1809	1826	1843	1860	1877	1894	1911	1928	1945	1962	1979	1996	2013	2030	2047	2064	2081	2098	2115	2132	2149	2166	2183	2200	2217	2234	2251	2268	2285	2302	2319	2336	2353	2370	2387	2404	2421	2438	2455	2472	2489	2506	2523	2540	2557	2574	2591	2608	2625	2642	2659	2676	2693	2710	2727	2744	2761	2778	2795	2812	2829	2846	2863	2880	2897	2914	2931	2948	2965	2982	2999	3016	3033	3050	3067	3084	3101	3118	3135	3152	3169	3186	3203	3220	3237	3254	3271	3288	3305	3322	3339	3356	3373	3390	3407	3424	3441	3458	3475	3492	3509	3526	3543	3560	3577	3594	3611	3628	3645	3662	3679	3696	3713	3730	3747	3764	3781	3798	3815	3832	3849	3866	3883	3900	3917	3934	3951	3968	3985	4002	4019	4036	4053	4070	4087	4104	4121	4138	4155	4172	4189	4206	4223	4240	4257	4274	4291	4308	4325	4342	4359	4376	4393	4410	4427	4444	4461	4478	4495	4512	4529	4546	4563	4580	4597	4614	4631	4648	4665	4682	4699	4716	4733	4750	4767	4784	4801	4818	4835	4852	4869	4886	4903	4920	4937	4954	4971	4988	5005	5022	5039	5056	5073	5090	5107	5124	5141	5158	5175	5192	5209	5226	5243	5260	5277	5294	5311	5328	5345	5362	5379	5396	5413	5430	5447	5464	5481	5498	5515	5532	5549	5566	5583	5600	5617	5634	5651	5668	5685	5702	5719	5736	5753	5770	5787	5804	5821	5838	5855	5872	5889	5906	5923	5940	5957	5974	5991	6008	6025	6042	6059	6076	6093	6110	6127	6144	6161	6178	6195	6212	6229	6246	6263	6280	6297	6314	6331	6348	6365	6382	6399	6416	6433	6450	6467	6484	6501	6518	6535	6552	6569	6586	6603	6620	6637	6654	6671	6688	6705	6722	6739	6756	6773	6790	6807	6824	6841	6858	6875	6892	6909	6926	6943	6960	6977	6994	7011	7028	7045	7062	7079	7096	7113	7130	7147	7164	7181	7198	7215	7232	7249	7266	7283	7300	7317	7334	7351	7368	7385	7402	7419	7436	7453	7470	7487	7504	7521	7538	7555	7572	7589	7606	7623	7640	7657	7674	7691	7708	7725	7742	7759	7776	7793	7810	7827	7844	7861	7878	7895	7912	7929	7946	7963	7980	7997	8014	8031	8048	8065	8082	8099	8116	8133	8150	8167	8184	8201	8218	8235	8252	8269	8286	8303	8320	8337	8354	8371	8388	8405	8422	8439	8456	8473	8490	8507	8524	8541	8558	8575	8592	8609	8626	8643	8660	8677	8694	8711	8728	8745	8762	8779	8796	8813	8830	8847	8864	8881	8898	8915	8932	8949	8966	8983	9000	9017	9034	9051	9068	9085	9102	9119	9136	9153	9170	9187	9204	9221	9238	9255	9272	9289	9306	9323	9340	9357	9374	9391	9408	9425	9442	9459	9476	9493	9510	9527	9544	9561	9578	9595	9612	9629	9646	9663	9680	9697	9714	9731	9748	9765	9782	9799	9816	9833	9850	9867	9884	9901	9918	9935	9952	9969	9986	10003
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------

§ 17. Параметры страницы. Печать таблицы

книжная. Установите режим просмотра страницы: Вид → Разметка страницы.

- 2) Установите ширину столбцов: А — 0,5 см, В — 3,5 см. Остальным (кроме С:АС) задайте ширину 0,5 см.
- 3) Установите высоту первой и четвертой строк — по 1,5 см. Остальным строкам задайте высоту 0,5 см.
- 4) Объедините ячейки первой строки таблицы так, как показано на рисунке, и нужные ячейки в таблице.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1																													
2																													
3																													



- 5) Выделите диапазон ячеек A2:AC:C49. Установите внутренние границы таблицы минимальной толщины. Установите другие границы разной толщины так, как показано на рисунке.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1																													
2																													
3																													
4																													
5																													
6																													
7																													
8																													

- 6) Сохраните файл с именем Упражнение 17 в своей папке. Завершите работу за компьютером.

Компьютерное тестирование

Выполните тестовое задание 17 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 18. Упорядочение данных. Условное форматирование

Обычно данные электронной таблицы, полученные в результате расчетов или после ввода с клавиатуры, требуют упорядочения и систематизации.

Сортировка данных



Сортировка данных в Excel — это установка порядка расположения строк в соответствии с данными определенного столбца (столбцов).

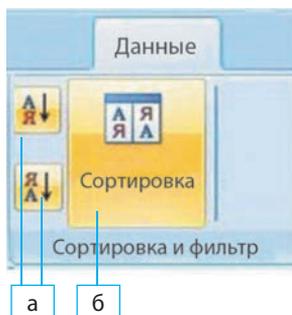


Рис. 18.1

Данные можно сортировать *по возрастанию* (в алфавитном порядке) или *убыванию* (в порядке, обратном алфавитному).

Чтобы выполнить сортировку данных по возрастанию или убыванию, нужно выделить связный диапазон ячеек с рассматриваемыми данными.

Для **простой сортировки данных по возрастанию или убыванию** нужно на вкладке **Данные** в группе **Сортировка и фильтр** щелкнуть кнопку соответствующего инструмента (рис. 18.1, *а*).

При этом строки выделенного диапазона переставятся в соответствии с данными первого столбца, пустые ячейки при этом расположатся внизу независимо от типа сортировки. Если выделить один столбец, то данные могут отсортироваться только в нем или в расширенном диапазоне по данным выделенного столбца. Какой именно диапазон сортировать при таком выделении, определяет пользователь.

Для **сложной сортировки данных** по нескольким условиям нужно на вкладке **Данные** в группе **Сортировка и фильтр** щелкнуть кнопку открытия диалогового окна **Сортировка** (рис. 18.1, *б*), в котором указать условия сортировки: по данным каких столбцов, по каким параметрам, в каком порядке и т. п.

При настройке сложной сортировки по данным нескольких столбцов данные верхней строки выделенного диапазона считаются заголовками и сортировке не подлежат.

35	52	70	87	105	122	140	157	175	8	3	6	9	54	0.0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88	3	6	9	64	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87	3	6	9	74	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	697	87	3	6	9	74	1219	1236	1253	1271	1288	1305
732	750	767	785	802	819	837												1461	1478

§ 18. Упорядочение данных. Условное форматирование

Порядок сортировки по нескольким столбцам задает пользователь в окне Сортировка. Вначале будут отсортированы данные по первому указанному столбцу. По каждому следующему столбцу будут сортироваться только те строки, в которых по предыдущему отсортированному столбцу данные совпадали (рис. 18.2).

A	B	C
1	Фамилия	Имя
2	Бабич	Виктор
3	Бойко	Тарас
4	Зленко	Тарас
5	Валеева	Ольга
6	Тойко	Тарас

A	B	C
1	Фамилия	Имя
2	Бабич	Виктор
3	Валеева	Ольга
4	Бойко	Тарас
5	Зленко	Тарас
6	Бойко	Тарас

A	B	C
1	Фамилия	Имя
2	Бабич	Виктор
3	Валеева	Ольга
4	Бойко	Тарас
5	Бойко	Тарас
6	Зленко	Тарас

A	B	C
1	Фамилия	Имя
2	Бабич	Виктор
3	Валеева	Ольга
4	Бойко	Тарас
5	Бойко	Тарас
6	Зленко	Тарас

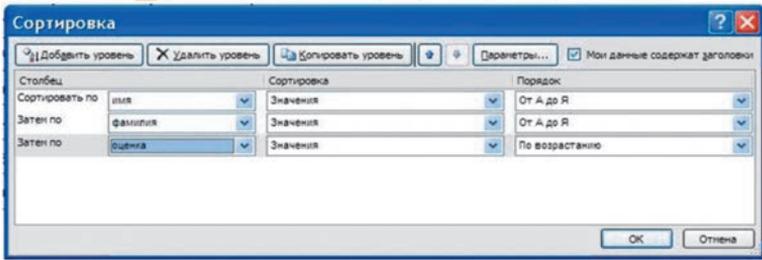


Рис. 18.2

Фильтрация данных



Фильтрация данных в Excel — это отображение в таблице только тех строк, данные которых соответствуют определенным условиям (условиям фильтрации).

При фильтрации строки таблицы, не соответствующие условиям фильтрации, будут скрыты и могут быть отображены при других условиях. Верхняя строка выделенного диапазона не фильтруется и всегда видима, поэтому в нее обычно вписывают заголовки столбцов.

Для включения режима фильтрации нужно активировать кнопку фильтра вкладки Данные в группе Сортировка и фильтр. В верхнем ряду связанного с активной ячейкой диапазона появятся значки развёртывания фильтра ▼ (рис. 18.3). Если связный диапазон ячеек предварительно выделить, то значки появятся во всех ячейках верхней строки выделенного диапазона (рис 18.4).

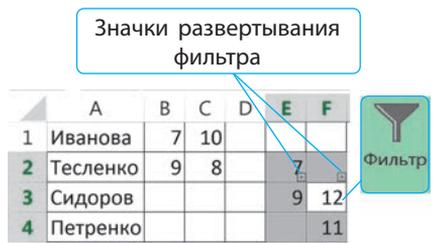


Рис. 18.3

18'	24	
924	958	1132
1097	1115	1305
1271	1288	1478

Обычно для фильтрации выделяют необходимые столбцы таблицы вместе с заголовками. Тогда при включении режима фильтрации значки разворачивания фильтра появятся во всех ячейках строки заголовков.

Если цвет шрифта или заливки отличается от черного и белого, можно установить Фильтр по цвету.

Для столбцов с числовыми данными доступным будет пункт Числовые фильтры, для столбцов с текстовыми данными — пункт Текстовые фильтры (рис. 18.4).

Настройки условия фильтрации осуществляются в окне Пользовательский автофильтр, которое открывается командой Текстовые фильтры (Числовые фильтры) → Настраиваемый фильтр.

Исходная таблица

Отображение строки с оценками Иванова

Начинаются на букву «И» и заканчиваются на «ов»

Рис. 18.4

После установки в столбце нового фильтра «старое» правило фильтрации аннулируется.

Фильтры можно применять к нескольким столбцам одновременно. При этом каждый фильтр применяется к строкам, которые остались видимыми после предыдущей фильтрации.

Условное форматирование

Чтобы ячейки с определенными данными выделить среди других ячеек, применяют условное форматирование.



Условное форматирование ячеек — автоматическое изменение формата ячейки на заданный, если данные в ней соответствуют определенному условию.

35	52	70	87	105	122	140	157	175	192	210	227	244	262	279	297	314	332	349	366	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	681	698	716	734	752	769	787	805	823	841	859	877	895	913	931	949	967	985	1003	1021	1039	1057	1075	1093	1111	1129	1147	1165	1183	1201	1219	1237	1255	1273	1291	1309	1327	1345	1363	1381	1399	1417	1435	1453	1471	1489	1507	1525	1543	1561	1579	1597	1615	1633	1651	1669	1687	1705	1723	1741	1759	1777	1795	1813	1831	1849	1867	1885	1903	1921	1939	1957	1975	1993	2011	2029	2047	2065	2083	2101	2119	2137	2155	2173	2191	2209	2227	2245	2263	2281	2299	2317	2335	2353	2371	2389	2407	2425	2443	2461	2479	2497	2515	2533	2551	2569	2587	2605	2623	2641	2659	2677	2695	2713	2731	2749	2767	2785	2803	2821	2839	2857	2875	2893	2911	2929	2947	2965	2983	3001	3019	3037	3055	3073	3091	3109	3127	3145	3163	3181	3199	3217	3235	3253	3271	3289	3307	3325	3343	3361	3379	3397	3415	3433	3451	3469	3487	3505	3523	3541	3559	3577	3595	3613	3631	3649	3667	3685	3703	3721	3739	3757	3775	3793	3811	3829	3847	3865	3883	3901	3919	3937	3955	3973	3991	4009	4027	4045	4063	4081	4099	4117	4135	4153	4171	4189	4207	4225	4243	4261	4279	4297	4315	4333	4351	4369	4387	4405	4423	4441	4459	4477	4495	4513	4531	4549	4567	4585	4603	4621	4639	4657	4675	4693	4711	4729	4747	4765	4783	4801	4819	4837	4855	4873	4891	4909	4927	4945	4963	4981	4999	5017	5035	5053	5071	5089	5107	5125	5143	5161	5179	5197	5215	5233	5251	5269	5287	5305	5323	5341	5359	5377	5395	5413	5431	5449	5467	5485	5503	5521	5539	5557	5575	5593	5611	5629	5647	5665	5683	5701	5719	5737	5755	5773	5791	5809	5827	5845	5863	5881	5899	5917	5935	5953	5971	5989	6007	6025	6043	6061	6079	6097	6115	6133	6151	6169	6187	6205	6223	6241	6259	6277	6295	6313	6331	6349	6367	6385	6403	6421	6439	6457	6475	6493	6511	6529	6547	6565	6583	6601	6619	6637	6655	6673	6691	6709	6727	6745	6763	6781	6799	6817	6835	6853	6871	6889	6907	6925	6943	6961	6979	6997	7015	7033	7051	7069	7087	7105	7123	7141	7159	7177	7195	7213	7231	7249	7267	7285	7303	7321	7339	7357	7375	7393	7411	7429	7447	7465	7483	7501	7519	7537	7555	7573	7591	7609	7627	7645	7663	7681	7699	7717	7735	7753	7771	7789	7807	7825	7843	7861	7879	7897	7915	7933	7951	7969	7987	8005	8023	8041	8059	8077	8095	8113	8131	8149	8167	8185	8203	8221	8239	8257	8275	8293	8311	8329	8347	8365	8383	8401	8419	8437	8455	8473	8491	8509	8527	8545	8563	8581	8599	8617	8635	8653	8671	8689	8707	8725	8743	8761	8779	8797	8815	8833	8851	8869	8887	8905	8923	8941	8959	8977	8995	9013	9031	9049	9067	9085	9103	9121	9139	9157	9175	9193	9211	9229	9247	9265	9283	9301	9319	9337	9355	9373	9391	9409	9427	9445	9463	9481	9499	9517	9535	9553	9571	9589	9607	9625	9643	9661	9679	9697	9715	9733	9751	9769	9787	9805	9823	9841	9859	9877	9895	9913	9931	9949	9967	9985	10003	10021	10039	10057	10075	10093	10111	10129	10147	10165	10183	10201	10219	10237	10255	10273	10291	10309	10327	10345	10363	10381	10399	10417	10435	10453	10471	10489	10507	10525	10543	10561	10579	10597	10615	10633	10651	10669	10687	10705	10723	10741	10759	10777	10795	10813	10831	10849	10867	10885	10903	10921	10939	10957	10975	10993	11011	11029	11047	11065	11083	11101	11119	11137	11155	11173	11191	11209	11227	11245	11263	11281	11299	11317	11335	11353	11371	11389	11407	11425	11443	11461	11479	11497	11515	11533	11551	11569	11587	11605	11623	11641	11659	11677	11695	11713	11731	11749	11767	11785	11803	11821	11839	11857	11875	11893	11911	11929	11947	11965	11983	12001	12019	12037	12055	12073	12091	12109	12127	12145	12163	12181	12199	12217	12235	12253	12271	12289	12307	12325	12343	12361	12379	12397	12415	12433	12451	12469	12487	12505	12523	12541	12559	12577	12595	12613	12631	12649	12667	12685	12703	12721	12739	12757	12775	12793	12811	12829	12847	12865	12883	12901	12919	12937	12955	12973	12991	13009	13027	13045	13063	13081	13099	13117	13135	13153	13171	13189	13207	13225	13243	13261	13279	13297	13315	13333	13351	13369	13387	13405	13423	13441	13459	13477	13495	13513	13531	13549	13567	13585	13603	13621	13639	13657	13675	13693	13711	13729	13747	13765	13783	13801	13819	13837	13855	13873	13891	13909	13927	13945	13963	13981	13999	14017	14035	14053	14071	14089	14107	14125	14143	14161	14179	14197	14215	14233	14251	14269	14287	14305	14323	14341	14359	14377	14395	14413	14431	14449	14467	14485	14503	14521	14539	14557	14575	14593	14611	14629	14647	14665	14683	14701	14719	14737	14755	14773	14791	14809	14827	14845	14863	14881	14899	14917	14935	14953	14971	14989	15007	15025	15043	15061	15079	15097	15115	15133	15151	15169	15187	15205	15223	15241	15259	15277	15295	15313	15331	15349	15367	15385	15403	15421	15439	15457	15475	15493	15511	15529	15547	15565	15583	15601	15619	15637	15655	15673	15691	15709	15727	15745	15763	15781	15799	15817	15835	15853	15871	15889	15907	15925	15943	15961	15979	15997	16015	16033	16051	16069	16087	16105	16123	16141	16159	16177	16195	16213	16231	16249	16267	16285	16303	16321	16339	16357	16375	16393	16411	16429	16447	16465	16483	16501	16519	16537	16555	16573	16591	16609	16627	16645	16663	16681	16699	16717	16735	16753	16771	16789	16807	16825	16843	16861	16879	16897	16915	16933	16951	16969	16987	17005	17023	17041	17059	17077	17095	17113	17131	17149	17167	17185	17203	17221	17239	17257	17275	17293	17311	17329	17347	17365	17383	17401	17419	17437	17455	17473	17491	17509	17527	17545	17563	17581	17599	17617	17635	17653	17671	17689	17707	17725	17743	17761	17779	17797	17815	17833	17851	17869	17887	17905	17923	17941	17959	17977	17995	18013	18031	18049	18067	18085	18103	18121	18139	18157	18175	18193	18211	18229	18247	18265	18283	18301	18319	18337	18355	18373	18391	18409	18427	18445	18463	18481	18499	18517	18535	18553	18571	18589	18607	18625	18643	18661	18679	18697	18715	18733	18751	18769	18787	18805	18823	18841	18859	18877	18895	18913	18931	18949	18967	18985	19003	19021	19039	19057	19075	19093	19111	19129	19147	19165	19183	19201	19219	19237	19255	19273	19291	19309	19327	19345	19363	19381	19399	19417	19435	19453	19471	19489	19507	19525	19543	19561	19579	19597	19615	19633	19651	19669	19687	19705	19723	19741	19759	19777	19795	19813	19831	19849	19867	19885	19903	19921	19939	19957	19975	19993	20011	20029	20047	20065	20083	20101	20119	20137	20155	20173	20191	20209	20227	20245	20263	20281	20299	20317	20335	20353	20371	20389	20407	20425	20443	20461	20479	20497	20515	20533	20551	20569	20587	20605	20623	20641	20659	20677	20695	20713	20731	20749	20767	20785	20803	20821	20839	20857	20875	20893	20911	20929	20947	20965	20983	21001	21019	21037	21055	21073	21091	21109	21127	21145	21163	21181	21199	21217	21235	21253	21271	21289	21307	21325	21343	21361	21379	21397	21415	21433	21451	21469	21487	21505	21523	21541	21559	21577	21595	21613	21631	21649	21667	21685	21703	21721	21739	21757	21775	21793	21811	21829	21847	21865	21883	21901	21919	21937	21955	21973	21991	22009	22027	22045	22063	22081	22099	22117	22135	22153	22171	22189	22207	22225	22243	22261	22279	22297	22315	22333	22351	22369	22387	22405	22423	22441	22459	22477	22495	22513	22531	22549	22567	22585	22603	22621	22639	22657	22675	22693	22711	22729	22747	22765	22783	22801	22819	22837	22855	22873	22891	22909	22927	22945	22963	22981	22999	23017	23035	23053	23071	23089	23107	23125	23143	23161	23179	23197	23215	23233	23251	23269	23287	23305	23323	
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

- 1) Запустите программу Excel. Создайте таблицу по образцу (см. рисунок).
- 2) Установите условное форматирование ячеек диапазона В3:Е7 в соответствии с данными о призовых местах:
 - «1-е место» — красная заливка;
 - «2-е место» — зеленая заливка;
 - «3-е место» — желтая заливка (цвет шрифта — по своему усмотрению).
- 3) Установите условное форматирование ячеек с данными о непризовых местах: шрифт — полужирный, цвет шрифта — синий, без заливки ячеек.
- 4) Отсортируйте строки таблицы так, чтобы учебные предметы располагались в алфавитном порядке.
- 5) Установите фильтры для отображения тех предметов, по которым учащиеся школы в 2015 году заняли первое место.
- 6) Сохраните файл с именем Упражнение 18 в своей папке. Завершите работу за компьютером.

	A	B	C	D	E
1	Достижение (место)				
2	Предмет	2013	2014	2015	2016
3	Математика	2	4	2	3
4	Физика	4	4	3	3
5	Химия	3	2	1	2
6	Биология	2	4	7	5
7	Информатика	3	1	1	3

Компьютерное тестирование



Выполните тестовое задание 18 с автоматической проверкой на сайте interactive.ranok.com.ua.

Практическая работа 9



Использование математических, логических и статистических функций табличного процессора. Условное форматирование

Задание: создать таблицу успеваемости учащихся в виде школьного журнала с автоматическим вычислением итоговых оценок и уровня учебных достижений.

Оборудование: компьютер с установленным табличным процессором Excel.

Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

35	52	70	87	95	122	140	157	0,0000	9	3	6	9	5*	0,0872	889	96	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	86*	3	6	9	8*	1401	1418	1435	1452	1469	1486
732	750	767	785	802	819	837	854	0,0872	85*	3	6	9	9*	1583	1600	1617	1634	1651	1668

Практическая работа 9

- ▶ 1. Запустите программу Excel. Установите параметры страницы: все поля — по 1 см, колонтитулы — 0 см, ориентация — книжная, размер бумаги — А4.
- ▶ 2. Создайте таблицу по образцу (см. рисунок). Объедините нужные ячейки, измените размеры соответствующих столбцов и строк. Установите для границ таблицы тип линий средней толщины.
- ▶ 3. Установите в ячейках заданную ориентацию текста и форматы данных: для оценок — числовой без десятичных знаков, для надписей — текстовый, для дат — короткий формат даты.
- ▶ 4. Введите фамилии учащихся и их оценки, а также другие текстовые данные.
- ▶ 5. Отсортируйте строки так, чтобы фамилии учащихся расположились в алфавитном порядке.
- ▶ 6. Введите нумерацию и даты в режиме автозаполнения.
- ▶ 7. Введите формулу для итоговой оценки каждого учащегося: среднее арифметическое его оценок, округленное до целого значения.
- ▶ 8. Рассчитайте количество отсутствующих в каждый из указанных дней, используя функцию СЧЁТЕСЛИ(COUNTIF).
- ▶ 9. По итоговым оценкам рассчитайте количество учащихся, оценки которых соответствуют начальному (менее 4 баллов) и высокому (более 9 баллов) уровням, используя функцию СЧЁТЕСЛИ(COUNTIF).
- ▶ 10. По итоговым оценкам учащихся рассчитайте количество тех, у кого достаточный (от 7 до 9 баллов) и средний (от 4 до 6 баллов) уровни, используя функцию СЧЁТЕСЛИМН(COUNTIFS).
- ▶ 11. Настройте условное форматирование ячеек с оценками в зависимости от уровня учебных достижений: высокий — красный цвет текста, средний — зеленый, достаточный — синий, начальный — черный (без заливки ячеек цветом).
- ▶ 12. Сохраните файл с именем Практическая работа 9 в своей папке. Завершите работу за компьютером.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Дата										
2	№ п/п	Фамилия	9.2	16.2	23.2	1.3	8.3	15.3					Итоговая
3	1	Иванова	6	8	4	4	5	н	н	н		5	
4	2	Петренко	12	10	12	9	н	н		н			
5	3	Сидоров	10		9	н	8						
6	4	Тесленко	11	10	9	н	8			8			
7	5	Айсман	5	7	8		6						
8	6	Бою				12	12		12	н			
9	7	Грицюк		9	8					н			
10	8	Буренко	н	12			12	9					
11	9	Вивчар	н				8	н		8	8		
12		отсутствующих											
13		начальный уровень											
14		достаточный уровень											
15		средний уровень											
16		высокий уровень											

Сделайте вывод: какие функции используются для расчетов; для чего предназначено условное форматирование.

18*	24	
	99	958
924	1111	1132
1097	1115	1305
1271	1288	1478
	1461	1478

§ 19. Расширенные фильтры. Промежуточные итоги

Расширенные фильтры

Использование автоматических фильтров, рассмотренных в предыдущем параграфе, не всегда позволяют решить стоящую перед пользователем задачу.

▶ Пусть требуется получить данные об успеваемости учащихся, имеющих оценки более 9 баллов по алгебре или информатике. Применение фильтров с условием (> 9) отдельно к столбцам Алгебра, Информатика позволит увидеть только строку Петренко, тогда как заданным критериям соответствуют строки: Иванова, Петренко, Исханова (рис. 19.1).

	A	B	C	D	E	F
1	Фамилия	Алгебра	Химия	Физика	Информатика	Итоговая
2	Иванова	7	10	11	12	10,0
3	Петренко	12	11	8	10	10,3
4	Исханова	11	9	8	9	9,3
5	Тесленко	9	8	9	7	8,3

Рис. 19.1

Для реализации сложных условий фильтрации используют **расширенные фильтры**.

Для применения расширенного фильтра необходимо:

- 1) скопировать в свободное на листе место заголовки столбцов, по которым будет осуществляться фильтрация (например, Алгебра, Информатика);
- 2) в ячейки под скопированными заголовками ввести критерии фильтрации (например, >9). При этом критерии, объединенные логической операцией ИЛИ, указываются в разных строках, а объединенные логической операцией И, — в одной строке;
- 3) выбрать Данные → Сортировка и фильтр → Дополнительно;
- 4) установить в окне Расширенный фильтр параметр фильтрации (рис. 19.2).

Чтобы **копировать результат в другое место**, достаточно в соответствующем поле указать адрес верхней левой ячейки новой таблицы с результатами фильтрации (рис. 19.2).



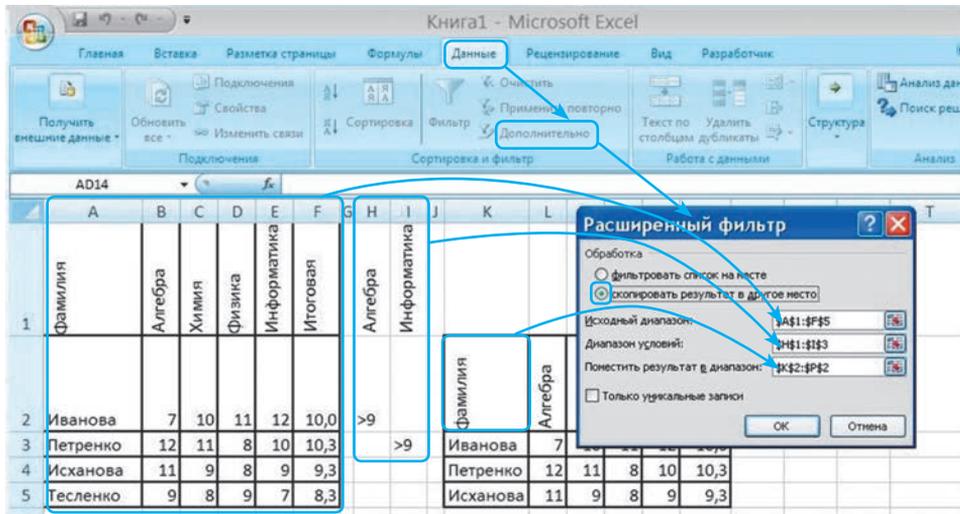


Рис. 19.2

Промежуточные итоги

В таблицах Excel удобно анализировать данные. Одним из способов является применение *промежуточных итогов*, когда подсчет проводится не по всем данным столбца, а по частям.

➤ Пусть требуется вычислить с помощью промежуточных итогов, сколько учащихся учатся по информатике на 10, 11, 12 баллов (рис. 19.3).

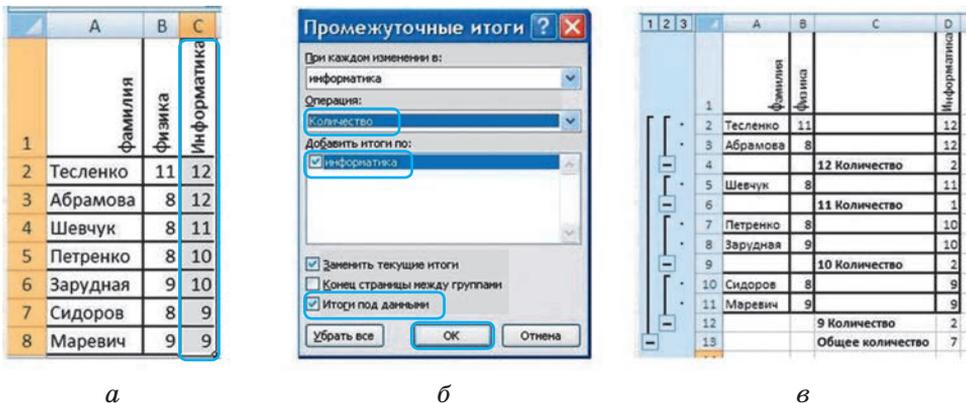


Рис. 19.3

Чтобы **настроить промежуточные итоги**, нужно отсортировать таблицу по столбцу, данные которого используются в вычислениях (в нашем случае — столбец Информатика); выделить ячейки с данными и заголовками (рис. 19.3, а); выполнить команду Данные → Структура → Промежуточный итог; установить критерии в окне Промежуточные итоги (рис. 19.3, б). При помощи кнопок, появившихся слева от таблицы (рис. 19.3, в), данные можно скрыть, оставив результаты промежуточных итогов.

Вопросы для самопроверки



1. В каких случаях применяют автофильтр?
2. Для решения каких задач целесообразно использовать расширенный фильтр?
3. Опишите порядок установки расширенного фильтра.
4. Как в расширенном фильтре записываются условия (критерии), объединенные логической операцией И?
5. Когда целесообразно использовать промежуточные итоги?

Упражнение 19



▶ Создать таблицы с применением расширенного фильтра и промежуточных итогов по описанию.

- 1) Запустите программу Excel. Создайте таблицу по образцу.
- 2) Примените к созданной таблице расширенный фильтр с копированием результата в новую таблицу, в которой будут данные о грузовых автомобилях или тех, объем продаж которых превысил 1 000 000.
- 3) Для исходной таблицы, используя промежуточные итоги, вычислите суммарные объемы продаж автомобилей каждого типа.
- 4) Скройте данные исходной таблицы, оставив промежуточные итоги продаж автомобилей по типам.
- 5) Сохраните файл с именем Упражнение 19 в своей папке. Завершите работу за компьютером.

	А	В	С
1	Тип	Марка	Объемы продаж
2	легковые	ЗАЗ	70 000
3	грузовые	МАЗ	500 000
4	легковые	Мерседес	3 000 000
5	грузовые	Мерседес	10 000 000
6	грузовые	Форд	750 000
7	тягачи	Форд	1 000 000
8	легковые	Форд	15 000 000
9	легковые	Хонда	1 500 000

Компьютерное тестирование



Выполните тестовое задание 19 с автоматической проверкой на сайте interactive.ranok.com.ua.

35	52	70	87	105	122	140	157	0,0000	9	3	6	9	5*	0,0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	86*	3	6	9	8*	1401	1418	1435	1452	1469	1486
732	750	767	785	802	819	837	854	0,0872	85*	3	6	9	9*	1584	1601	1618	1635	1652	1669

§ 20. Диаграммы и графики

§ 20. Диаграммы и графики

Диаграммы

В программе Excel имеются специальные средства, называемые деловой графикой, которые позволяют представить табличные данные в графическом виде.

Для визуального представления данных в Excel есть большое количество стандартных диаграмм, в том числе круговая, столбчатая (гистограмма) и график (рис. 20.1).

Круговая диаграмма (рис. 20.2, а) наглядно демонстрирует соотношение между частями в целом. Для построения круговой диаграммы используют числовые данные, размещенные в одной строке или столбце.

Столбчатая диаграмма (рис. 20.2, б) иллюстрирует соотношение между числовыми данными, размещенными в нескольких строках и столбцах.

График (рис. 20.2, в) удобно использовать для изображения изменений показателей в течение определенного времени.

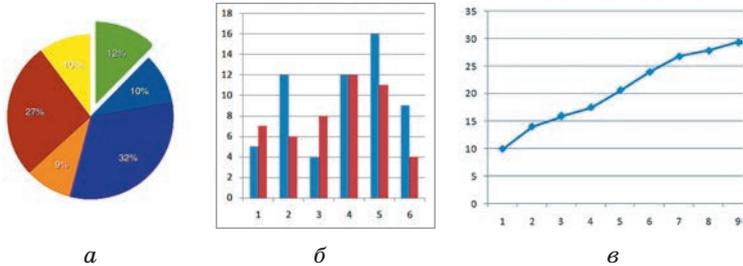
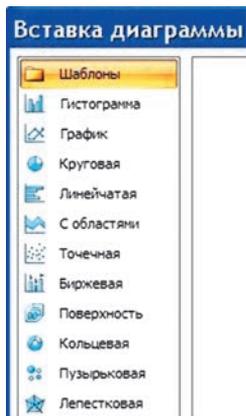


Рис. 20.1

Рис. 20.2

Диаграммы, построенные по данным таблицы, меняются в соответствии с изменениями данных в таблице.

Построение диаграммы

Для создания диаграммы необходимо:

- 1) заполнить (создать) таблицу данных;
- 2) выделить нужные данные вместе с названиями строк (столбцов);
- 3) выполнить команду Вставка → Диаграмма и выбрать тип диаграммы.

18*	24	30
924	941	958
1097	1115	1132
1271	1288	1305
1461	1478	1495

Рассмотрим объекты диаграммы (рис. 20.3).



Рис. 20.3

С помощью инструментов Конструктор или кнопок быстрой настройки можно изменить тип диаграммы, макет, стиль оформления:

Изображение кнопки	Назначение быстрой настройки
Элементы	Редактирование названия, легенды, подписей
Стили диаграмм	Настройка стиля и цветовой гаммы
Фильтры	Выбор источника данных

Диаграмма может быть размещена на одном листе с таблицей данных, а может быть перенесена на отдельный лист. Чтобы **переместить диаграмму в другое место листа**, следует перетянуть область диаграммы с нажатой левой кнопкой мыши. Чтобы разместить диаграмму на отдельном листе, нужно выполнить команду Переместить диаграмму контекстного меню.

Чтобы **изменить размеры диаграммы** или ее объектов, следует выделить (щелкнуть) объект и перетянуть соответствующие маркеры изменения размера. Секторы круговой диаграммы, например, можно разъединить, перетягивая их мышью. С помощью контекстного меню можно изменить цвет заливки секторов, их контур, размер и т. п.

Графики

Рассмотрим ход построения графика функции в Excel.

➤ Построить график функции $y = \frac{2}{x}$ при $1 \leq x \leq 19$ с шагом $h = 1,5$.

Создадим в Excel таблицу значений функции на заданном промежутке. Для этого:

1. Впишем в ячейках A1, A2, A3 заголовки строк — h, x, y.
2. В ячейку B1 введем значение шага изменения аргумента x ($h = 1,5$).
3. В ячейку B2 введем первое значение из заданного диапазона x ($x = 1$).

35	52	70	87	105	122	140	157	0,0000	9	3	6	9	5*	0,0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	86*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	88*	3	6	9	8*	1461	1478	1495	1512	1529	1546
732	750	767	785	802	819	837	854	0,0872	89*	3	6	9	9*	1703	1720	1737	1754	1771	1788

§ 20. Диаграммы и графики

- В ячейку C2 введем формулу для вычисления текущего значения аргумента x ($=B2+\$B\1).
- Скопируем формулу из ячейки C2 в остальные ячейки диапазона координаты x , пока не получим конечное значение $x=19$.
- В ячейку B3 введем формулу для вычисления значения функции ($=2/B2$).
- Установим для диапазона ячеек B3:N3 числовой формат с двумя десятичными знаками.
- Скопируем формулу из ячейки B3 в ячейки диапазона C3:N3.
- В полученной таблице выделим все значения вместе с буквами x и y — заголовками строк (диапазон B2:N3).
- Выберем на вкладке Вставка в группе Диаграммы тип диаграммы — точечная. Получим искомый график (рис. 20.4).

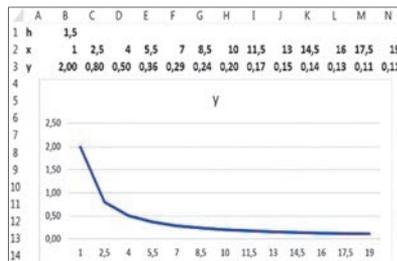


Рис. 20.4

Вопросы для самопроверки



- Какие типы диаграмм можно создать в Excel?
- Когда целесообразно использовать круговую диаграмму?
- Когда следует использовать столбчатую диаграмму?
- Каковы основные шаги построения диаграммы?
- Как изменить размер диаграммы или ее объектов?

Упражнение 20



- Построить график функции $y = \frac{a}{x}$ ($a = \text{const}$ для произвольного диапазона x).
- Запустите программу Excel. Для переменной a отведите отдельную ячейку и введите произвольное значение.
 - Введите в отдельную ячейку произвольное значение шага h аргумента x . Создайте таблицу значений функции по заданным значениям аргумента.
 - Измените значение a и проанализируйте изменение графика.
 - Сохраните файл с именем Упражнение 20 в своей папке. Завершите работу за компьютером.

Компьютерное тестирование



Выполните тестовое задание 20 с автоматической проверкой на сайте interactive.ranok.com.ua.

18*	24	30*
924	105	958
1097	1115	1132
1271	1288	1305
1461	1478	1495



Практическая работа 10

Упорядочение данных в таблицах.

Автофильтр и расширенный фильтр

Задание: создать таблицу по образцу, упорядочить данные в соответствии с заданными условиями и правилами.

Оборудование: компьютер с установленным табличным процессором Excel.

Ход работы

Во время работы с компьютером соблюдайте правила безопасности.

- ▶ 1. Запустите программу Excel. Создайте таблицу (Лист1) по образцу (см. рисунок). Скопируйте таблицу еще на пяти листах рабочей книги.
- ▶ 2. Отсортируйте данные в таблице по возрастанию цены (Лист2).
- ▶ 3. Отсортируйте строки таблицы так, чтобы марки автомобилей были расположены в алфавитном порядке, а их количество — в порядке убывания (Лист3).
- ▶ 4. Установите автофильтры для всех столбцов таблицы и настройте автофильтр соответствующего столбца на отображение строк только марок автомобилей «Форд» и «Мерседес» (Лист4).
- ▶ 5. Настройте в таблице расширенный фильтр отображения данных автомобилей, цена которых меньше 123 000 или количество которых не превышает 150 (Лист5).
- ▶ 6. Установите условное форматирование данных таблицы так, чтобы надпись «Мерседес» отображалась красным цветом, а «Форд» — зеленым (Лист6).
- ▶ 7. Сохраните файл с именем Практическая работа 10 в своей папке. Завершите работу за компьютером.

Цена, грн	Марка автомобиля	Количество, шт.
120 000	МАЗ	100
150 000	Форд	125
110 000	Форд	150
110 000	Мерседес	200
100 000	Хонда	225
120 500	Мерседес	250

Сделайте вывод: как можно упорядочить данные в таблице; какие фильтры можно применять; для чего используют условное форматирование.

РАЗДЕЛ 6

ОСНОВЫ СОБЫТИЙНО- И ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ



§ 21. Знакомство со средой программирования

§ 22. Создание простейшего проекта

§ 23. Создание программы

§ 24. Основные компоненты программы

§ 25. Свойства и методы элементов управления

§ 26. Отладка программного кода

Практическая работа 11. Создание объектно-ориентированной программы, отображающей окно сообщения

Практическая работа 12. Создание программы с кнопками и надписями

ПОВТОРЯЕМ

В 5–7 классах вы знакомились с алгоритмами и составляли программы в учебной программной среде.

Алгоритм — это четко определенная последовательность команд для решения определенной задачи. Алгоритмы можно представлять словесным и графическим способами, а также в виде программы, записанной на определенном языке программирования.

Язык программирования — это искусственный язык, представляющий собой систему обозначений и правил для записи алгоритмов в форме, пригодной для их выполнения на компьютере.

Для удобной разработки программ существуют специальные средства создания — системы программирования.



1. Что такое программа?
2. Что такое программирование?
3. Что такое язык программирования?
4. С какими языками программирования вы уже знакомы?
5. Что такое система программирования?



Освоив материал этого раздела, вы научитесь создавать программы с интересным интерфейсом, используя возможности среды с визуальным конструктором интерфейса пользователя.

§ 21. Знакомство со средой программирования

Материал учебника ориентирован на работу со средой программирования Lazarus*, в основе которого лежат принципы объектно-ориентированного программирования. Объектно-ориентированное программирование — это метод программирования, основанный на представлении программы в виде совокупности взаимодействующих объектов. Каждый из объектов характеризуется состоянием, которое описывается с помощью свойств, и поведением, определяемым установленными действиями. При создании программы используются готовые компоненты, свойства и действия которых настраиваются в соответствии с требованиями пользователя.

Среда программирования

В основе среды программирования Lazarus лежит язык программирования Object Pascal, берущий свое начало от классического языка Pascal (Паскаль). Одна из распространенных сред программирования Delphi тоже основана на этом языке. Язык программирования Pascal был разработан в 1971 году профессором Института информатики швейцарской высшей политехнической школы Никлаусом Виртом (род. 1934) и назван в честь Блеза Паскаля (1623–1662).

Среда программирования Lazarus — это комплекс программных средств, состоящий из текстового редактора, компилятора, отладчика. Среда программирования имеет графический интерфейс, содержащий набор меню, панелей и окон для разработки программ.

При работе в среде программирования Lazarus программист описывает определенную совокупность объектов и их взаимодействие. Каждый объект обладает набором свойств, может выполнять определенные действия и реагировать на указанные события. Действия, которые может выполнять объект, называют его **методами**.

Автоматизация программирования достигается благодаря возможности создавать на форме объект, выбирая соответствующий компонент на Палитре компонентов, менять значения его свойств, не внося вручную изменений в программный код, и вызывать различные его методы.

* Официальный сайт: www.lazarus-ide.org.

Таким образом, технология работы в среде Lazarus базируется на идеях объектно-ориентированного, событийного программирования и визуального конструирования интерфейса пользователя.

Интерфейс среды программирования Lazarus

Для **открытия среды** Lazarus нужно выполнить команды:
Пуск → Программы → Lazarus → Lazarus.

Интерфейс среды программирования Lazarus содержит четыре основных окна (рис. 21.1).

Главное окно, на котором расположены три панели: меню, панель быстрого доступа, палитра компонентов.

Окно формы — заготовка окна будущей программы. *Форма* — это главный компонент программы, контейнер для всех других ее компонентов.

Окно Инспектор объектов — окно, в котором настраивают свойства размещенных на форме объектов.

Окно Редактор исходного кода — окно текстового редактора, предназначенное для ввода и редактирования программного кода.

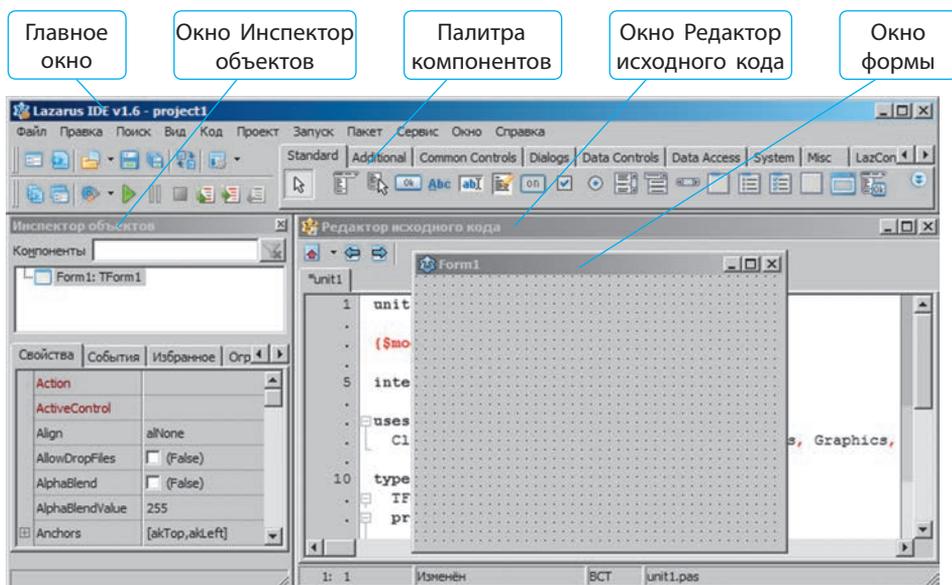


Рис. 21.1

Меню главного окна программы состоит из таких пунктов: Файл, Правка, Поиск и др., в которых команды сгруппированы по назначению. Меню (рис. 21.2) используют для открытия и закрытия файлов, настройки среды программирования и пр. На Панели быстрого доступа расположены значки наиболее часто используемых команд меню. Палитра компонентов содержит вкладки Standard, Additional, Dialogs и др., на которых компоненты сгруппированы по назначению. С помощью Палитры компонентов выбирают нужные компоненты для размещения их на форме.

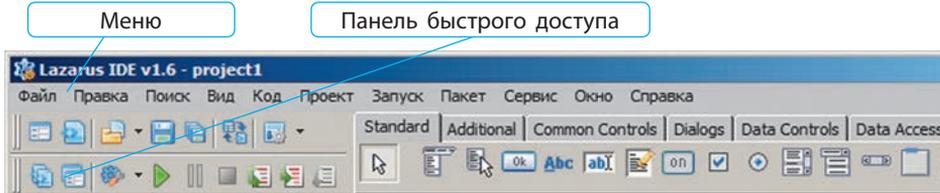


Рис. 21.2

Знакомство с объектами программы

Объектами программы являются форма и содержащиеся на ней объекты — компоненты с настроенными свойствами и методами.

Для добавления компонента на форму нужно:

- 1) щелкнуть мышью значок компонента на Палитре компонентов;
- 2) щелкнуть мышью в любом месте на форме.

Выбранный компонент появится на форме, после чего его нужно настроить.

» Ознакомьтесь с компонентом Label. Добавим на форму компонент Label **Аbc** (*Надпись*) из Палитры компонентов. На форме появится объект Label1, который можно передвигать по форме и менять его размеры с помощью маркеров. Менять вид и поведение объекта можно с помощью окна Инспектор объектов. В этом окне расположены вкладки Свойства и События, содержимое которых отличается для разных объектов (рис. 21.3).

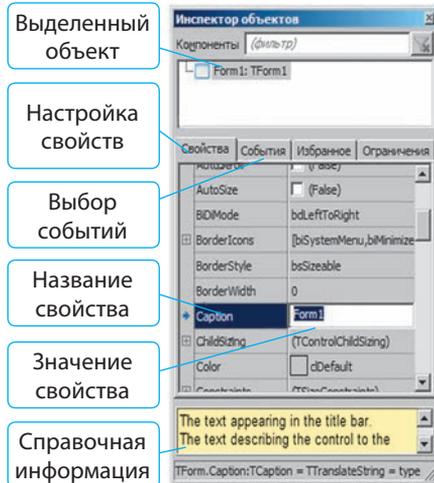


Рис. 21.3

Для объекта Label1 на вкладке Свойства вы можете определить расположение и размер, меняя значение свойств Left (*Расстояние от левого края формы*), Top (*Расстояние от верхнего края формы*), Height (*Высота*) и Width (*Ширина*). На вкладке События содержится список событий, на которые может реагировать данный объект.

При разработке программы в среде программирования Lazarus на форму, содержащуюся в окне формы, добавляют компоненты будущей программы из Палитры компонентов. Каждый из компонентов настраивают как отдельный объект программы.

Вопросы для самопроверки



1. Назовите основные окна интерфейса среды программирования Lazarus.
2. Для чего предназначена Палитра компонентов?
3. Для чего предназначено окно Инспектор объектов?
4. Как добавить компонент на форму?
5. Как изменить расположение объекта на форме?
6. Как изменить свойства объекта на форме?

Упражнение 21



- Ознакомиться с последовательностью действий для настройки свойств формы.
- 1) Откройте среду Lazarus, выбрав команды: Пуск → Программы → Lazarus → Lazarus.
 - 2) Выделите форму: щелкните на ней или выберите объект Form1 в окне Инспектор объектов.
 - 3) Измените размеры и расположение формы с помощью мыши.
 - 4) Перейдите в окно Инспектор объектов. Задайте значение свойств Width и Height по 500 пикселей.
 - 5) В окне Инспектор объектов измените заголовок формы. Для этого в строке свойства Caption (*Заголовок*) замените название объекта Form1 на «Упражнение 21».
 - 6) Измените цвет формы с помощью свойства Color.

Компьютерное тестирование



Выполните тестовое задание 21 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 22. Создание простейшего проекта

Понятие проекта

Для решения задач пользователя существует много прикладных программ (приложений). Прикладная программа, как правило, состоит из множества файлов, каждый из которых имеет определенное назначение. Набор файлов в определенной среде программирования, необходимых для создания прикладной программы, называют **проектом**.

Основные файлы проекта

Файлы проекта	Ярлык	Тип файла	Назначение
Главный файл проекта (Lazarus Project Information)		.lpi	Подключает все используемые программные модули и содержит операторы для запуска программы
Файл основной программы		.lpr	Содержит код основной программы, подключает внешние библиотеки, все необходимые модули (создается вместе с файлом .lpi)
Файлы описания форм		.lfm	Сохраняют значения свойств форм
Файлы программных модулей		.pas	Содержат коды программных модулей

В среде программирования Lazarus при компиляции из файлов проекта формируется единый исполняемый файл и создается новое приложение, предназначенное для решения определенной задачи.

Создание проекта

Для создания проекта нужно выполнить команду: меню Проект → Создать проект... → Проект → Приложение (рис. 22.1). На экране появится окно формы и окно Редактор исходного кода.

На форме можно разместить различные объекты: надписи (Label), кнопки (Button), текстовые поля (Edit) и прочие

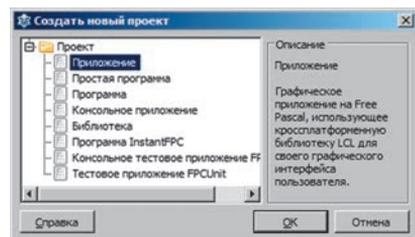


Рис. 22.1

и настроить их для управления программой. Созданные таким образом объекты называют также **элементами управления**.

Рассмотрим подробнее процесс создания элемента управления Label. Этот элемент управления используют для вывода заголовков, подписей других элементов управления и т. п.

Для **добавления компонента Label на форму** нужно:

- 1) выбрать компонент Label из вкладки Standard на Палитре компонентов (рис. 22.2) и щелкнуть на нем мышью;
- 2) щелкнуть мышью в любом месте на форме.

На форме созданный таким образом объект содержит надпись Label1 (рис. 22.3).

На вкладке Standard



Рис. 22.2

На форме

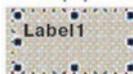


Рис. 22.3

Для изменения значения свойств созданного объекта Label1 следует перейти в окно Инспектор объектов. На вкладке Свойства находятся все свойства объекта, в данном случае свойства надписи (рис. 22.4). Можно изменить значение свойств Caption (*Заголовок*), Color (*Цвет фона*), Font (*Параметры шрифта*). С тем, как настроить события для элемента управления, вы ознакомитесь в § 24.

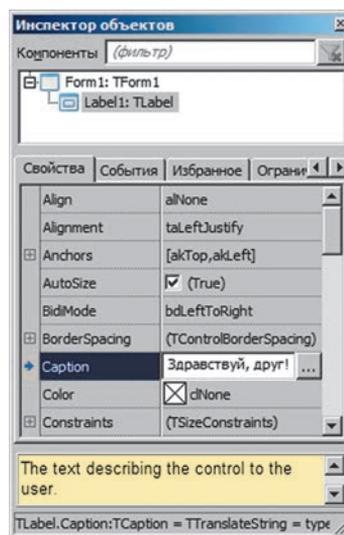


Рис. 22.4

Компиляция проекта и его выполнение

Для создания единого исполняемого файла проекта нужно осуществить компиляцию.



Компилятор — это специальная программа, которая преобразует исходную программу, написанную на языке программирования высокого уровня, в программу, представленную на машинном языке, «понятном» компьютеру.

Для компиляции проекта в среде Lazarus нужно выбрать команду меню Запуск → Компилировать. Компилятор последовательно обрабатывает файлы проекта и строит из них **исполняемый файл**.

При компиляции текст программы проверяется на отсутствие ошибок, переводится на машинный язык, после чего устанавливаются связи между файлами проекта и формируется ехе-файл (исполняемый файл).

Если при компиляции обнаружится, что текст программы содержит синтаксические ошибки, то в нижней части окна Редактор исходного кода выводятся сообщения об ошибках.

Если в программе нет синтаксических ошибок — создается файл, готовый к выполнению. Название такого файла совпадает с названием файла проекта и имеет расширение ехе. Среда Lazarus помещает исполняемый файл в папку, содержащую файл проекта.

Запустить проект на выполнение можно разными способами: с помощью команды меню **Запуск** → **Запустить**; щелкнув кнопку  на Панели быстрого доступа (рис. 22.5) или нажав клавишу F9.

Остановить выполнение проекта можно: щелкнув кнопку **Заккрыть**  в окне программы; с помощью кнопки **Остановить** (см. рис. 22.5) на Панели быстрого доступа или нажав комбинацию клавиш **Ctrl + F2**.

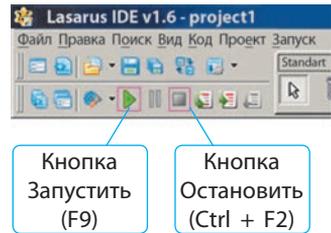


Рис. 22.5

Сохранение проекта

При первом сохранении проекта нужно:

- 1) выбрать команду: меню **Файл** → **Сохранить все** (или щелкнуть кнопку **Сохранить все**  на Панели быстрого доступа);
- 2) в открывшемся диалоговом окне **Сохранить проект project1(*.lpi)**, создать новую папку для проекта;
- 3) сохранить в этой папке файл программного модуля (Unit1.pas) и файл проекта (project1.lpi).

Среда программирования предлагает для сохранения стандартное название проекта, но вы можете его изменить.

Вопросы для самопроверки



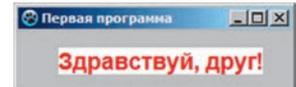
1. Из каких файлов состоит проект в среде Lazarus?
2. Укажите последовательность добавления компонента на форму и изменения его свойств.
3. Как выполняется компиляция программы?
4. Какими способами можно запустить проект на выполнение?

5. Какими способами можно остановить выполнение проекта?
6. Укажите порядок действий для сохранения проекта.

Упражнение 22



- ▶ Создать проект по заданному алгоритму.
- 1) Создайте новый проект, выполнив команду: меню Проект → Создать проект... → Проект → Приложение.
 - 2) В окне Инспектор объектов измените заголовок формы (значение свойства Caption) на «Первая программа».
 - 3) Разместите на форме компонент Label. Измените значение свойства Caption на «Здравствуй, друг!» (см. рисунок).
 - 4) Измените значения свойств Color и Font по собственному усмотрению.
 - 5) Запустите проект на выполнение. Открывшееся окно имеет все атрибуты, присущие окнам программ Windows.
 - 6) Остановите выполнение проекта. Сохраните проект, выполнив команду: меню Файл → Сохранить все. В открывшемся диалоговом окне Сохранить проект project1(*.lpi) создайте новую папку Упражнение 22. Сохраните в папке файлы Unit1.pas и project1.lpi, изменив имена файлов по собственному усмотрению.



Компьютерное тестирование



Выполните тестовое задание 22 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 23. Создание программы

Составляющие языка программирования

Основными составляющими любого языка программирования являются алфавит, синтаксис и семантика.



Алфавит языка программирования — это набор символов, которые могут использоваться при создании программ.

В алфавит языка Object Pascal входят следующие символы:

- 1) заглавные и строчные буквы латинского алфавита: A...Z, a...z;
- 2) цифры: 0...9;
- 3) знаки арифметических операций, знаки препинания, специальные символы: + - * / \ = < > () . , ; : @ \$ # _;
- 4) составные символы: <=, >=, < >, :=.

В языке Object Pascal есть ключевые (служебные) слова, которые имеют фиксированный смысл: If, For, while, array и т. д.



Синтаксис языка программирования — совокупность правил построения языковых конструкций (команд, идентификаторов и т. п.) в определенном языке программирования.

Семантика языка программирования — совокупность правил использования языковых конструкций, записанных на языке программирования.

Событийно-ориентированное программирование

Управление выполнением программы в среде программирования Lazarus осуществляется с помощью событий: когда происходит определенное событие (*event*), то выполняется процедура обработки этого события (*event procedure*).

Событие — это любое воздействие на элемент управления в активном окне, в частности, щелчок кнопки мыши или нажатие клавиши клавиатуры. В Lazarus каждое событие имеет название. Например, щелчок мышью — это событие OnClick. Для каждого объекта существует набор стандартных событий, связанных с данным объектом. Чтобы программа выполняла определенные действия как реакцию на состоявшееся событие, программист должен написать процедуру обработки соответствующего события.

Редактор кода

Программный код формы содержится в окне Редактор исходного кода. В ходе редактирования программы возникает необходимость переключения между окном Редактор исходного кода и окном формы. Сделать это можно, щелкнув кнопку Переключить форму/модуль (рис. 23.1) или нажав клавишу F12.

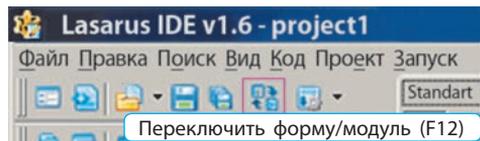


Рис. 23.1

Проект в Lazarus (рис. 23.2) состоит из набора программных единиц — **модулей** (unit). Главный модуль программы создается автоматически и содержит инструкции, с которых начинается выполнение проекта.

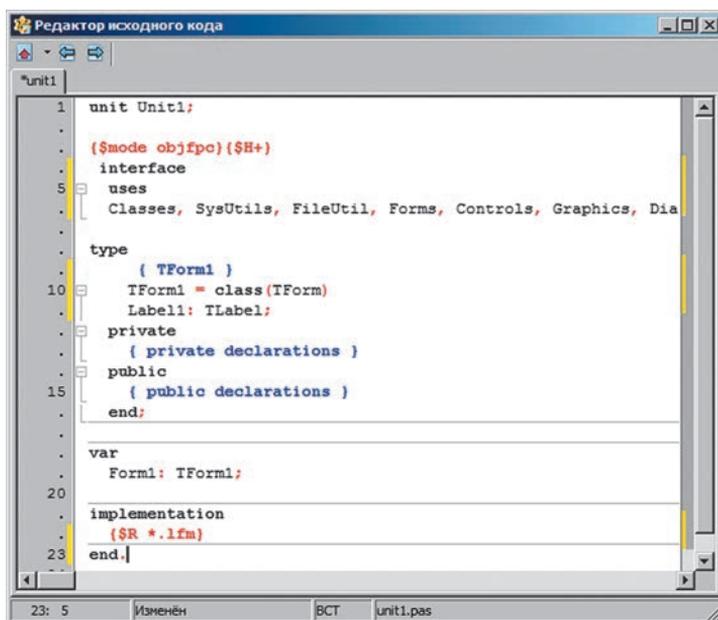


Рис. 23.2

Текстовый редактор, встроенный в среду Lazarus, похож на обычные текстовые редакторы. Но для удобства строки текста программы пронумерованы. Также предусмотрено выделение отдельных составляющих текста программы определенными цветами и начертанием:

- все ключевые слова (**unit**, **uses**, **var**, **end** и др.) автоматически выделяются полужирным;
- знаки препинания становятся красными;
- комментарии выделяются синим цветом;
- строки с ошибками выделяются оранжевыми полосами.

В окне Редактор исходного кода содержатся ключевые слова, образующие структуру программы. Текст модуля начинается с ключевого слова **unit**, далее следует название модуля: **unit Unit1**;

Каждый модуль состоит из двух блоков: интерфейсного (**interface**) и реализации (**implementation**). Интерфейсный блок содержит описание объектов программы, а блок реализации — программный код, который заканчивается ключевым словом **end**.

Создание процедуры обработки события

Для создания процедуры обработки события для объекта нужно:

- 1) выделить объект;
- 2) в окне Инспектор объектов перейти на вкладку События;
- 3) дважды щелкнуть мышью в строке с названием нужного события. Заголовок создаваемой процедуры обработки события имеет формат:

procedure НазваниеОбъектаНазваниеСобытия(Список параметров);

- » Создать процедуру обработки события `OnClick` для командной кнопки.

Добавим на форму компонент `Button` (Кнопка) из вкладки `Standard` Палитры компонентов (рис. 23.3). Автоматически в части описаний появится:

`Button1: TButton;`

Выделим на форме созданную кнопку. В окне Инспектор объектов перейдем на вкладку События. Дважды щелкнем в строке с названием события `OnClick` (рис. 23.4). После этого будет автоматически создана заготовка процедуры обработки события `OnClick` для кнопки `Button1`:

procedure TForm1.Button1Click(Sender: TObject);
begin
end;

Одновременно в части описаний появится заголовок обработчика события:

procedure Button1Click(Sender: TObject);

Запустим проект на выполнение. Щелчок командной кнопки не вызовет никаких действий, поскольку в процедуре `Button1Click` между ключевыми словами **begin** и **end** не записана ни одна команда.

Создадим программный код, с помощью которого при щелчке кнопки будет меняться цвет формы. Запрограммируем изменение значения свойства `Color` объекта `Form1`. Для этого в программном коде между ключевыми словами **begin** и **end** запишем команду:

`Form1.Color := clYellow;`

Запустим проект на выполнение. Проверим действие кнопки.



Рис. 23.3

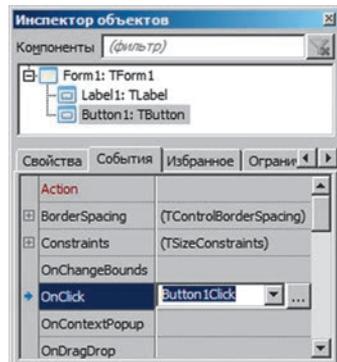


Рис. 23.4

Программа, созданная в среде программирования Lazarus, должна обеспечивать реакцию на события. Процедуру, которая выполняется, когда происходит определенное событие, называют **обработчиком события**. Если в программе нет обработчика какого-либо события, то соответственно нет и реакции программы на данное событие.

Вопросы для самопроверки



1. Объясните, что такое событие. Приведите примеры событий.
2. Почему описанную в параграфе технологию называют событийно-ориентированным программированием?
3. С какого ключевого слова начинается блок реализации процедур обработки событий?
4. Что такое процедура обработки события?
5. Как создать процедуру обработки события OnClick для объекта?

Упражнение 23



- ▶ Составить программу для изменения надписи на форме по щелчку командной кнопки.
- 1) Откройте проект, сохраненный в папке Упражнение 22:
 - выполните команду: меню Проект → Открыть проект...;
 - в диалоговом окне Открыть проект (*.lpi) откройте папку Упражнение 22;
 - найдите в папке файл проекта project1.lpi и откройте его.
 - 2) Добавьте на форму компонент Button из вкладки Standard на Палитре компонентов.
 - 3) Для созданной кнопки (Button1) измените значение свойства Caption на «Изменить надпись».
 - 4) Создайте процедуру обработки события OnClick для кнопки. Перейдите на вкладку События и дважды щелкните в строке с названием события OnClick.
 - 5) В созданной процедуре **procedure** TForm1.Button1Click(Sender: TObject); между ключевыми словами **begin** и **end** запишите команду:


```
Label1.Caption := 'Я программирую в Lazarus';
```

 Запустите проект на выполнение.
 - 6) Добавьте на форму кнопку Button2 и измените ее заголовок на «Изменить цвет». Создайте процедуру обработки события OnClick для кнопки Button2, в которой запрограммируйте изменение цвета формы. Сохраните проект в папке Упражнение 23.

Компьютерное тестирование



Выполните тестовое задание 23 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 24. Основные компоненты программы

Свойства формы

Изучение среды программирования Lazarus вы начали со знакомства с формой. Как и любой объект, форма имеет свойства. Свойствами формы являются заголовок, высота, ширина, место расположения, цвет фона и т. п. В среде программирования Lazarus для формы установлены стандартные значения свойств. Но эти значения можно изменить при проектировании формы в окне Инспектор объектов или предусмотреть их изменение при выполнении программы с помощью операторов, записанных в тексте программы.

Выбрав форму, в окне Инспектор объектов увидим ее свойства.

Основные свойства формы

Свойство	Что определяет
Name	Название (имя) — идентификатор, который используется для обращения к форме в теле программы (по умолчанию первой форме нового проекта присваивается название Form1)
Caption	Текст заголовка окна формы
Width, Height	Ширина и высота формы (в пикселях)
Position	Позиция окна формы на экране при выполнении программы
Color	Цвет фона формы (базовые цвета представлены в списке свойства Color именованными константами)
Hint	Текст подсказки, появляющейся при наведении указателя мыши на объект
ShowHint	Разрешение вывода подсказки

Справа от значений некоторых свойств расположена кнопка . Это значит, что для изменения значения свойства можно воспользоваться дополнительным диалоговым окном.

Перечень объектов, размещенных на форме, выводится в окне Инспектор объектов. Для **выбора объекта** нужно щелкнуть его или выбрать его название в окне Инспектор объектов (рис. 24.1).

В программах полезны всплывающие подсказки.

Для **создания подсказки**, появляющейся при наведении указателя мыши на объект, нужно:

- 1) выделить данный объект;
- 2) ввести текст подсказки в строку свойства Hint (*Подсказка*);
- 3) включить вывод подсказки, установив значение True для свойства ShowHint (*Показать подсказку*).

➤ Создать подсказку, которая будет появляться при наведении указателя мыши на форму.

Выделим объект Form1 в окне Инспектор объектов (рис. 24.2).

Введем текст «Это форма» в строку свойства Hint (см. рис. 24.2).

Для свойства ShowHint установим значение True.

В режиме выполнения проекта при наведении указателя мыши на форму появится подсказка (рис. 24.3).

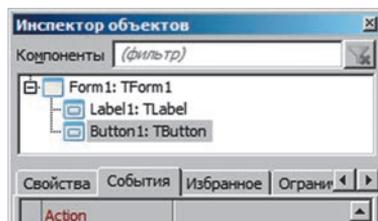


Рис. 24.1

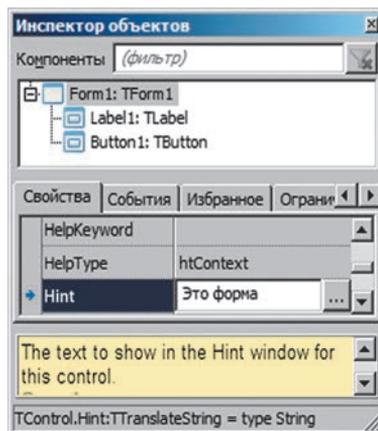


Рис. 24.2



Рис. 24.3

События для формы

Форма, как любой объект программы, имеет свои свойства, методы и события.

Формат названия процедуры обработки событий для формы: `FormНазваниеСобытия`

Рассмотрим события для формы `OnCreate` и `OnClick`. Событие `OnCreate` происходит при загрузке формы. В обработчике события `OnCreate` устанавливаются исходные значения различных свойств объекта.

Вы уже знаете, что событие `OnClick` происходит, когда пользователь щелкает объект мышью.

Заготовка процедуры обработки события `OnClick` в окне Редактор исходного кода имеет вид:

```
procedure TForm1.FormClick(Sender: TObject);
begin
end;
```

Между ключевыми словами `begin` и `end` нужно записать операторы, которые будут определять реакцию формы на щелчок мышью.

Если написать название объекта, в данном случае `Form1`, и поставить точку, раскроется список свойств и методов для объекта `Form1`.

Найдем свойство `Color`. Нажмем букву `C` — в окне выведется список свойств и методов, названия которых начинаются с этой буквы. Выделим строку `Color` (рис. 24.4) и нажмем клавишу `Enter`. В результате получим:

```
Form1.Color
```

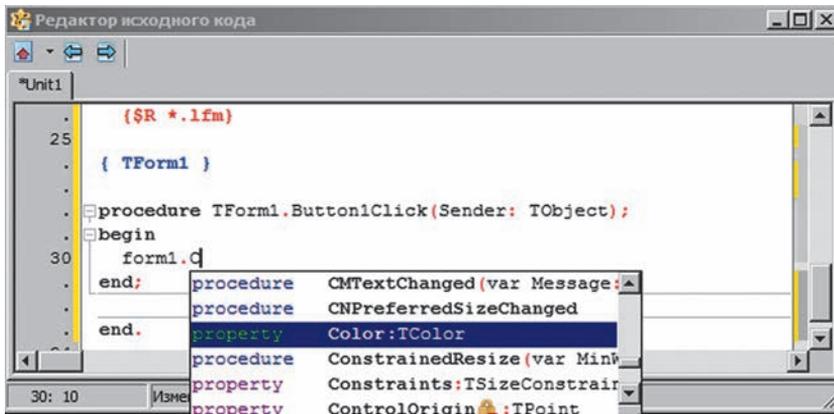


Рис. 24.4

Установим для свойства `Color` желтый цвет:

```
Form1.Color := cYellow; // cYellow — константа, задающая желтый цвет
```

Теперь, если при работе проекта щелкнуть мышью на форме — она станет желтого цвета.

Использование окон сообщений

Окна сообщений используются для предоставления пользователю определенной информации. С помощью процедуры `ShowMessage`

(Показать сообщение) на экран выводится информационное окно с заданным сообщением и кнопкой ОК. Заголовок окна содержит название исполняемого файла.

Синтаксис вызова процедуры:

ShowMessage(Текст);

Здесь Текст — переменная или константа, содержащая текст сообщения.

Чтобы продолжить работу с программой, пользователь должен щелкнуть кнопку ОК. Пока пользователь этого не сделает, дальнейшая работа программы становится невозможной, то есть программа не будет реагировать на другие события.

» Вывести в окне программы сообщение (рис. 24.5):
 ShowMessage('Hello, World');

Окна сообщений можно использовать для вывода результатов выполнения определенного этапа программы. Использование окон сообщений позволяет программисту избежать добавления на форму лишних элементов управления.



Рис. 24.5

Вопросы для самопроверки



1. В каком окне среды программирования Lazarus выводится перечень объектов, размещенных на форме?
2. Как создать подсказку, появляющуюся при наведении указателя мыши на объект?
3. Как изменить заголовок формы?
4. Как создать обработчик события OnClick для формы?
5. Какой оператор предназначен для изменения цвета фона?
6. Приведите примеры, когда целесообразно использовать окно сообщения ShowMessage.

Упражнение 24



- ▶ Составить процедуру для настройки цвета и заголовка формы при запуске проекта на выполнение.
 - 1) Создайте новый проект.
 - 2) Дважды щелкните на форме. В окне Редактор исходного кода откроется заготовка обработчика события OnCreate.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
end;
```

- 3) Настройте красный цвет для фона формы. Между ключевыми словами **begin** и **end** запишите оператор, который задает значение свойства `Color` формы:


```
Form1.Color := clRed;
```
- 4) Запишите оператор изменения заголовка формы:


```
Form1.Caption := 'My program';
```
- 5) Запустите проект на выполнение. При необходимости исправьте ошибки. Сохраните проект в папке Упражнение 24.
- 6) Настройте для формы вывод подсказки «Это форма».



Компьютерное тестирование

Выполните тестовое задание 24 с автоматической проверкой на сайте interactive.ranok.com.ua.

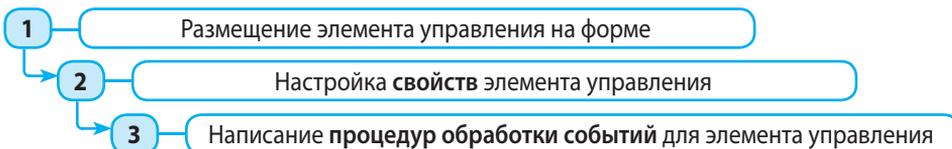


§ 25. Свойства и методы элементов управления

Этапы создания элемента управления

Как вы уже знаете, элементы управления — это визуальные компоненты, каждый из которых имеет определенный набор свойств и методов. Элементы управления — это «строительные блоки»: вы берете их с Палитры компонентов и переносите на форму для создания графического интерфейса пользователя программы. Графический интерфейс позволяет организовать диалог человека и компьютера.

Процесс создания элемента управления делится на три этапа:



Свойства, методы и события

Свойства характеризуют состояние элемента управления, **методы** — действия, которые он может выполнить, а **события** — внешнее воздействие на элемент управления, на который этот элемент может реагировать.



Элемент управления = свойства (состояние) + методы (действия) + события (обратные связи).

Свойства — это атрибуты элемента управления, определяющие, как данный элемент выглядит на экране (например, ширина и высота элемента управления, его отображение и т. п.). Исходные значения свойств элемента управления устанавливаются на этапе разработки интерфейса в окне Инспектор объектов.

Некоторые свойства являются общими для многих элементов управления: Name (*Название*); Enabled (*Доступность*); Height (*Высота*); Width (*Ширина*); Hint (*Подсказка*); Left (*Расстояние от элемента управления до левого края формы*); Top (*Расстояние от элемента управления до верхнего края формы*); Visible (*Видимость*).

Значение свойств элемента управления можно изменять различными способами.

» Изменить размеры кнопки (рис. 25.1) можно путем перетягивания маркеров выделения. Того же результата можно достичь, задав новые значения свойств Height и Width в окне Инспектор объектов.

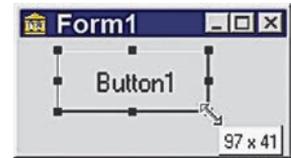


Рис. 25.1

Метод — это фрагмент программного кода, связанный с определенным элементом управления. Запрограммировать вызов метода можно только в программном коде, поэтому он происходит исключительно при выполнении проекта.

Некоторые из методов характерны для многих объектов, например методы setFocus (*Установить фокус*), Show (*Показать*), Hide (*Скрыть*).

» Процедура setFocus передает фокус вводу текстовому полю:
Edit2.SetFocus;

Среда программирования Lazarus позволяет легко менять значения свойств элемента управления как в режиме разработки, так и в режиме выполнения проекта.

Изменение значений свойств элементов управления в режиме выполнения проекта

Для изменения значений свойств элементов управления в режиме выполнения проекта нужно, чтобы программный код содержал необходимые операторы в следующем формате:

НазваниеКомпонента.НазваниеСвойства := ЗначениеСвойства;

» Присвоить свойству Width кнопки значение 40 пикселей:
 Button1.Width := 40;

Следует помнить, что тип свойства и его значение должны совпадать.

Вопросы для самопроверки



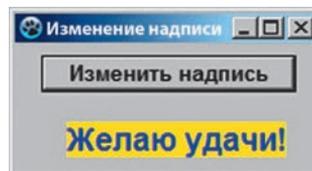
1. Что такое графический интерфейс программы?
2. Назовите этапы создания элемента управления.
3. Что такое свойства элемента управления?
4. Что такое методы элемента управления?
5. Как изменить значение свойства элемента управления при разработке проекта?
6. Как изменить значение свойства элемента управления при выполнении проекта?

Упражнение 25



» Составить программу для изменения свойств элемента управления Label в режиме выполнения проекта.

- 1) Создайте новый проект. Измените заголовков формы на «Изменение надписи». Разместите на форме компоненты Button и Label согласно рисунку.
- 2) Выделите созданную кнопку (Button1) и в окне Инспектор объектов измените значение свойства Font по собственному усмотрению, значение свойства Caption — на «Изменить надпись».
- 3) Измените для надписи Label1 значение свойств Alignment, Autosize, Color. Свойству Caption задайте значение «Я программирую в Lazarus».
- 4) Создайте процедуру обработки события OnClick для кнопки Изменить надпись.



- 5) Запишите операторы для изменения значений свойств:
- ```
Label1.Caption := 'Желаю удачи!';
Label1.Color := clYellow;
Label1.Font.Color := clBlue;
```

! Свойству `Caption` можно задавать только текстовые значения, которые в программном коде заключаются в одинарные кавычки. При установке значения свойства `Caption` в окне Инспектор объектов текст записывают без кавычек.

- 6) Запустите проект на выполнение. Проверьте действие кнопки. Сохраните проект в папке Упражнение 25.



### Компьютерное тестирование

Выполните тестовое задание 25 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 26. Отладка программного кода

Текст программы, введенный с клавиатуры, часто содержит ошибки, которые программист должен устранить. Ошибки, которые могут возникнуть в процессе разработки программы, делят на:

- синтаксические ошибки (англ. *compile error*);
- ошибки периода выполнения (англ. *run-time error*);
- логические (алгоритмические) ошибки.

### Синтаксические ошибки в программном коде

Компилятор создает исполняемый файл лишь в случае, если текст не содержит ошибок. Большинство ошибок, которые обнаруживаются при компиляции программы, являются синтаксическими. Это ошибки в написании или размещении ключевых слов и других элементов кода. При выявлении ошибок курсор устанавливается возле первого оператора с ошибкой. Строка, в которой есть ошибка, выделяется оранжевой полосой. В нижней части экрана появляется текстовое окно, содержащее сведения об этой ошибке: имя файла, в котором найдена

ошибка, номер строки с ошибкой и тип ошибки. Для быстрого перехода к определенной ошибке необходимо дважды щелкнуть мышью на строке с ее описанием.

- » Типичная ошибка *Syntax error, «;» expected* (*Синтаксическая ошибка, «;» ожидалось*) — после инструкции не поставлена точка с запятой (рис. 26.1).

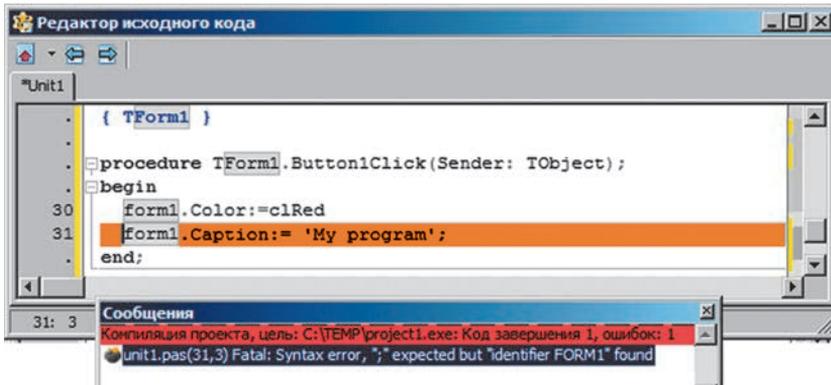


Рис. 26.1

Компилятор при обнаружении синтаксической ошибки останавливает процесс компиляции, выводит сообщение о найденной ошибке. Программисту нужно исправить эту ошибку и выполнить повторную компиляцию.

## Ошибки выполнения

Некоторые ошибки обнаруживаются при выполнении программы, когда программа пытается обработать недопустимые данные. Такие ошибки называют **ошибками периода выполнения**.

- » Рассмотрим фрагмент программы:

```
X := a/b;
Edit1.Text := FloatToStr(X);
```

Если программу с таким фрагментом кода запустить из среды программирования Lazarus, то она скомпилируется и будет работать правильно при любых значениях переменных  $a$  и  $b$ , кроме  $b=0$ . Но при  $b=0$  в момент выполнения оператора  $X := a/b$  остановит свою работу, выведя на экран соответствующее сообщение (см. рис. 26.2).

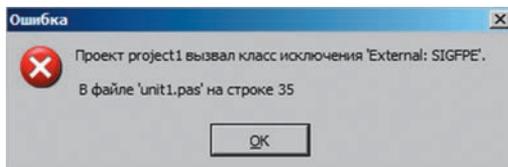


Рис. 26.2

Выполнение программы будет прервано на строке, содержащей ошибку (рис. 26.3). Если навести на переменную указатель мыши, то появится подсказка о текущем значении переменной.

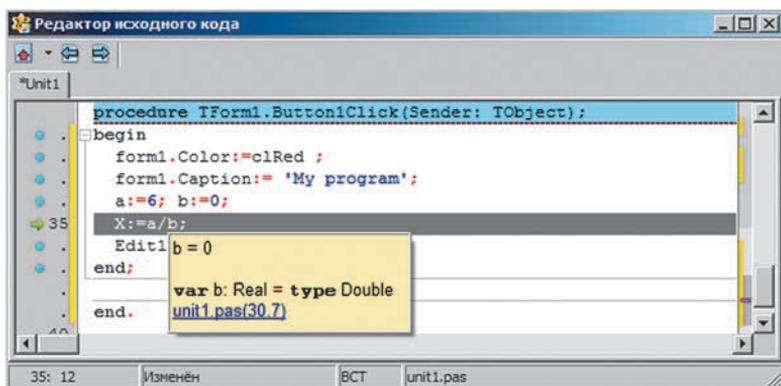


Рис. 26.3



Процесс устранения ошибок называют **отладкой программного кода**.

Целью отладки является выявление ошибок, которые возникают на этапе выполнения программы и приводят к неверному результату.

При отладке программы нужно внимательно анализировать сообщение об ошибке, которое выводит отладчик, чтобы определить, какая ошибка допущена и как ее исправить.

Для управления отладкой программы предназначены кнопки для пошагового выполнения программы Шаг со входом и Шаг в обход (рис. 26.4). Щелкая кнопки пошагового выполнения, можно по строкам выполнять программу, контролируя изменение значений переменных и правильность вычислений. Если команда вызова процедуры является текущей, то при

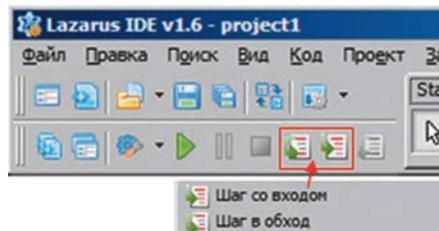


Рис. 26.4

щелчке кнопки Шаг со входом произойдет переход к пошаговому выполнению данной процедуры. По завершении пошагового выполнения процедуры произойдет переход к следующему оператору программы.

Кнопка Шаг в обход позволяет пропустить пошаговое выполнение процедуры и сразу перейти к следующему оператору программы.

Чтобы прервать режим пошагового выполнения программы, нужно нажать комбинацию клавиш Ctrl + F2.

## Логические ошибки

Самым сложным является выявление и исправление ошибок, вызванных неправильной логикой программы, то есть ошибок, допущенных при разработке алгоритма. Такие ошибки называют **логическими** (алгоритмическими). В некоторых случаях логическая ошибка может привести к ошибке выполнения (см. пример на с. 129–130).

Даже если программа успешно скомпилирована, результат ее выполнения может оказаться ошибочным.

» Если вы, желая увеличить число  $a$  на 1, вместо  $a := a + 1$  случайно запишете  $a := a + 2$ , то компилятор не выявит ошибки, но результат работы программы будет неверным.

Одним из способов выявления логической ошибки является выполнение программы в пошаговом режиме с анализом значений переменных на каждом шаге.

Для описания алгоритма программы и ее составляющих используют комментарии, которые можно включать в текст программы.



**Комментарий** — это текст, который игнорируется компилятором и включается в текст программы с целью облегчения понимания программного кода.

Чтобы компилятор отличил команды от комментариев, в языке Object Pascal комментарии следует заключать в фигурные скобки:

```
S := Edit3.Text; { ввод даты рождения }
```

Если текст комментария не занимает более одной строки, то можно перед комментарием вместо скобок поставить две косые черты:

```
S := Edit3.Text; // ввод даты рождения
```

Удачно сформулированные комментарии значительно упрощают понимание алгоритма программы.

## Вопросы для самопроверки



1. Какие ошибки называют синтаксическими?
2. Какие ошибки относятся к ошибкам периода выполнения?
3. Какие ошибки называют логическими?
4. Определите ошибку в приведенном фрагменте программного кода:  
 $x := 5; y := 1; z := x / (y - 1);$
5. В результате работы программы получен ответ: «1,5 землекопа». Определите, к какому виду относится ошибка.
6. Объясните, как выполняется программа в пошаговом режиме.

## Упражнение 26



- Исправить ошибки в приведенном программном коде, ознакомиться с режимом пошагового выполнения программы.
- 1) Создайте новый проект. Разместите на форме компонент Button. Создайте процедуру обработки события OnClick для командной кнопки (Button1).
  - 2) В код процедуры добавьте следующие команды:  
`Label1.Caption := 'Желаю удачи!';`  
`Label1.Color := clYellow;`  
`Label1.Font.Color := clBlue;`
  - 3) Запустите проект на выполнение. Строка `Label1.Caption := 'Желаю удачи!';` выделена оранжевой полосой, в окне ошибок содержится сообщение об ошибке:  
`[Error] Unit1.pas(28): Undeclared identifier: 'Label1'`  
 Ошибка возникла потому, что объекта Label1 на форме не существует.
  - 4) Добавьте на форму компонент Label.
  - 5) Запустите проект на выполнение в пошаговом режиме, нажав клавишу F8. Начинает выполняться код проекта:  
**begin**  
     Application.Initialize;  
     Application.CreateForm(TForm1, Form1);  
     Application.Run;  
**end.**

Продолжайте нажимать клавишу F8, пока не появится окно формы. Щелкните командную кнопку и продолжите пошаговое выполнение.

- 6) Прервите режим пошагового выполнения комбинацией клавиш Ctrl + F2. Сохраните проект в папке Упражнение 26.



### Компьютерное тестирование

Выполните тестовое задание 26 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## Практическая работа 11



### Создание объектно-ориентированной программы, отображающей окно сообщения

**Задание:** создать программу, в ходе выполнения которой отображается окно сообщения.

**Оборудование:** компьютер с установленной средой программирования Lazarus.

#### Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

#### I. Размещение элементов управления на форме

- ▶ 1. Откройте новый проект.
- ▶ 2. Добавьте на форму компонент Label, в окне Инспектор объектов измените значение его свойств:
  - Caption — «Следите за ходом выполнения программы»;
  - Color — clSkyBlue;
  - AutoSize — False (*запрет автоматического изменения размера надписи по горизонтали*);
  - WordWrap — True (*разрешение вывода текста в несколько строк*);
  - Font — шрифт: *Comic Sans MS*, размер: 14; цвет: белый.
- ▶ 3. Добавьте на форму компонент Button. Измените значение свойства Caption созданной кнопки (Button1) на «Выполнить программу».

## II. Написание процедур обработки событий

- ▶ 4. Создайте процедуру обработки события OnClick для командной кнопки. Запишите операторы, которые изменяют цвет фона формы на красный и выводят на экран окно сообщения о выполненном действии:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
 Form1.Color := clRed;
 ShowMessage('Изменился цвет формы');
end;
```

- ▶ 5. Измените вид окна сообщения (рис. 1). Задайте значение свойства Title для объекта Application:

```
Application.Title := 'My program';
ShowMessage('Изменился цвет формы');
```

- ▶ 6. Добавьте в программный код операторы:
 

```
Label1.Color := clGreen;
ShowMessage('Изменился цвет надписи на зеленый');
```

- ▶ 7. Разделите текст сообщения на две строки: вставьте служебные символы возврата каретки (#13), то есть установку курсора в начале строки, и переноса строки (#10).
 

```
ShowMessage('Изменился цвет '+#13+#10+' надписи на зеленый');
```

Запустите проект на выполнение. Проанализируйте изменения в виде окна сообщения (рис. 2).

- ▶ 8. Установите значение ширины формы 500 пикселей:

```
Form1.Width := 500;
```

- ▶ 9. Измените заголовок формы на «Практическая работа 11».
- ▶ 10. Добавьте оператор, меняющий цвет фона формы на желтый.
- ▶ 11. Запрограммируйте вывод окна сообщения после каждого выполненного действия.

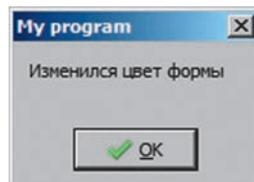


Рис. 1

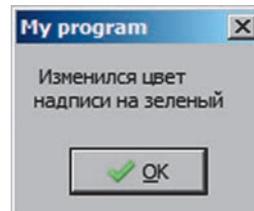


Рис. 2

## III. Тестирование проекта

- ▶ 12. После запуска проекта проверьте действие командной кнопки и убедитесь, что каждое выполненное действие сопровождается соответствующим сообщением. Сохраните проект в папке Практическая работа 11. Завершите работу за компьютером.

**Сделайте вывод:** как с помощью программного кода менять свойства объектов в ходе выполнения программы.



## Практическая работа 12

### Создание программы с кнопками и надписями

**Задание:** создать проект с разными кнопками и надписями.

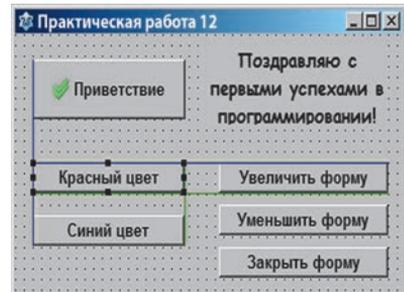
**Оборудование:** компьютер с установленной средой программирования Lazarus.

#### Ход работы

Во время работы за компьютером соблюдайте правила безопасности.

#### 1. Размещение элементов управления на форме

- ▶ 1. Откройте новый проект. Измените заголовок формы на «Практическая работа 12».
- ▶ 2. Добавьте на форму компонент StaticText (вкладка Additional на Палитре компонентов), предназначенный для вывода сообщения. В окне Инспектор объектов измените значения его свойств:
  - Caption — «Поздравляю с первыми успехами в программировании!»;
  - BorderStyle — sbsSunken (*стиль границы — утопленное окно*);
  - AutoSize — False;
  - Visible — False.
- ▶ 3. Добавьте на форму компонент BitBtn (вкладка Additional на Палитре компонентов). Измените вид кнопки, задав в окне Инспектор объектов свойству Kind значения bkAll. Измените параметры шрифта, задайте свойству Caption значения «Приветствие».
- ▶ 4. Добавьте на форму пять компонентов Button. Измените значение свойства Caption добавленных командных кнопок согласно рисунку.



## II. Написание процедур обработки событий

- ▶ **5.** Создайте процедуру обработки события `OnClick` для компонента `BitBtn` и в созданной процедуре `BitBtn1Click` запишите оператор, который сделает видимым объект `StaticText1`:  
`StaticText1.Visible := True;`
- ▶ **6.** Создайте процедуру обработки события `OnClick` для кнопки Красный цвет. Запрограммируйте изменение цвета фона формы на красный.  
`Form1.Color := clRed;`
- ▶ **7.** Создайте аналогичную процедуру обработки события для кнопки Синий цвет.
- ▶ **8.** Создайте процедуру обработки события `OnClick` для кнопки Увеличить форму. Запрограммируйте увеличение высоты и ширины формы на 10 пикселей:  
`Form1.Width := Form1.Width + 10;`  
`Form1.Height := Form1.Height + 10;`
- ▶ **9.** Создайте аналогичную процедуру обработки события для кнопки Уменьшить форму.
- ▶ **10.** Создайте для кнопки Закрыть форму процедуру обработки события `OnMouseMove`. В созданной процедуре `Button5MouseMove` запишите операторы:  
`Button5.Left := 100 + Random(250);`  
`Button5.Top := 30 + Random(250);`  
Значением функции `Random(250)` является случайное число в диапазоне от 0 до 249.
- ▶ **11.** Запустите проект на выполнение. Попробуйте щелкнуть кнопку Закрыть форму (кнопка должна «убегать» от указателя мыши).

## III. Тестирование проекта

- ▶ **12.** После запуска проекта проверьте работу всех командных кнопок. Сохраните проект в папке Практическая работа 12. Завершите работу за компьютером.

**Сделайте вывод:** как создавать кнопки и надписи.

# РАЗДЕЛ 7

## АЛГОРИТМЫ РАБОТЫ С ОБЪЕКТАМИ И ВЕЛИЧИНАМИ



§ 27. Величины и их свойства

§ 28. Элементы управления для ввода данных

Практическая работа 13. Составление и выполнение линейных алгоритмов обработки величин в учебной среде программирования

§ 29. Величины целых типов

§ 30. Величины вещественных типов

§ 31. Величины логического типа

§ 32. Величины символьного типа

§ 33. Величины строкового типа

§ 34. Преобразование величин одного типа в другой

Практическая работа 14. Отладка готовой программы

## ПОВТОРЯЕМ



Компьютер как исполнитель обрабатывает данные по определенной программе. Отдельный информационный объект, например число, символ, называют *величиной*. Из курса физики и математики вы уже знакомы с такими величинами, как скорость движения, расстояние и др., знаете, что они бывают постоянными и переменными. Величина, характеризующая состояние некоторого объекта, является базовым понятием математических и естественных наук. Все действия в программе выполняются с определенными величинами.

Любая величина, которую обрабатывает программа, занимает определенное место в памяти компьютера. *Значение величины* — это данные, хранящиеся в ячейках памяти компьютера.

1. При изучении каких учебных предметов вы встречались с понятием величины?
2. Приведите примеры величин.
3. Какими бывают величины?
4. Как компьютер обрабатывает величины?



В этом разделе вы узнаете об алгоритмах работы с объектами, типах величин и алгоритмах их обработки.

## § 27. Величины и их свойства



Отдельный информационный объект (число, символ, строка, таблица и пр.) называют **величиной**.

Основными характеристиками величины являются название, вид, тип и значение.

**Название (имя, идентификатор) величины** — это набор символов, который выбирают для обозначения констант, переменных, типов данных и т. п. В языке Object Pascal название может состоять из латинских букв, цифр, знака подчеркивания, а начинаться только с буквы или знака подчеркивания. При этом регистр символов значения не имеет.

**Вид величины** определяет способ использования величины в программе. Величина может быть константой (то есть постоянной) или переменной. *Константы* — это величины, значения которых не могут изменяться в ходе выполнения программы. *Переменные* — величины, значения которых могут изменяться в ходе выполнения программы.

**Тип величины** определяется объемом памяти, необходимым для ее хранения, множеством допустимых значений и операциями, которые можно над ней выполнять.

» Стандартными типами данных являются целый (Integer), вещественный (Real), строковый (String), логический (Boolean) и символьный (Char) типы.

*Порядковые* типы данных — такие типы, множество допустимых значений в которых упорядочены (пронумерованы), то есть для каждого из значений можно указать предыдущее и следующее значения. К порядковым относятся целые, символьные и логические типы данных.

### Объявление переменных

В языке Object Pascal каждая переменная перед использованием должна быть объявлена. При объявлении (описании) переменной указывается ее тип. Переменные объявляются в программе в блоке описания переменных, который начинается словом **var**:

**var** Название: Тип;

В модуле перед ключевым словом **implementation** расположен блок описания глобальных переменных:

**var** Form1: TForm1;

Переменные, объявленные в этом блоке, называют **глобальными**, поскольку они могут использоваться в любой процедуре обработки событий и сохраняют свои значения до тех пор, пока выполняется программный код формы.

Блок описания локальных переменных при необходимости можно создать в каждой процедуре. Этот блок расположен между заголовком данной процедуры и ключевым словом **begin**. Переменные, объявленные таким образом, являются **локальными**. Они становятся недоступными за пределами данной процедуры.

» Если нужно описать несколько переменных одного типа, то их записывают через запятую:

```
var A, B, C: Integer;
```

## Константы

В языке Object Pascal существует два вида констант: обычные и именованные.



**Обычная константа** — целое или дробное число, строка символов либо отдельный символ, логическое значение.

**Именованная константа** — идентификатор, который в тексте программы используется вместо значения постоянной величины.

Именованную константу, как и переменную, перед использованием нужно объявить. Блок объявления констант начинается словом **const**:  
**const** Имя = Значение;

» Объявление константы *g*:

```
const g = 9.81;
```

## Оператор присваивания

Переменная может принимать разные значения в процессе выполнения программы, например в результате выполнения оператора присваивания.

Синтаксис оператора присваивания:

Название := Выражение;

Сначала вычисляется значение выражения, которое потом присваивается переменной.

» **var** S: String; Age: Integer;

**begin**

```
 S := 'Мой текст';
```

```
Label1.Caption := S;
Age := 12 + 2;
end;
```



Найти ошибку в данном фрагменте программы:

```
var A, B, C: Integer;
begin
 A := 5;
 B := 10;
 C := A + B;
 Label1.Caption := C;
end;
```

Поскольку свойство `Caption` имеет строковый тип (`String`), а переменные `A`, `B`, `C` — числовой тип (`Integer`), то нужно представить числовое значение переменной `C` в виде текста. Для этого существует встроенная функция `IntToStr`. Строка в тексте программы, которая вызвала ошибку, должна выглядеть так:

```
Label1.Caption := IntToStr(C);
```

### Вопросы для самопроверки



1. Назовите и охарактеризуйте виды величин.
2. Назовите основные характеристики величины.
3. Определите, какие из приведенных последовательностей символов не могут быть идентификаторами и почему: `sum`; `W1`; `Primer 1`; `(SUM)`; `A-4`; `Primer_1`; `var`; `9A`.
4. Приведите примеры стандартных типов данных в Object Pascal.
5. Как в программе объявить, что переменная `A` — целого типа, `B` — логического типа, `C` — строкового типа?

### Упражнение 27



- ▶▶ Составить программу для сложения двух целых чисел.
- 1) Создайте новый проект. Разместите на форме кнопку `Button1`, измените заголовок созданной кнопки на «Вычислить».
  - 2) Разместите на форме компонент `Label`.
  - 3) Создайте процедуру обработки события `OnClick` для кнопки `Вычислить`. В окне программного кода между заголовком процедуры и ключевым словом `begin` опишите переменные `A`, `B`, `C` целого типа:

```
procedure TForm1.Button1Click (Sender: TObject);
var A, B, C: Integer;
begin
```

- 4) В программном коде процедуры Button1Click запишите операторы для нахождения суммы двух чисел и вывода результата в надпись Label1:

```
begin
 A := 5;
 B := 10;
 C := A + B;
 Label1.Caption := IntToStr(C);
end;
```

Запустите проект на выполнение, проверьте действие кнопки. Сохраните проект в папке Упражнение 27.



- 5) Добавьте на форму надпись Label2. Дополните программный код операторами вычисления произведения чисел A и B и вывода результата в надпись Label2. Проверьте действие кнопки.

- 6) Добавьте на форму кнопку Button2. Измените заголовок созданной кнопки на «Вычислить 2». Создайте процедуру обработки события OnClick для этой кнопки, в которой опишите переменные x и y целого типа. Добавьте в программный код процедуры следующие операторы:

```
x := 5; y := 6; x := y + x;
```

Выведите новые значения переменных x и y в надписях Label1 и Label2. Проверьте действие кнопки.

### Компьютерное тестирование



Выполните тестовое задание 27 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 28. Элементы управления для ввода данных

Большинство задач по программированию предусматривают ввод данных пользователем, после чего над ними выполняются некоторые операции, а результат обработки выводится в заданном виде:

ввод → обработка → вывод.

## Ввод и вывод текстовых данных

Для ввода данных пользователем и вывода результатов работы программы используют компонент Edit (рис. 28.1), который представляет собой однострочное текстовое поле. Основное свойство компонента Edit — это свойство Text строкового типа (String). Начальное значение свойства Text можно задавать в окне Инспектор объектов (рис. 28.2). Это значение можно изменить при выполнении программы, предусмотрев в коде программы соответствующий оператор присваивания.

» Содержимое текстового поля можно присваивать переменной строкового типа:

```
var S: String;
begin
 Edit1.Text := 'Текст, который выводится';
 S := Edit1.Text;
end;
```

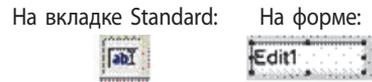


Рис. 28.1

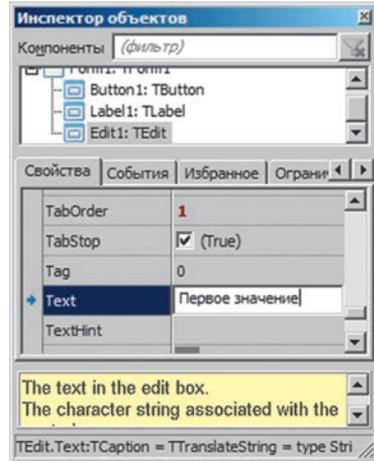


Рис. 28.2

## Преобразование типов

Иногда при разработке программы возникает необходимость преобразования одного типа данных в другой. Например, для вывода числа в текстовое поле нужно представить это число как строку символов.

Для решения этой проблемы в языке Object Pascal существуют встроенные функции преобразования типов:

| Функция    | Назначение                                |
|------------|-------------------------------------------|
| IntToStr   | Преобразует целое число в строку          |
| StrToInt   | Преобразует строку в целое число          |
| FloatToStr | Преобразует действительное число в строку |
| StrToFloat | Преобразует строку в действительное число |

- Найти сумму двух целых чисел, которые вводятся в текстовые поля Edit1 и Edit2. Значение суммы вывести в текстовое поле Edit3.

```
var A, B, C: Integer;
begin
 A := StrToInt(Edit1.Text);
 B := StrToInt(Edit2.Text);
 C := A + B;
 Edit3.Text := IntToStr(C);
end;
```

## Компонент ComboBox

Компонент ComboBox  (*Комбинированный список*) находится на вкладке Standard Палитры компонентов. Он представляет собой объединение списка строк с однострочным текстовым полем Edit. Каждая строка списка ComboBox является строкой данных типа String.

При этом список строк элемента управления ComboBox сначала скрыт и раскрывается при щелчке мышью кнопки раскрытия списка, которая находится справа в строке ввода (рис. 28.3).

Для добавления строк на этапе разработки интерфейса нужно:

- 1) на вкладке Свойства в окне Инспектор объектов в строке Items щелкнуть кнопку ;
- 2) ввести нужные данные в окне Диалог ввода строк (рис. 28.4);
- 3) щелкнуть кнопку ОК.

- Вывести в текстовое поле номер выбранной строки списка ComboBox (рис. 28.5):

```
var S: String; N: Integer;
begin
 N := ComboBox1.ItemIndex;
 Edit1.Text := IntToStr(N);
 { если выбрана строка «Венера», N = 1 }
end;
```



Рис. 28.3

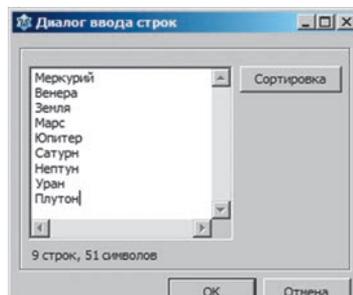


Рис. 28.4

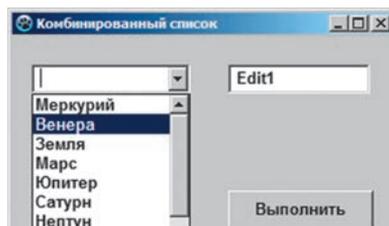


Рис. 28.5

» Вывести в текстовое поле содержимое выбранной строки списка ComboBox:

```
var S: String; N: Integer;
begin
 N := ComboBox1.ItemIndex;
 S := ComboBox1.Items[N];
 Edit1.Text := S;
end;
```

Чтобы при выполнении программы можно было вводить значение в текстовое поле ComboBox (свойство Text) и добавлять введенное значение в список ComboBox, процедура обработчика событий должна содержать вызов метода Add с помощью команды:

```
ComboBox1.Items.Add(ComboBox1.Text);
```

## Функция InputBox

Для ввода данных можно использовать функцию InputBox. Эта функция выводит диалоговое окно, в котором пользователь должен ввести данные и щелкнуть кнопку ОК (подтверждение действия) или Cancel (отмена действия).

Синтаксис функции:

```
Value := InputBox (Заголовок, Подсказка, Значение);
```

где Value — переменная строкового типа, принимающая значение функции;

Заголовок — заголовок окна функции (необязательный параметр);

Подсказка — сообщение, которое будет выведено в диалоговом окне функции;

Значение — строка текста, которая будет отображена в поле ввода и выделена синим цветом. В случае отказа пользователя от ввода текста нажатием клавиши Esc или кнопки Cancel функция InputBox вернет это значение.

» Задать значение переменной с помощью функции InputBox (рис. 28.6).

```
var myname: String;
begin
 myname := InputBox('Name',
 'Ваше имя?', 'Не задано');
end;
```

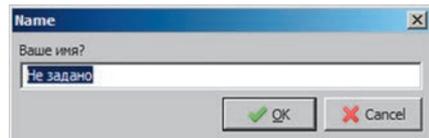


Рис. 28.6

С помощью функции `InputDialog` можно организовать диалог пользователя с программой, не перегружая форму лишними элементами управления для ввода данных.

## Вопросы для самопроверки



1. Назовите основные свойства компонента `Edit`.
2. Запишите оператор присваивания переменной целого типа значения, содержащегося в текстовом поле `Edit1`.
3. Запишите оператор вывода значения переменной целого типа в текстовое поле `Edit2`.
4. Значением какого свойства являются элементы списка `ComboBox`?
5. Какое свойство сохраняет номер выбранного элемента в списке `ComboBox`?
6. Объясните назначение функций и их параметров в операторе:  
`age := StrToInt(InputBox('Возраст', 'Сколько тебе лет?', '0'));`

## Упражнение 28



► Составить программу, реализующую диалог пользователя с программой по заданному плану:

- 1) по щелчку кнопки выводится запрос об имени пользователя;
- 2) после ответа открывается второе окно с запросом о возрасте (рис. 1);
- 3) после ответа в окне выводится сообщение с приветствием и указанием класса, в котором учится пользователь (рис. 2).

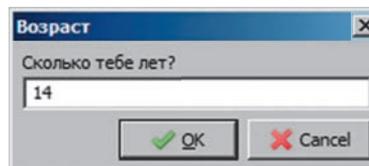


Рис. 1

1) Создайте новый проект. Разместите на форме кнопку `Button1` и измените заголовок созданной кнопки на «Выполнить».

2) Создайте процедуру обработки события `OnClick` для кнопки `Выполнить`.

3) Опишите переменные:

`var myname: String; age, clas: Integer;`

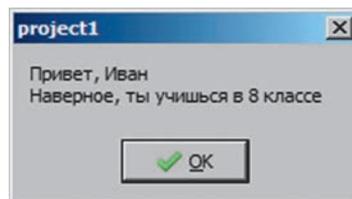


Рис. 2

- 4) В программном коде процедуры Button1Click запишите операторы:

**begin**

```
myname := InputBox('Name', 'Твое имя?', 'Не задано');
age := StrToInt(InputBox('Возраст', 'Сколько тебе лет?', '0'));
clas := age - 6;
ShowMessage('Привет, '+myname+#10#13+ 'Наверное, ты учишься
в ' + IntToStr(clas)+ ' классе ');
```

**end;**

- 5) Запустите проект на выполнение. Сохраните проект в папке Упражнение 28.



- 6) Дополните проект выводом еще одного сообщения: «Угадал класс обучения?», ответом на которое будет «Да» или «Нет». Проверьте работу программы.



### Компьютерное тестирование

Выполните тестовое задание 28 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## Практическая работа 13

### Составление и выполнение линейных алгоритмов обработки величин в учебной среде программирования



**Задание:** создать проект для решения задачи.

*Два поезда выехали одновременно навстречу друг другу. Найти время, через которое они встретятся, если известны расстояние между поездами в момент начала движения и скорости движения каждого поезда.*

**Оборудование:** компьютер с установленной средой программирования Lazarus.

#### Ход работы

При работе за компьютером соблюдайте правила безопасности.

#### I. Размещение элементов управления на форме

- 1. Создайте новый проект. Разместите на форме четыре компонента Edit, компонент Button и четыре компонента Label.

Измените значение свойства Caption согласно рисунку, задайте значение свойства Font по собственному усмотрению.

## II. Создание процедур обработки событий

- ▶ 2. Создайте процедуру обработки события OnClick для кнопки Вычислить. Выполнение процедуры должно привести к выводу искомого времени в текстовое поле Edit4, если значения скоростей движения поездов и расстояния между ними введены в соответствующие поля Edit1, Edit2, Edit3.

Ориентировочный программный код:

```
var v1, v2, s, t: Real;
begin
 v1 := StrToFloat(Edit1.Text);
 v2 := StrToFloat(Edit2.Text);
 s := StrToFloat(Edit3.Text);
 t := s/(v1 + v2);
 Edit4.Text := FloatToStr(t);
```

end;

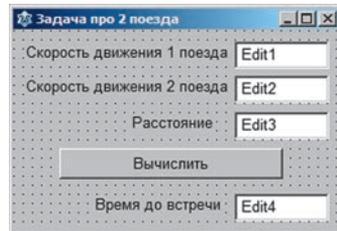
- ▶ 3. Получите результат работы программы для следующих исходных данных: расстояние — 1000 км, скорость движения — 120 и 130 км/ч соответственно.
- ▶ 4. Внесите изменения в интерфейс программы и программный код, чтобы проект реализовывал решение другой задачи.

*Два поезда выехали одновременно с одной станции в противоположных направлениях. Нужно найти расстояние, на котором они окажутся друг от друга через заданное время, если известны скорости движения каждого поезда.*

## III. Тестирование проекта

- ▶ 5. Создайте три набора входных данных для тестирования проекта. Вычислите ожидаемые результаты по составленным тестам.
- ▶ 6. Проверьте работу программы на созданных тестовых данных. Сохраните проект в папке Практическая работа 13. Завершите работу за компьютером.

**Сделайте вывод:** как составлять и выполнять линейные алгоритмы для решения задач.



## § 29. Величины целых типов

### Целые типы данных

Данные целых типов — это целые числа, которые могут использоваться в арифметических выражениях и занимают в памяти компьютера от 1 до 8 байтов. Характеристики некоторых целых типов:

| Тип данных | Объем памяти (байты) | Диапазон значений       |
|------------|----------------------|-------------------------|
| LongInt    | 4                    | -2147483648..2147483647 |
| Integer    | 2                    | -32768..32767           |
| Byte       | 1                    | 0..255                  |
| ShortInt   | 1                    | -128..127               |

При записи в тексте программы арифметических выражений следует соблюдать определенные правила. Для данных целых типов определены следующие арифметические операции: +, −, \*, div, mod. Результат выполнения этих операций также относится к целому типу. С целыми числами можно выполнять операцию деления (/), но результатом ее выполнения будет действительное число.

Рассмотрим подробнее операции div и mod.

| Название операции                         | Обозначение | Пример          | Результат |
|-------------------------------------------|-------------|-----------------|-----------|
| Целочисленное деление                     | div         | Res := 11 div 3 | Res = 3   |
| Остаток от деления<br>(деление по модулю) | mod         | Res := 11 mod 3 | Res = 2   |

### Стандартные функции

В языке Object Pascal существует ряд встроенных (стандартных) функций. Их используют в выражениях.

Рассмотрим некоторые стандартные функции, которые можно применять к данным целого типа.

| Функция | Назначение                        | Тип результата | Пример                          |
|---------|-----------------------------------|----------------|---------------------------------|
| Abs(x)  | Вычисляет $ x $                   | Integer        | Abs(-5) = 5                     |
| Sqr(x)  | Вычисляет $x^2$                   | Integer        | Sqr(5) = 25                     |
| Sqrt(x) | Вычисляет $\sqrt{x}$              | Real           | Sqrt(64) = 8                    |
| Odd(x)  | Проверяет аргумент на нечетность  | Boolean        | Odd(5) = True<br>Odd(4) = False |
| Succ(x) | Возвращает следующее целое число  | Integer        | Succ(5) = 6                     |
| Pred(x) | Возвращает предыдущее целое число | Integer        | Pred(5) = 4                     |

» Вычислить сумму цифр введенного трехзначного числа A (рис. 29.1).

a1, a2, a3 — переменные для сохранения соответствующих цифр числа; S — сумма цифр.

**var** A, a1, a2, a3, S: Integer;

**begin**

A := StrToInt(Edit1.Text);

a1 := A div 100;

a2 := A div 10 mod 10;

a3 := A mod 10;

S := a1 + a2 + a3;

Edit4.Text := IntToStr(S);

**end;**

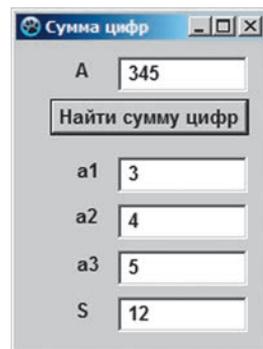


Рис. 29.1

Результат выполнения операций над данными целых типов (например, Integer и Byte) относится к типу того операнда, который обладает большим диапазоном значений.

Выход за границы диапазона (переполнение памяти, выделенной под переменную) может привести к неверным результатам вычисления.

» Вычислить выражение  $10x$  (рис. 29.2):

**var** x, a: Byte; y: Integer;

**begin**

x := 255;

a := 10 \* x; { переполнение памяти, так как  $0 \leq a \leq 255$  }

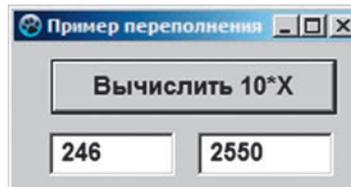


Рис. 29.2

```

y := 10 * x; // нет переполнения памяти, так как $-2^{32} \leq y \leq 2^{32} - 1$
Edit1.Text := IntToStr(a);
Edit2.Text := IntToStr(y);
end;

```

## Генератор случайных чисел

Случайные числа часто применяют в программировании для моделирования результатов какого-либо эксперимента при разных условиях, создании игровых или тестовых программ и т. п. Для получения при каждом запуске программы разных случайных чисел необходимо выполнить процедуру *Randomize* (*запуск генератора случайных чисел*). Случайные целые числа генерируются с помощью функции *Random*:

*Random*(диапазон)

Здесь диапазон — параметр, указывающий верхнюю границу диапазона, из которого выбирается случайное число.

» Получить случайное число  $x$  в диапазоне от 0 до 9.

```

var x: Integer;
begin
 Randomize;
 x := Random(10);
 Edit1.Text := IntToStr(x);
end;

```

## Вопросы для самопроверки



1. Для каких значений аргумента  $x$  функции *Abs*( $x$ ) и *Sqr*( $x$ ) возвращают результат целого типа?
2. Вычислите значения выражений:  
а) *Odd*(6); б) *Odd*(3); в) *Succ*(10); г) *Pred*(7); д) *Sqr*(2); е) *Sqrt*(16).
3. Вычислите значения выражений:  
а)  $123 \text{ div } 100$ ; б)  $123 \text{ mod } 10$ ; в)  $123 \text{ div } 10 \text{ mod } 10$ .
4. Запишите операторы присваивания, реализующие следующие действия: а) переменной  $S$  присвоить значение суммы переменных  $A$  и  $B$ ; б) удвоить значение переменной  $A$ ; в) переменную  $A$  увеличить на 10.
5. Запишите выражение, результатом которого является целое случайное число в диапазоне от 40 до 100.

## Упражнение 29



- ▶ Составить программу «Прыгающая кнопка».
- 1) Создайте новый проект. Разместите на форме компоненты Label и Button. Измените заголовок кнопки Button1 на «Поймай меня!».
  - 2) Создайте процедуру обработки события OnClick для кнопки Поймай меня!, которая будет считать количество щелчков кнопки и выводить результат с помощью компонента Label. Опишите переменную K для хранения количества щелчков кнопки:
 

```
var K: Integer;
```

 В программный код процедуры запишите операторы:
 

```
K := K + 1;
Label1.Caption := IntToStr (K);
```
  - 3) Добавьте на форму компонент Timer  (вкладка System на Палитре компонентов). Свойство таймера Interval оставьте без изменений.
  - 4) Создайте процедуру обработки события OnTimer для объекта Timer. В программном коде процедуры запишите операторы, определяющие случайное движение кнопки (свойства Left и Top). Кнопка не должна «выпрыгнуть» за границы формы, поэтому нужно настроить свойства Left и Top согласно размерам формы (ClientWidth и ClientHeight) и кнопки (Width и Height).
 

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
 Randomize;
 Button1.Left := Random(ClientWidth - Button1.Width);
 Button1.Top := Random(ClientHeight - Button1.Height)
end;
```
  - 5) Запустите проект на выполнение. Удастся ли «поймать» кнопку? Если нет, остановите выполнение и увеличьте значение свойства таймера Interval. Сохраните проект в папке Упражнение 29.
  - 6) Добавьте на форму кнопку Button2, в программном коде которой запрограммируйте увеличение значения Interval:
 

```
Timer1.Interval := Timer1.Interval + 100;
```

 Добавьте на форму кнопку Button3, предназначенную для уменьшения значения Interval. Проверьте действие кнопок.



## Компьютерное тестирование



Выполните тестовое задание 29 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 30. Величины вещественных типов

### Действительные типы данных

Данными вещественных типов могут быть любые действительные (вещественные) числа из определенного диапазона. Они занимают в памяти компьютера от 4 до 10 байтов. Характеристики некоторых вещественных типов:

| Тип данных | Объем памяти (байты) | Количество точных цифр | Диапазон значений                                |
|------------|----------------------|------------------------|--------------------------------------------------|
| Extended   | 10                   | 19–20                  | $3,4 \cdot 10^{-4932} \dots 1,1 \cdot 10^{4932}$ |
| Double     | 8                    | 15–16                  | $5 \cdot 10^{-324} \dots 1,7 \cdot 10^{308}$     |
| Real       | 6                    | 11–12                  | $2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{38}$     |

В таблице указаны диапазоны положительных значений, однако допустимыми являются отрицательные значения с такими же диапазонами модулей, а также число 0.

Для данных вещественных типов определены арифметические операции: +, -, \*, /. Результат выполнения таких операций также имеет действительный тип.

Для обработки данных вещественного типа в языке Object Pascal применяют следующие стандартные функции:

| Функция  | Назначение                       | Тип результата | Пример                              |
|----------|----------------------------------|----------------|-------------------------------------|
| Frac(x)  | Возвращает дробную часть числа x | Real           | Frac(1.234) = 0.234                 |
| Int(x)   | Возвращает целую часть числа x   | Real           | Int(1.234) = 1                      |
| Round(x) | Округляет число                  | Int64          | Round(1.234) = 1<br>Round(1.56) = 2 |
| Trunc(x) | Возвращает целую часть числа x   | Int64          | Trunc(1.234) = 1<br>Trunc(1.56) = 1 |
| Abs(x)   | Вычисляет $ x $                  | Real           | Abs(-5.3) = 5.3                     |
| Cos(x)   | Вычисляет $\cos x$               | Real           | Cos(Pi) = -1                        |
| Sin(x)   | Вычисляет $\sin x$               | Real           | Sin(Pi/2) = 1                       |
| Sqr(x)   | Вычисляет $x^2$                  | Real           | Sqr(5) = 25                         |
| Sqrt(x)  | Вычисляет $\sqrt{x}$             | Real           | Sqrt(255) =<br>= 15.9687194226713   |

Аргумент функций  $\text{Sin}(x)$  и  $\text{Cos}(x)$  задают в радианах. При необходимости перевода значения градусов в радианы следует воспользоваться формулой:

$$\text{радианы} = \text{градусы} \cdot \pi / 180.$$

Если дробная часть аргумента функции  $\text{Round}(x)$  равна 0,5, то округление выполняется до ближайшего четного числа.

» Значения функций  $\text{Round}(x)$  и  $\text{Trunc}(x)$  целые, хотя аргумент является действительным.

```
var x, y: Integer; a: Real;
begin
 a := 4.65;
 x := Round(a); Edit1.Text := IntToStr(x); // x=5
 y := Trunc(a); Edit2.Text := IntToStr(y); // y=4
end;
```

Значение выражения, в котором используются переменные целого и вещественного типов, имеет вещественный тип. При попытке присваивания переменной вещественного типа значения целого типа получаем действительное значение.

При попытке переменной целого типа присвоить значение вещественного типа получаем сообщение об ошибке совместимости типов.

» Присвоить переменной  $x$  целого типа значение  $\text{Sqrt}(25)$  (рис. 30.1).

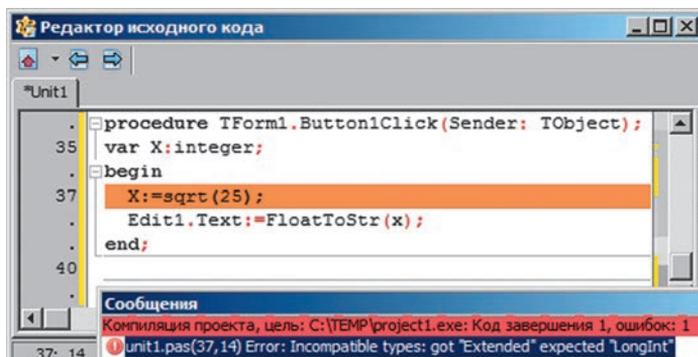


Рис. 30.1

## Правила записи арифметических выражений

Арифметические выражения составляются из переменных, констант, функций, знаков операций и круглых скобок по математическим правилам.

При записи арифметических выражений в языке Object Pascal следует соблюдать следующие правила:

| Правила записи арифметических выражений                                                                                      | Математическая запись | Запись на языке Object Pascal |
|------------------------------------------------------------------------------------------------------------------------------|-----------------------|-------------------------------|
| 1. Выражение записывают в виде строки символов. Порядок выполнения операций одного приоритета регулируется с помощью скобок. | $\frac{2x-5}{3+x}$    | $(2*x-5)/(3+x)$               |
|                                                                                                                              | $2ab$                 | $2*a*b$                       |
| 2. Нельзя опускать знак операции умножения.                                                                                  | $v_0^2$               | $Sqr(v0)$                     |
| 3. Аргументы функции записывают в круглых скобках                                                                            | $b+\sqrt{a}$          | $b+Sqrt(a)$                   |

Вычисление выражений выполняется по определенным правилам. В первую очередь вычисляется значение функции. Если аргумент функции является выражением, то сначала вычисляется значение аргумента. Далее выполняются арифметические операции слева направо с учетом приоритета и расположения скобок. Сначала выполняются операции высшего приоритета: \*, /, div, mod, а потом — низшего приоритета: +, -.

» Вычислить значение выражения  $\text{Sin}(2*a)$ .  
В выражении  $\text{Sin}(2*a)$  сначала вычисляется значение аргумента  $2*a$ , а потом — значение функции  $\text{Sin}$ .

» Вычислить значения выражений  $24/2*3$  и  $24/(2*3)$ .  
В результате выполнения операторов присваивания  
 $A := 24/2*3$ ;  $B := 24/(2*3)$ ;  
переменная A примет значение 36, переменная B — значение 4.

### Вопросы для самопроверки



- Определите тип (целый или вещественный) выражений:  
а)  $20/4$ ; б)  $\text{Sqr}(4)$ ; в)  $\text{Sqrt}(16)$ ; г)  $\text{Trunc}(-3.14)$ .
- Какие функции имеют разный тип аргумента и результата?
- Вычислите значения выражений:  
а)  $\text{Trunc}(6.9)$ ;  $\text{Round}(6.9)$ ;  $\text{Int}(6.9)$ ; б)  $\text{Trunc}(-1.8)$ ;  $\text{Round}(-1.8)$ ;  $\text{Int}(-1.8)$ .
- Объясните правила записи арифметических выражений.
- Запишите данные формулы по правилам языка Object Pascal:  
а)  $\frac{x-5}{2x}$ ; б)  $x^5$ ; в)  $2a+10^5d-4\frac{3}{7}c^5+\frac{2}{3}$ ; г)  $\sqrt{|x-1|}$ .
- Вычислите значение выражения  $\text{Succ}(\text{Round}(5/2))-\text{Pred}(3)$ .

## Упражнение 30

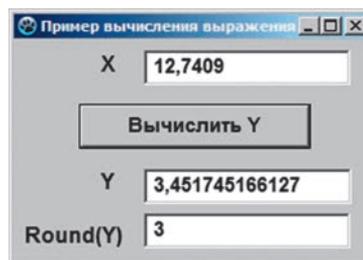


▶ Дана функция  $y = \sqrt{|x-1| + \sin x}$ . Составить программу для нахождения значения функции при заданном значении  $x$ .

- 1) Создайте новый проект. Измените заголовок формы на «Вычисление значения функции». Разместите на форме кнопку Button1 и измените заголовок созданной кнопки на «Вычислить Y».
- 2) Разместите на форме три компонента Label, измените их заголовки согласно рисунку.
- 3) Разместите на форме три текстовых поля Edit согласно рисунку.
- 4) Создайте процедуру обработки события OnClick для кнопки Вычислить Y.

В программном коде созданной процедуры запишите операторы:

```
var A: Integer; X, Y: Real;
begin
 X := StrToFloat(Edit1.Text);
 Y := Sqrt (Abs(X-1)+Sin(X));
 Edit2.Text := FloatToStr(Y);
 A := Round(Y);
 Edit3.Text := IntToStr(A);
end;
```



- 5) Запустите проект на выполнение. На рисунке приведен результат работы программы при  $x=12,7409$ .

! В программном коде целую и дробную части действительного числа отделяют точкой «.». Разделитель при вводе дробного числа в поле Edit1 зависит от настроек системы (в русском варианте — запятая «,»).

- 6) Проверьте работу программы для разных значений  $x$ . Сохраните проект в папке Упражнение 30.



- 7) Измените программный код для нахождения значения функции  $y = 1 + \frac{1}{x} + \frac{1}{x^2}$ . Вычислите значение данной функции для  $x = -0,387$ .



## Компьютерное тестирование



Выполните тестовое задание 30 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 31. Величины логического типа

### Простые условия

Простое логическое выражение (простое условие) образуется с помощью операции отношения:

| Знак операции | Содержание       | Пример | Результат |
|---------------|------------------|--------|-----------|
| =             | Равно            | 8 = 9  | False     |
| >             | Больше           | 8 > 9  | False     |
| <             | Меньше           | 8 < 9  | True      |
| >=            | Больше или равно | 5 >= 5 | True      |
| <=            | Меньше или равно | 5 <= 2 | False     |
| <>            | Не равно         | 2 <> 5 | True      |

Логические выражения принимают значение True или False. В языке Object Pascal для величины, значением которой является True или False, используют логический тип (Boolean).

» Вычислить значения логических выражений:

```
var X, Y: Integer; A: Boolean;
```

```
begin
```

```
 X := 5; Y := 2;
```

```
 A := X > Y; // A = True, так как выполняется условие X > Y
```

```
 A := X < Y; // A = False
```

```
 A := X - 4.5 < Y * 2; // A = True
```

```
end;
```

### Составные условия

**Составное выражение (составное условие)** — несколько простых выражений, соединенных логическими операциями and (*и*), or (*или*), xor (*исключающее или*), not (*логическое отрицание*).

Таблицы истинности

| A     | B     | A and B | A or B | A xor B |
|-------|-------|---------|--------|---------|
| False | False | False   | False  | False   |
| False | True  | False   | True   | True    |
| True  | False | False   | True   | True    |
| True  | True  | True    | True   | False   |

| A     | not A |
|-------|-------|
| False | True  |
| True  | False |

Логические операции, операции отношения и арифметические операции могут встречаться в одном выражении. Вычисление значений таких выражений выполняется слева направо с учетом приоритета операций:

- 1) not;
- 2) and, \*, /, div, mod;
- 3) or, xor, +, -;
- 4) операции сравнения.

» Рассмотрим примеры составных логических выражений:

- а) not ( $A \leq 3$ ) — то же, что  $A > 3$ .
- б)  $(A > 10)$  and  $(A < 18)$  — принимает значение True, если значение  $A$  принадлежит промежутку от 10 до 18.
- в)  $(A \leq 10)$  or  $(A \geq 18)$  — принимает значение True, если значение  $A$  не принадлежит промежутку от 10 до 18.

» Определить, принадлежит ли точка с координатой  $x$  отрезку  $[-5; 5]$  (рис. 31.1). Точка принадлежит отрезку, если выполняется неравенство  $-5 \leq x \leq 5$ . В программировании такое двойное неравенство записывают как составное условие:  
( $x \geq -5$ ) and ( $x \leq 5$ ).



Рис. 31.1

» Вычислить значения логических выражений:

```
var X, Y, Z: Integer; A: Boolean;
```

```
begin
```

```
 X := 1; Y := 2; Z := 3;
```

```
 A := (X < Y) and (Y < Z); // A = True
```

```
 A := (X > Y) or (Y > Z); // A = False
```

```
 A := (X > Y) xor (Y < Z); // A = True
```

```
end;
```

## Преобразование типов

При выводе логических значений нужно преобразовывать логическое значение в строку символов, и наоборот, при вводе преобразовывать строку символов в соответствующее логическое значение. Для этого в языке Object Pascal существуют встроенные функции преобразования типов:

- StrToBool(S) — преобразование строки символов  $S$  в логическое значение;
- BoolToStr(b,True) — преобразование логического значения в строку символов.

Если второй аргумент функции BoolToStr равен True, то получаем значение функции в виде строки 'True' или 'False'. Если второй аргумент заменить на False, то значением функции для истинного  $b$  будет строка '-1', а для ложного — '0'.

### Вопросы для самопроверки



1. Какие значения может принимать переменная логического типа?
2. Какие операции можно выполнять над переменными логического типа?
3. Переменные логического типа A и B принимают значения в результате выполнения операторов:  $A := X > 3$ ;  $B := X < 0$ ; При каких значениях X выражение  $A \text{ or } B$  принимает значение True? При каких значениях X выражение  $A \text{ and } B$  принимает значение False?
4. Запишите в виде логических выражений данные условия:  
а)  $3 \leq x \leq 10$ ; б)  $-5 \leq x \leq 5$  при условии, что  $x \neq 0$ .
5. Вычислите значения логических выражений:  
а)  $(A \leq B) \text{ and } (A = B - 1)$  при  $A = 2, B = 4$ ;  
б)  $(A \leq B) \text{ or } (A = B - 1)$  при  $A = 2, B = 4$ .

### Упражнение 31

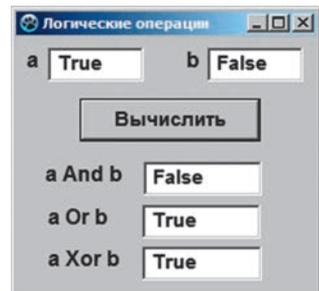


►► По введенным значениям переменных a, b: Boolean вычислить значения логических выражений a and b, a or b, a xor b.

- 1) Создайте новый проект. Измените заголовок формы на «Логические операции». Разместите на форме кнопку Button1 и измените заголовок созданной кнопки на «Вычислить».

- 2) Разместите на форме по пять компонентов Label и Edit согласно рисунку.
- 3) Создайте процедуру обработки события OnClick для кнопки Вычислить. В программном коде созданной процедуры запишите операторы:

```
var a, b, v1, v2, v3: Boolean;
begin
 a := StrToBool(Edit1.Text);
 b := StrToBool(Edit2.Text);
 v1 := a and b; v2 := a or b; v3 := a xor b;
 Edit3.Text := BoolToStr(v1, True);
```



```
Edit4.Text := BoolToStr(v2,True);
```

```
Edit5.Text := BoolToStr(v3,True);
```

```
end;
```

- 4) Запустите проект на выполнение. Проверьте результат работы программы при  $a = \text{True}$ ,  $b = \text{False}$ .
- 5) Проверьте работу программы для разных значений  $a$  и  $b$ . Сохраните проект в папке Упражнение 31.



### Компьютерное тестирование

Выполните тестовое задание 31 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua)

## § 32. Величины символьного типа

Для величины, значением которой является отдельный символ, используют символьный тип (Char). Как вам уже известно, каждый символ имеет свой номер, который называют его кодом.

Таким образом, значением переменной символьного типа может быть любой символ из таблицы кодов: буква; цифра; знак препинания, а также служебные символы, не отображаемые на экране.

В языке Object Pascal определены два символьных типа, один из которых обеспечивает кодирование символов в системе ASCII, а второй — в системе Юникод.

Значения величин символьного типа (Char) в тексте программы заключают в одинарные кавычки, например 'A', '3', '!', ' ' (пробел), или используют его код, добавляя префикс #. В частности, так удобно обозначать служебные символы. Например, символ возврата каретки — #13, символ абзаца — #10.

» Присвоить значения переменным символьного типа:

```
var Letter, Symbol, Ch: Char;
```

```
begin
```

```
 Letter := 'a';
```

```
 Symbol := '7';
```

```
 Ch := #65;
```

```
end;
```

Величины символьного типа можно сравнивать. Сравнение символов сводится к сравнению их кодов, поскольку каждому символу соответствует определенное число.

» Сравнить величины символического типа:

```
var b: Boolean;
begin
 b := 'A' < 'B'; // True
end;
```

» Вывести на форму таблицу кодов для прописных букв латиницы. По щелчку командной кнопки в поле Label1 будет выводиться часть таблицы кодов (от 65 до 90), соответствующих прописным буквам латиницы (рис. 32.1).

```
var c: Char; n: Byte; st: String;
begin
 st := '';
 For n := 65 to 90 do
 st := st + IntToStr(n) + ' ' + Chr(n) + #13;
 Label1.Caption := st;
end;
```

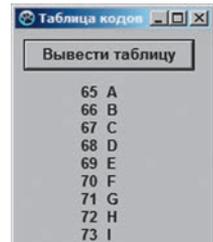


Рис. 32.1

К величинам символического типа в языке Object Pascal применяют следующие встроенные функции:

| Функция           | Назначение                             | Тип результата | Пример                               |
|-------------------|----------------------------------------|----------------|--------------------------------------|
| Chr(x)            | Возвращает символ по его коду          | Char           | S := Chr (65);<br>S = 'A'            |
| Ord(c)            | Возвращает числовой код символа с      | Byte           | K := Ord ('Z');<br>K = 90            |
| Pred(c)           | Возвращает символ, предыдущий с        | Char           | S := Pred ('b');<br>S = 'a'          |
| Succ(c)           | Возвращает символ, следующий после с   | Char           | S := Succ ('b');<br>S = 'c'          |
| AnsiUpperCase(c)* | Преобразует строчную букву в прописную | Char           | S := AnsiUpperCase ('b');<br>S = 'B' |

### Вопросы для самопроверки



1. Какие значения могут принимать переменные символического типа?
2. Какие функции можно использовать с символическими данными?
3. Какая функция возвращает символ по его коду?

\* Прописные буквы и символы, не являющиеся буквами, функция AnsiUpperCase не изменяет.

4. Какая функция возвращает числовой код символа?
5. Как сравниваются значения символьного типа?
6. Найдите значения выражений:
  - а)  $\text{Pred}('n')$ ; б)  $\text{Ord}('z') - \text{Ord}('y')$ ; в)  $\text{Ord}(\text{Chr}(13))$ .

### Упражнение 32



► Для символа, который вводится в текстовое поле Edit1, определить его код, предыдущий и следующий символы. Для строчных букв латиницы вывести соответствующие прописные буквы.

- 1) Создайте новый проект. Измените заголовок формы на «Стандартные функции для работы с символами». Разместите на форме кнопку Button1, измените заголовок созданной кнопки на «Выполнить».
- 2) Разместите на форме пять компонентов Label согласно рисунку.
- 3) Разместите на форме пять компонентов Edit согласно рисунку.
- 4) Создайте процедуру обработки события OnClick для кнопки Выполнить. В программном коде созданной процедуры запишите операторы:

```
var c: Char; st: String;
```

```
begin
```

```
 st := Edit1.Text;
```

```
 c := st[1];
```

```
 Edit2.Text := IntToStr(Ord(c));
```

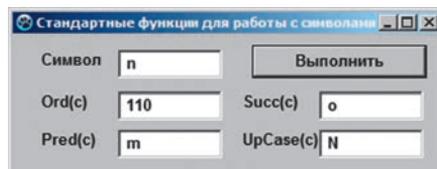
```
 Edit3.Text := Pred(c);
```

```
 Edit4.Text := Succ(c);
```

```
 Edit5.Text := AnsiUpperCase(c);
```

```
end;
```

- 5) Запустите проект на выполнение. Проверьте результат работы программы для буквы n.
- 6) Проверьте работу программы для других символов. Сохраните проект в папке Упражнение 32.



### Компьютерное тестирование



Выполните тестовое задание 32 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua)

## § 33. Величины строкового типа

**Строка** — это последовательность символов. Для хранения строк предназначен строковый тип данных (String), который широко используется в языке Object Pascal для обработки текстовых данных.

Строковые константы в тексте программы заключают в одинарные кавычки. Описание строковой переменной имеет вид:

```
var S: String;
```

Количество символов в строке может достигать  $2^8 - 1 = 255$  символов.

Длину строковой величины можно ограничить, если в ее описании указать максимально допустимое количество символов:

```
var S: String[10];
```

Обратиться к отдельному символу строковой переменной можно по его номеру в строке:  $S[i]$  —  $i$ -й символ строки  $S$ .

» Преобразовать текст «to learn» в текст «to reach», выполнив операции над символами заданной строки.

```
var x: String;
```

```
begin X := 'to learn';
```

```
 X[4] := X[7]; X[7] := 'c'; X[8] := 'h';
```

```
end;
```



**Конкатенацией** двух или более строк называют их слияние в одну строку. Эту операцию обозначают символом '+

Операция конкатенации позволяет объединять в одну строку переменные и константы строкового типа (String) и символьного типа (Char).

» Выполнить конкатенацию строк.

```
var X: String;
```

```
begin
```

```
 X := 'a' + 'Ab' + 'Bc'; // получаем X = 'aAbBc';
```

```
end;
```

### Стандартные функции для обработки строк

Для работы с величинами строкового типа в языке Object Pascal существуют стандартные процедуры и функции. Рассмотрим некоторые из них.

- 1) Функция `Length(S: String)` возвращает длину строки (количество символов).

```

>> var S: String; L: Integer;
 begin
 S := 'Hello!';
 L := Length(S); // L = 6
 end;

```

- 2) Функция `Pos(SubS, S: String)` возвращает позицию первого вхождения подстроки `SubS` в строку `S`. Нумерация символов начинается с единицы. В случае отсутствия подстроки в строке функция возвращает 0.

```

>> var S, SubS: String; P: Integer;
 begin
 S := 'Hi! How do you do?';
 SubS := 'do';
 P := Pos(SubS, S); // P = 9
 end;

```

- 3) Функция `Copy(S: String; n, Length: Integer)` возвращает подстроку строки `S` длиной `Length`, начиная с  $n$ -го символа.

```

>> var S1, S2: String;
 begin
 S1 := 'This is a test for Copy function.';
 S2 := Copy(S1, 11, 4); // S2 = 'test'
 end;

```

- 4) Процедура `Delete(S: String; i, n: Integer)` удаляет  $n$  символов из строки `S`, начиная с  $i$ -го символа.

```

>> var S: String;
 begin
 S := 'Hello, world!';
 Delete(S, 6, 7); // S = 'Hello!'
 end;

```

- 5) Процедура `Insert(SubS: String; S: String; Pos: Integer)` вставляет в строку `S` подстроку `SubS` в позицию `Pos`.

```

>> var S: String;
 begin
 S := 'Hello, world!';
 Insert(' my', S, 8); // S = 'Hello, my world!'
 end;

```

- 6) Функция `StringReplace(S, OldS, News, [rfReplaceAll, rflgnoreCase])` заменяет в строке `S` подстроку `OldS` на строку `News`. В квадратных скобках задаются параметры замены:

**rfReplaceAll:** заменять все вхождения подстроки `OldS` в строке `S`;

**rflgnoreCase:** пренебрегать регистром при поиске подстроки `OldS` в строке `S`.

```

>> var before, after: String;
begin // Замена всех вхождений A на THE
 before := 'This is a way to live A big life';
 after := StringReplace(before, 'a', 'THE', [rfReplaceAll, rflgnoreCase]);
 { after= 'This is THE wTHEy to live THE big life }
end;

```

- 7) Функция `ReverseString(S: String)` располагает символы строки `S` в обратном порядке. Для применения этой функции необходимо подключить к программе модуль `StrUtils`.

```

>> uses StrUtils;
var S1, S2: String;
begin
 S1 := '1234abcd';
 S2 := ReverseString(S1); // S2 = 'dcba4321'
end;

```

## Сравнение строк

Строки можно сравнивать друг с другом, используя лексикографический порядок (подобный порядку слов в словаре). Строки сравниваются посимвольно согласно кодам символов в таблице кодов.

```
var S1, S2, S3: String; B1, B2: Boolean;
```

```
begin
 S1 := '123'; S2 := '456';
 B1 := (S1 = S2); // B1 = False
 B2 := (S1 < S2); // B2 = True
 S1 := 'abcd'; S2 := 'bcd';
 B1 := (S1 < S2); // B1 = True
end;

```

## Вопросы для самопроверки

1. Что такое длина строки? Как ее определить?
2. Как можно выполнить слияние строк?



3. Какая функция возвращает позицию первого вхождения подстроки в строку?
4. Какое значение возвращает функция Pos(SubS, S), если подстрока SubS в строке S отсутствует?
5. Как можно получить подстроку заданной строки?
6. Как можно удалить подстроку из заданной строки?

### Упражнение 33



► Составить программу, выполняющую заданную замену символов в строке.

- 1) Создайте новый проект. Измените заголовок формы на «Поиск и замена». Разместите на форме кнопку Button1 и измените заголовок созданной кнопки на «Заменить».
- 2) Разместите на форме четыре компонента Label согласно рисунку. Для элемента Label4 задайте следующие значения свойств: AutoSize: False; WordWrap: True.
- 3) Разместите на форме три компонента Edit согласно рисунку.
- 4) Создайте процедуру обработки события OnClick для кнопки Заменить.

В программном коде созданной процедуры запишите операторы:

```
var S, S1, S2, S3: String;
begin
```

```
 S := Edit1.Text;
```

```
 S1 := Edit2.Text; S2 := Edit3.Text;
```

```
 S3 := StringReplace(S, S1, S2, [rfReplaceAll, rfIgnoreCase]);
```

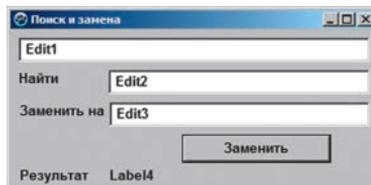
```
 Label4.Caption := S3;
```

```
end;
```

- 5) Запустите проект на выполнение. Проверьте работу программы для S = 'Why do You cry, Willy, why do You cry?';

S1 = 'cry'; S2 = 'smile'. Сохраните проект в папке Упражнение 33.

- 6) Измените программный код так, чтобы в предложении, приведенном в п. 5, буквы «W» и «w» были заменены символом «\*».



### Компьютерное тестирование

Выполните тестовое задание 33 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 34. Преобразование величин одного типа в другой

Вам уже приходилось преобразовывать строковые данные в числовые, числовые — в строковые и т. п. Рассмотрим функции языка Object Pascal для преобразования величин разных типов.

### Функции преобразования чисел в строки и строк в числа

Вы уже использовали функции `StrToInt`, `StrToFloat`, `IntToStr` и `FloatToStr`. Рассмотрим еще несколько примеров.

» Преобразовать целое число в строку цифр:

```
var a: Integer; s: String;
begin
 a := 12345;
 s := IntToStr(a); // s = '12345'
end;
```

» Преобразовать строку цифр в целое число:

```
var a: Integer; s: String;
begin
 s := '12345';
 a := StrToInt(s); // a = 12345
end;
```

» Преобразовать строку символов в действительное число (рис. 34.1):

```
var a: Real;
begin
 a := StrToFloat('12,45');
 a := 2 * a;
 Edit1.Text := FloatToStr(a);
end;
```

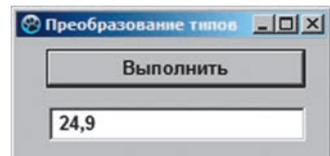


Рис. 34.1

! При вводе десятичных дробей следует помнить, что выбор разделителя зависит от настройки системы (см. с. 156).

В случае неправильного выбора разделителя программа прерывает работу и выводится окно с сообщением об ошибке (рис. 34.2).

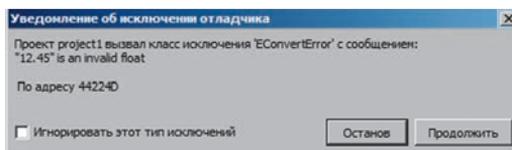


Рис. 34.2

## Вывод дробных чисел по формату

Для вывода дробных чисел с заданным количеством знаков в дробной части используется функция

`FormatFloat(const Format: String; Value: Extended): String;`

Данная функция преобразует значение действительного числа (Value) в строку, представленную по шаблону (параметр Format). Шаблон может содержать следующие символы форматирования:

|   |                                                                                                        |
|---|--------------------------------------------------------------------------------------------------------|
| . | Место расположения десятичного разделителя                                                             |
| 0 | Вывод цифры десятичной дроби в соответствующую позицию. Если цифры в позиции нет — выводится "0"       |
| # | Вывод цифры десятичной дроби в соответствующую позицию. Если цифры в позиции нет — не выводится ничего |

» Вывести действительное число с двумя знаками после запятой (рис. 34.3):

`var S: String;`

`begin`

`S := FormatFloat('#.##', Pi); // S := '3.14'`

`MessageDlg('Pi :=' + S, mtInformation, [mbOk], 0);`

`end;`

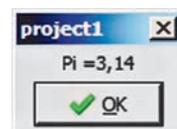


Рис. 34.3

## Неявное преобразование типов данных

Неявными называют такие преобразования типов переменных, которые происходят без непосредственного указания пользователя. В частности, при присваивании переменной вещественного типа целого значения оно преобразуется в действительное.

» `var d: Double; a: Integer;`

`begin`

`a := 100;`

`d := a; // верно`

`a := d; // ошибка`

`end;`

При присвоении переменной определенного типа значения, которое не может быть автоматически преобразовано согласно типу переменной, возникает ошибка компиляции:

Incompatible types:... (*Несовместимые типы...*).

» Найти ошибку и объяснить пример.

```
var a: Integer;
begin
 a := Sqrt(100); // Несовместимые типы
end;
```

Функция Sqrt(x) возвращает результат вещественного типа.

При выполнении программы будет получено сообщение об ошибке Incompatible types: 'Integer' and 'Extended'.

## Функции преобразования даты и времени

При создании проектов вам могут пригодиться следующие функции:

| Функция                    | Тип результата | Назначение                                                             |
|----------------------------|----------------|------------------------------------------------------------------------|
| Date                       | TDateTime      | Возвращает текущую дату                                                |
| Time                       | TDateTime      | Возвращает текущее время                                               |
| DayOfWeek(Date: TDateTime) | Integer        | Возвращает текущий номер дня недели: от 1 (воскресенье) до 7 (суббота) |

Функции преобразования даты и времени из формата TDateTime в строковый формат (у всех функций тип результата — String):

| Функция                            | Назначение                                 |
|------------------------------------|--------------------------------------------|
| DateTimeToStr(DateTime: TDateTime) | Преобразует дату и время DateTime в строку |
| DateToStr(Date: TDateTime)         | Преобразует дату Date в строку             |
| TimeToStr(Time: TDateTime)         | Преобразует время Time в строку            |

Для задания, ввода и вывода значений типа DateTime используют следующий формат: hh:mm:ss — для времени; dd.mm.yyyy — для даты.

» Присвоить значения типа DateTime:

```
var s: String; t, d: TDateTime;
begin
 s := '8:30:15'; t := StrToTime(s);
 s := '19.02.2016'; d := StrToDate(s);
end;
```

» Вывести значения текущих даты и времени (рис. 34.4):

```
var t, d: TDateTime;
begin
 t := Time; d := Date;
 ShowMessage ('текущая дата =' + DateToStr(d));
 ShowMessage ('текущее время =' + TimeToStr(t));
end;
```

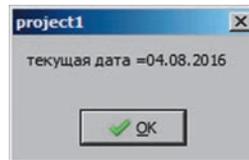


Рис. 34.4

### Вопросы для самопроверки



1. Как преобразовать значение вещественного типа в строку символов?
2. Как преобразовать значение целого типа в строку символов?
3. Как преобразовать строку символов в значение числового типа?
4. Как вывести десятичную дробь с заданным количеством знаков в дробной части?
5. В каких случаях происходит неявное преобразование типов данных?
6. Какие функции предназначены для работы с датой и временем?

### Упражнение 34



► Составить программу «Часы».

- 1) Создайте новый проект. Измените заголовок формы на «Часы». Разместите на форме два компонента Edit для вывода текущих значений времени и номера дня недели. Рядом с полем Edit2 создайте надпись (Label1) «День недели».
- 2) Добавьте на форму компонент Timer (см. рисунок).
- 3) Создайте процедуру обработки события OnTimer для объекта Timer. В программном коде запишите операторы вывода в поля Edit1 и Edit2 текущих времени и дня недели соответственно.

**procedure** TForm1.Timer1Timer(Sender: TObject);

**var** t: TDateTime; day: String;

**begin**

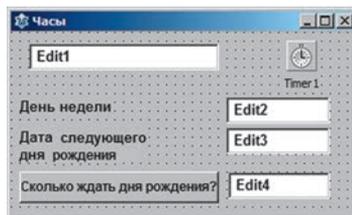
t := Time;

Edit1.Text := TimeToStr(t);

day := IntToStr(DayOfWeek(Date));

Edit2.Text := day;

**end;**



- 4) Добавьте на форму кнопку Сколько ждать дня рождения? и текстовые поля Edit3, Edit4 согласно рисунку. Рядом с полем Edit3 создайте надпись (Label2) «Дата следующего дня рождения».

- 5) Создайте процедуру обработки события OnClick для кнопки. В программном коде созданной процедуры запишите операторы для определения количества дней между датой следующего дня рождения и текущей датой:
- ```
var day: Double; S: String;
begin
  S := Edit3.Text; // ввод даты следующего дня рождения
  day := Date - StrToDate(S); // разница между двумя датами
  Edit4.Text := FloatToStr(day); { преобразование действительного
  числа day в строку символов }
end;
```
- 6) Проверьте работу программы. Почему в поле выводится отрицательное число? Исправьте ошибку. Сохраните проект в папке Упражнение 34.

Компьютерное тестирование

Выполните тестовое задание 34 с автоматической проверкой на сайте interactive.ranok.com.ua.

Практическая работа 14

Отладка готовой программы

Задание: создать проект для решения задачи.

Известны название планеты, радиус (км) ее орбиты, скорость движения (км/ч) по орбите. Вычислить продолжительность года на планете (период ее вращения). Отладить программный код. Проверить работу программы для заданных тестовых данных:

Название планеты	Радиус орбиты (км)	Скорость движения (км/ч)	Период вращения (сутки)
Земля	149 600 000	107 250	365,18
Венера	108 200 000	126 110	
Нептун	4 496 600 000	19 550	
Марс	227 900 000	86 870	

Оборудование: компьютер с установленной средой программирования Lazarus.

Ход работы

При работе за компьютером соблюдайте правила безопасности.

I. Размещение элементов управления на форме

- ▶ 1. Создайте новый проект. Разработайте интерфейс программы согласно рис. 1. Измените для объектов Label значение свойства WordWrap на True.

II. Отладка программного кода

- ▶ 2. Создайте процедуру обработки события OnClick для кнопки Вычислить. В программном коде созданной процедуры запишите операторы:

```
var Nazvanie: String; R, Period: Real;
```

```
Const Pi = 3.1415926;
```

```
begin
```

```
  Nazvanie := Edit1.Text; R = Edit2.Text;
```

```
  V = Edit3.Text;
```

```
  Period := 2 * Pi * R / V * 24;
```

```
  Edit4.Text := FloatToStr(Period);
```

```
  ShowMessage('Год на планете ' + 'Nazvanie' + ' длится ' +  
  FormatFloat('#0.00',Period) + ' суток');
```

```
end;
```

В программном коде допущены ошибки, возможно, вы заметите их при наборе текста программы и исправите, если нет — вам поможет исправить ошибки компилятор.

III. Тестирование программы

- ▶ 3. После исправления синтаксических ошибок проверьте работу программы для данных планеты Земля (рис. 2). Совпадает ли ответ с указанным в таблице? Если нет — проанализируйте выражение, вычисляющее значение Period : $\text{период вращения} = \frac{\text{длина орбиты}}{\text{скорость движения} \cdot 24}$, и исправьте ошибку.
- ▶ 4. Выполните вычисления для Венеры, Нептуна, Марса. Дополните таблицу.
- ▶ 5. Какие изменения нужно внести, чтобы результат вычислений выводился в земных годах?
- ▶ 6. Сохраните проект в папке Практическая работа 14. Завершите работу за компьютером.

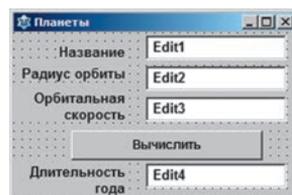


Рис. 1

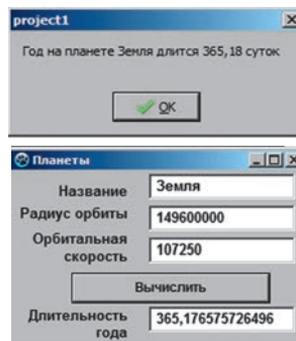
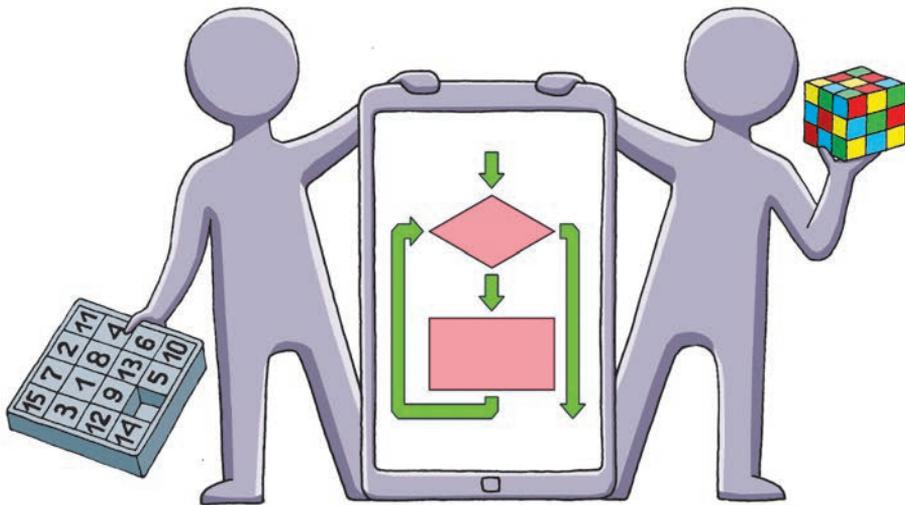


Рис. 2

Сделайте вывод: как исправлять синтаксические и логические ошибки в программном коде и анализировать результат работы программы.

РАЗДЕЛ 8

АЛГОРИТМЫ С ПОВТОРЕНИЯМИ И ВЕТВЛЕНИЯМИ



- § 35. Алгоритмы с ветвлениями
 - § 36. Составление программ с ветвлениями
 - § 37. Поливариантное ветвление
 - § 38. Компоненты выбора
 - § 39. Алгоритмы с повторениями. Цикл с предусловием
 - § 40. Алгоритмы с повторениями. Цикл с постусловием
 - § 41. Алгоритмы с повторениями. Цикл с параметром
 - § 42. Составление циклических алгоритмов обработки величин
- Практическая работа 15. Составление и выполнение алгоритмов с повторениями и ветвлениями для обработки величин

ПОВТОРЯЕМ

Алгоритм — это четко определенная последовательность указаний для выполнения действий, направленная на решение определенной задачи.

Из курса математики вам хорошо известны алгоритмы выполнения арифметических операций над многозначными числами; нахождения корней линейного и квадратного уравнений; деления отрезка на равные части; построения треугольника по заданным сторонам и т. п. В курсе информатики вы составляли алгоритмы не только вычислительных, а и игровых задач.

Вы уже знаете, что любой алгоритм можно построить, используя три базовые алгоритмические структуры: следование, ветвление, повторение. Для графического представления алгоритмов используют блок-схемы.

В *линейных алгоритмах* используют только структуру следования. Алгоритмы, в основе которых лежит структура ветвления, называют *алгоритмами с ветвлениями*. Алгоритмы, в основе которых лежит структура повторения, называют *циклическими*.



1. Что такое алгоритм?
2. Какие базовые алгоритмические структуры используют для конструирования алгоритмов?
3. С какой целью составляют блок-схемы алгоритмов?
4. Какую алгоритмическую структуру используют в линейных алгоритмах?



В этом разделе вы ознакомитесь с правилами реализации алгоритмов с ветвлениями и повторениями в языке программирования *Object Pascal*.

§ 35. Алгоритмы с ветвлениями



Ветвление — это алгоритмическая структура, в которой в зависимости от соблюдения или несоблюдения некоторого условия выполняется та или иная последовательность действий.

Для организации ветвлений в языке Object Pascal предназначены условные операторы `If...Then` и `If...Then...Else`.

Условный оператор `If...Then`

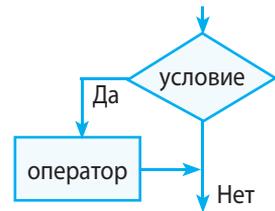
Оператор `If...Then` предназначен для выполнения некоторой последовательности действий в случае, если указанное условие является истинным. Этот условный оператор соответствует алгоритмической конструкции «неполное ветвление».

Синтаксис оператора:

```
If <условие> Then <оператор>;
```

Оператор `If` проверяет истинность указанного условия. В качестве условия используется логическое выражение (или логическая переменная как частный случай логического выражения). Условие истинно, если логическое выражение принимает значение `True`, и ложно в противоположном случае.

Если условие истинно (`True`), программа выполнит действие, указанное в части `<оператор>`. Если же условие ложно (`False`), то управление передается оператору, следующему после оператора `If`.



» Увеличить значение переменной `A` на единицу, если ее текущее значение меньше 5.

Оператор ветвления имеет вид: `If A < 5 Then A := A + 1;`

Оператор `A := A + 1` выполняется только в том случае, когда истинным является условие `A < 5`:

Исходное значение <code>A</code>	Значение условия	Оператор <code>A := A + 1</code>	Значение <code>A</code> после выполнения оператора <code>If</code>
1	True	Выполняется	2
5	False	Не выполняется	5
10	False	Не выполняется	10

Если в случае истинности условия нужно выполнить некоторую последовательность действий, операторы, реализующие эти действия, заключаются в операторные скобки **begin...end**. Такую конструкцию называют **составным оператором**.

» Поменять местами значения переменных a и b , чтобы было верным неравенство $a \leq b$.

```

var a, b, x: Integer;
begin
  a := StrToInt(TextBox('Введите a', 'a=', '0'));
  b := StrToInt(TextBox('Введите b', 'b=', '0'));
  if a > b then begin
    x := a; // для обмена значениями между переменными a и b
    a := b; // используется дополнительная переменная x
    b := x
  end;
  Edit1.Text := IntToStr(a);
  Edit2.Text := IntToStr(b);
end;

```

Если условие ложно, управление передается оператору
 Edit1.Text := IntToStr(a).

Условный оператор If...Then...Else

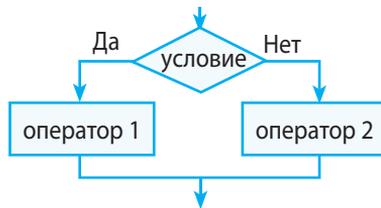
Алгоритмической конструкции «полное ветвление» соответствует условный оператор **If...Then...Else**.

Синтаксис оператора:

```

if <условие> then <оператор 1>
  else <оператор 2>;

```



! Перед ключевым словом **Else** точка с запятой не ставится.

Если результатом проверки условия является значение **True**, то выполняется блок действий <оператор 1>, следующий после служебного слова **Then**. Если проверка условия дала результат **False**, выполняется блок действий <оператор 2>, следующий после служебного слова **Else**.

» Увеличить переменную a на 1, если $a < 5$, и уменьшить на 1, если $a \geq 5$.
 If $a < 5$ Then $a := a + 1$ Else $a := a - 1$;

» Проверить существование треугольника со сторонами a , b , c .
Условие существования треугольника с заданными длинами сторон: сумма двух любых длин сторон должна быть больше третьей.

var a, b, c : Integer;

begin

a := StrToInt(InputBox('Введите a', 'a='; '0'));

b := StrToInt(InputBox('Введите b', 'b='; '0'));

c := StrToInt(InputBox('Введите c', 'c='; '0'));

If (a < b + c) and (b < a + c) and (c < a + b) Then

 Edit1.Text := 'треугольник существует'

Else Edit1.Text := 'треугольник не существует';

end;

Вопросы для самопроверки



1. Дайте определение ветвления как алгоритмической конструкции.
2. Какие выражения используют в качестве условия в условном операторе?
3. Как выполняется условный оператор в неполной форме?
4. Как записывается и выполняется условный оператор в полной форме?
5. Исходные значения переменных $A = -3$; $B = 5$. Найдите значения переменных A и B после выполнения следующих операторов:
 - а) If $A > B$ Then $A := A - B$ Else $B := B - A$;
 - б) If $A <> B$ Then $A := B$;
 - в) If $A < B$ Then $A := 2 * A$ Else $B := B * A$;
6. Запишите условные операторы, с помощью которых можно выполнить следующие действия:
 - а) проверить, является ли число a четным;
 - б) из большего из данных чисел a и b вычесть меньшее;
 - в) проверить, есть ли среди чисел a , b , c равные.

Упражнение 35



» Определить наибольшее из трех чисел a , b , c .

- 1) Создайте новый проект. Измените заголовков формы на «Поиск максимального». Разместите на форме четыре компонента Edit для ввода значений a , b , c и вывода значения Max наибольшего из трех чисел согласно рисунку.

- 2) Добавьте на форму кнопку Button1 и измените заголовок добавленной кнопки на «Найти Max».
- 3) Создайте процедуру обработки события OnClick для кнопки Найти Max. В программном коде созданной процедуры запишите операторы:


```
var a, b, c, Max: Integer;
begin
  a := StrToInt(Edit1.Text); b := StrToInt(Edit2.Text); c := StrToInt(Edit3.Text);
  If a > b Then Max := a Else Max := b;
  If c > Max Then Max := c;
  Edit4.Text := IntToStr(Max);
end;
```
- 4) Проверьте работу программы для разных наборов чисел. Сохраните проект в папке Упражнение 35.
- 5) Измените программный код так, чтобы программа находила наименьшее из трех чисел.
- 6) Добавьте на форму кнопку Button2 и измените ее заголовок на «Упорядочить a, b, c». Создайте процедуру обработки события OnClick для этой кнопки, чтобы значения переменных a, b и c были упорядочены по неубыванию (поменяйте местами a, b и c так, чтобы стало $a \leq b \leq c$).



Компьютерное тестирование



Выполните тестовое задание 35 с автоматической проверкой на сайте interactive.ranok.com.ua.

§ 36. Составление программ с ветвлениями

Вложенные ветвления

При решении многих задач возникает необходимость в случае выполнения (либо невыполнения) некоторого условия снова делать выбор. В такой ситуации применяют так называемые **вложенные ветвления**: в условном операторе после служебного слова Then (или Else) используют оператор If.

» Записать условный оператор для вычисления значения функции:

$$y = \begin{cases} x, & \text{если } x < 0; \\ x^2, & \text{если } 0 \leq x \leq 5; \\ 2x, & \text{если } x > 5; \end{cases}$$

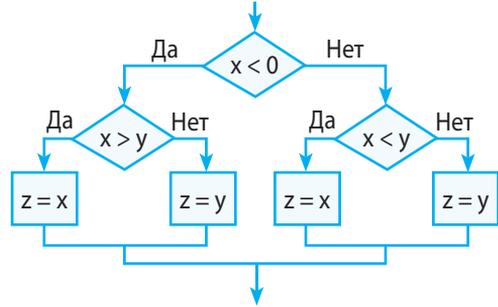
```
If x < 0 Then y := x
Else If x <= 5 Then y := x * x
Else y := 2 * x;
```

При записи программного кода используйте структурированную запись операторов, то есть связанные пары ключевых слов или их группы (например, Then...Else, **begin...end**) записывайте на одной вертикали. Это сделает текст программы нагляднее и понятнее.

» Вычислить:

$$Z = \begin{cases} \max(x, y) & \text{при } x < 0; \\ \min(x, y) & \text{при } x \geq 0. \end{cases}$$

```
If x < 0 Then
  If x > y Then z := x
  Else z := y
Else If x < y Then z := x
Else z := y;
```



» Найти корни квадратного уравнения вида $ax^2 + bx + c = 0$. Процедура обработки события для кнопки Найти корни уравнения (рис. 36.1):

```
var a, b, c, x1, x2, D: Real;
{ ввод коэффициентов a, b, c }
a := StrToFloat(Edit1.Text);
b := StrToFloat(Edit2.Text);
c := StrToFloat(Edit3.Text);
D := b*b - 4*a*c; // вычисление дискриминанта
Edit4.Text := FloatToStr(D); // вывод значения дискриминанта
If D > 0 Then begin
  x1 := (-b + Sqrt(D)) / (2*a); // вычисление и вывод значений корней
  x2 := (-b - Sqrt(D)) / (2*a);
  Edit5.Text := FormatFloat('0.##', x1);
  Edit6.Text := FormatFloat('0.##', x2);
end
Else
  If D = 0 Then begin
    x1 := -b / (2*a);
    Edit5.Text := FormatFloat('0.##', x);
  end
  Else Edit5.Text := 'Корней нет';
end;
```



Рис. 36.1

Вопросы для самопроверки



1. Что такое вложенные ветвления?
2. В каких случаях в записи оператора If используют операторные скобки **begin...end**?
3. Какие значения примут переменные A и B после выполнения условного оператора If для данных наборов исходных значений?

Фрагмент программного кода	Исходные значения A, B	A	B
If A < 0 Then	A = -3, B = 5		
If A > B Then A := B	A = 3, B = 5		
Else B := A	A = 8, B = 5		
Else If A > B Then B := A	A = -3, B = -5		
Else A := B;			

4. Запишите условный оператор, выполняющий следующее действие: вывести в текстовое поле Edit1 количество цифр в целом числе A, которое принадлежит диапазону от 1 до 999.
5. Запишите условный оператор, выполняющий следующее действие: наибольшее из трех разных значений переменных действительного типа X, Y и Z уменьшить на 0,3.
6. Создайте программу, с помощью которой определяется век, к которому относится указанный год y ($y > 0$). (Учтите, что, например, началом XX в. был 1901, а не 1900 год.)

Упражнение 36



► Составить программу «Угадай-ка!» по следующему плану:

- 1) Компьютер загадывает случайное число от 1 до 100.
- 2) Пользователь совершает попытку угадать число.
- 3) Компьютер сравнивает введенное число с загаданным и дает ответ «Мало» или «Много». Если пользователь угадал число, выводится надпись «Ура! Ты угадал число!».

- 1) Разместите на форме компоненты согласно рис. 1*.



Рис. 1

* Как добавлять рисунки из внешних файлов, вы ознакомитесь в § 47.

- 2) В окне Инспектор объектов задайте значения свойств компонентов согласно таблице:

Свойство	Значение	Свойство	Значение
Form1.Caption	Угадай-ка!	Button1.Caption	Загадать новое число
Label1.Caption	Ваша попытка:	Button1.WordWrap	True
Label3.Caption	Ура! Ты угадал число!	Button2.Caption	Проверить

- 3) Опишите переменную b (загаданное компьютером число) как глобальную:

```
var Form1: TForm1;
    b: Integer;
```

- 4) Создайте процедуру обработки события OnClick для кнопки Загадать новое число. В программном коде процедуры запишите операторы:

```
begin
    Randomize;
    b := Random(100);
    Label2.Visible := False;
    Label3.Visible := False;
    Edit1.Clear;
    Edit1.SetFocus;
```

end;

- 5) Создайте процедуру обработки события OnClick для кнопки Проверить (рис. 2, 3). В программном коде процедуры запишите операторы:

```
var a: Integer;
begin
    a := StrToInt(Edit1.Text); { a — число, вводимое пользователем }
```

```
{ Если число угадано, объект Label3 становится видимым, а объект Label2 (для слов «Мало» и «Много») — невидимым }
```

```
If a = b Then begin
    Label3.Visible := True;
    Label2.Visible := False;
end
```

```
Else If a < b Then begin
    Label2.Visible := True;
```

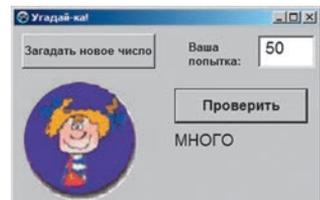


Рис. 2

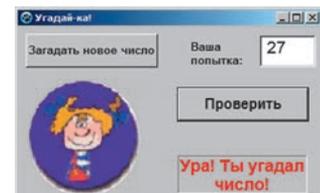


Рис. 3

```

Label2.Caption := 'МАЛО';
end
Else begin
Label2.Visible := True;
Label2.Caption := 'МНОГО';
end;
end;

```

- 6) Запустите проект на выполнение. Сохраните проект в папке Упражнение 36.
- 7) Как нужно действовать, чтобы отгадать число как можно быстрее?



Компьютерное тестирование

Выполните тестовое задание 36 с автоматической проверкой на сайте interactive.ranok.com.ua.



§ 37. Поливариантное ветвление

В большинстве случаев, когда приходится использовать несколько команд ветвления, вложенных друг в друга, предпочтительнее воспользоваться оператором выбора.

Оператор выбора Case

Оператор выбора Case позволяет выбрать один из нескольких операторов в зависимости от значения селектора.

Синтаксис оператора выбора Case:

```

Case <Селектор> of
  <список меток 1>: <оператор1>;
  ...
  <список меток N>: <операторN>;
end;

```

Здесь <Селектор> — это переменная или выражение, значение которой будет проверяться; <список меток> — некоторый набор возможных значений.



Селектор — переменная или выражение порядкового типа (целого, символьного и т. п.).

Список меток можно задавать в виде отдельного значения, перечня значений либо диапазона значений.

Выполнение оператора `Case` начинается с вычисления значения селектора. Затем выполняется оператор, одна из меток которого совпадает со значением селектора. Оператор может быть как простым, так и составным. После выполнения этого оператора управление передается оператору, записанному после ключевого слова **end**. То есть может быть выполнена только одна из ветвей оператора выбора.

Ветви оператора `Case` будут перебираться последовательно, до первого совпадения значения <Селектор> со значением одной из меток. Если значение <Селектор> не равно ни одному из значений меток, ни одна из ветвей выполнена не будет.

➤ Определить, каким является однозначное число x — простым или составным.

Case x of

1: ShowMessage ('Это — 1');

2, 3, 5, 7: ShowMessage ('Это — простое число');

4, 6, 8, 9: ShowMessage ('Это — составное число');

end;

Если существуют действия, которые нужно выполнить в случае, когда значение <Селектор> не совпадает ни с одним из значений меток, то используют оператор `Case` в полной форме:

Case <Селектор> of

<список меток 1>: <оператор 1>;

...

<список меток N-1>: <оператор N-1>;

Else <оператор N>;

end;

При использовании полной формы записи оператора в случае, когда значение <Селектор> не совпадает ни с одним из значений меток, будет выполнена группа операторов, расположенная в ветви `Else`.

➤ Указать словесную оценку в зависимости от количества полученных баллов:

Case N of // N — количество баллов

4..6: ShowMessage ('Ваша оценка — удовлетворительно');

7..9: ShowMessage ('Ваша оценка — хорошо');

10..12: ShowMessage ('Ваша оценка — отлично');

Else ShowMessage ('Выполните тест еще раз');

end;

Вопросы для самопроверки



1. Для решения каких задач используют оператор выбора?
2. Какие выражения можно использовать в качестве селектора в операторе Case?
3. Как в операторе Case можно задать список меток?
4. Дополните таблицу выполнения оператора Case для разных исходных значений переменной A.

Фрагмент программного кода	Исходное значение A	Значение A после выполнения оператора Case
Case A mod 10 of 0..3: A := A * 2; 3..5: A := A + 10; 6..9: A := A * 10; end;	13	
	40	
	88	
	11	
	26	

5. Рассмотрите фрагмент программы для определения времени года по номеру месяца. Представьте данный фрагмент программы в виде команды выбора:


```

            If (N = 12) or (N = 1) or (N = 2) Then S := 'зима'
            Else If (N >= 3) and (N <= 5) Then S := 'весна'
            Else If (N >= 6) and (N <= 8) Then S := 'лето'
            Else If (N >= 9) and (N <= 11) Then S := 'осень'
            Else ShowMessage ('Такого номера месяца нет');
            
```
6. Напишите программу, которая по номеру дня недели определяет, сколько в этот день в вашем классе уроков: K1 — количество уроков по гуманитарным предметам, K2 — по естественно-математическим дисциплинам.

Упражнение 37



- Создать проект для решения задачи.
Кассир продает билеты на автобус, курсирующий по маршруту от города A в город B. Цена билета зависит от расстояния, которое нужно проехать пассажиру:

$$X = \begin{cases} 20 \text{ грн, до } 50 \text{ км;} \\ 75 \text{ грн, от } 51 \text{ до } 100 \text{ км;} \\ 125 \text{ грн, от } 101 \text{ до } 150 \text{ км;} \\ 175 \text{ грн, от } 151 \text{ км.} \end{cases}$$

Сколько будут стоить N билетов до населенного пункта, расстояние* до которого вводится с клавиатуры?

- 1) Создайте новый проект. Измените заголовок формы на «Кассир».
- 2) Возможный интерфейс программы приведен на рисунке. Разместите на форме два компонента Edit для ввода значений расстояния и количества билетов и два компонента Label. Измените заголовки надписей.
- 3) Добавьте на форму кнопку Button1 и измените заголовок добавленной кнопки на «Общая сумма». Создайте процедуру обработки события OnClick для этой кнопки:



```
procedure TForm1.Button1Click(Sender: TObject);
```

- 4) Опишите необходимые переменные.
`var X, V: Real; R, N: Integer; { R — расстояние, X — цена билета, N — количество билетов, V — общая стоимость билетов }`

- 5) В программном коде процедуры Button1Click запишите операторы:

```
begin
```

```
  R := StrToInt(Edit1.Text);
```

```
  N := StrToInt(Edit2.Text);
```

```
  Case R of
```

```
    1..50: X := 20;
```

```
    51..100: X := 75;
```

```
    101..150: X := 125;
```

```
    Else X := 175;
```

```
  end;
```

```
  V := X * N;
```

```
  ShowMessage ('Стоимость билетов ' + FloatToStr(V) + ' грн');
```

```
end;
```

- 6) Запустите проект на выполнение. Проверьте работу программы для разных значений расстояния и количества билетов. Сохраните проект в папке Упражнение 37.

Компьютерное тестирование

Выполните тестовое задание 37 с автоматической проверкой на сайте interactive.ranok.com.ua.



* Значение расстояния является целым числом, большим или равным 1.

§ 38. Компоненты выбора

При работе с разными программами пользователю зачастую приходится делать выбор определенных элементов или настроек из предложенных в диалоговом окне с помощью таких элементов управления, как переключатель и флажок (рис. 38.1).

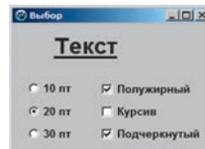


Рис. 38.1

Компонент Переключатель (RadioButton)

Переключатели используют в ситуациях, когда нужно выбрать один из нескольких существующих вариантов. Переключатели всегда размещаются на форме группами, в которых нужно выбрать только один переключатель. При выборе другого переключателя предыдущий выключается.

Компонент RadioButton  (вкладка Standard на Палитре компонентов) состоит из круглого окошка и текстовой строки. В окошке появляется отметка выбора данного варианта, а текстовая строка объясняет его назначение. За содержание текстовой строки отвечает свойство Caption. Основное свойство элемента RadioButton — Checked (*Выбрано*) логического типа (Boolean) — может принимать значение True или False. У выбранного элемента свойство Checked равно True.

Чтобы переключатели были удобно размещены на форме, их нужно выровнять.

Для **выравнивания переключателей** следует:

- 1) выбрать все переключатели группы, щелкнув их поочередно, удерживая нажатой клавишу Shift;
- 2) щелкнуть по группе правой кнопкой мыши для вызова контекстного меню;
- 3) выбрать команду Выровнять;
- 4) в открывшемся диалоговом окне Выравнивание (рис. 38.2) выбрать нужные параметры выравнивания, например горизонтальное выравнивание — Левые края, вертикальное выравнивание — Равномерно. Объекты расположатся согласно выбранным параметрам.

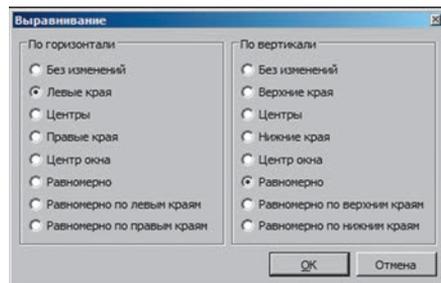


Рис. 38.2

» Запрограммировать включение переключателя
 RadioButton2 (рис. 38.3).

```
RadioButton1.Checked = False;
RadioButton2.Checked = True;
RadioButton3.Checked = False;
```

Для снятия выбора в программе нужно записать оператор:
 RadioButton2.Checked := False; // теперь ни один из компонентов не выбран.

Основным событием компонента RadioButton является OnClick. Можно **создать общий обработчик события OnClick** для всех переключателей на форме (в группе):

- 1) выбрать группу переключателей;
- 2) в окне Инспектор объектов на вкладке События дважды щелкнуть в строке OnClick.

После этого процедура TForm1.RadioButton1Click будет выполняться для всех переключателей, которые были выбраны в момент ее создания.

```
» procedure TForm1.RadioButton1Click(Sender: TObject);
var V: Integer;
begin
```

```
  with (Sender as TRadioButton) do
  V := StrToInt(copy(Name,12,1)); // Для RadioButton1 V = 1,
                                // для RadioButton2 V = 2 и т. д.

  Case V of
    1: Form1.Color := clRed;
    2: Form1.Color := clYellow;
    3: Form1.Color := clGreen;
  end;
end;
```

Оператор with (Sender as TRadioButton) означает, что свойство Name, используемое в нем, является свойством именно того из элементов управления RadioButton, с которым состоялось событие.

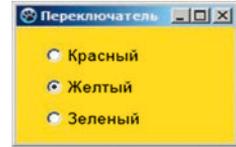


Рис. 38.3

Компонент Флажок (CheckBox)

Элемент управления CheckBox используется в программе для визуализации состояний включено / выключено, когда среди независимых параметров или характеристик пользователю необходимо выбрать нужные.

Компонент CheckBox (вкладка Standard на Палитре компонентов) является сочетанием квадратного окошка, которое и показывает состояние компонента (выбран / не выбран), и текстовой строки (свойство Caption) — заголовка, объясняющего назначение флажка.

Основным свойством компонента `CheckBox` является свойство `Checked` логического типа (`Boolean`). Щелчок компонента `CheckBox` меняет значение свойства `Checked` на противоположное. Изменение состояния переключателя доступно и в программе:

```
CheckBox1.Checked = False; // снять отметку
CheckBox2.Checked = True; // установить отметку
```

При каждом изменении состояния флажка, то есть при его установке или сбросе, для данного элемента управления происходит событие `OnClick`. Выяснить, установлен флажок или сброшен, можно по значению свойства `Checked`.

После двойного щелчка компонента `CheckBox` на форме в режиме разработки создается процедура обработки события:

```
procedure TForm1.CheckBox1Click(Sender: TObject);
```

В коде процедуры проверяется состояние флажка и активируются параметры, связанные с флажком:

```
If CheckBox1.Checked Then
    <активация параметров, связанных с флажком>
Else
    <отключение параметров, связанных с флажком>;
```

Вопросы для самопроверки



1. В каких случаях целесообразно использовать компонент `RadioButton`?
2. Какие значения может принимать свойство `Checked` компонента `RadioButton`?
3. Какое событие является основным для компонента `RadioButton`?
4. Как создать общий обработчик события `OnClick` для группы переключателей?
5. С какой целью в программе может быть использован компонент `CheckBox`?
6. Как организовать проверку значения свойства `Checked` при изменении состояния флажка?

Упражнение 38



- Составить программу, в которой реализована головоломка Лойда: из заданного набора чисел выбрать те, сумма которых равна 50.
- 1) Создайте новый проект. Измените заголовок формы на «Головоломка Лойда».

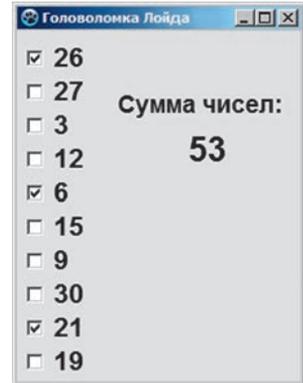
- 2) Добавьте на форму десять компонентов CheckBox согласно рисунку. Измените заголовки компонентов CheckBox на числа из набора, предложенного С. Лойдом: 25, 27, 3, 12, 6, 15, 9, 30, 21, 19. Добавьте на форму два компонента Label. Измените заголовок надписи Label1 на «Сумма чисел:». Надпись Label2 предназначена для вывода суммы чисел, соответствующих выбранным флажкам.

- 3) Опишите переменную S как глобальную:
- ```
var Form1: TForm1;
```

```
S: Integer;
```

- 4) Создайте процедуру обработки события OnCreate для формы. Переменная S принимает значение 0 при загрузке формы:

```
procedure TForm1.FormCreate(Sender:
TObject);
begin
 S := 0;
end;
```



- 5) Создайте общий для всех флажков обработчик события OnClick. В программном коде реализуется следующий алгоритм: если флажок установлен, соответствующее число прибавляется к сумме S, если флажок сброшен, соответствующее число вычитается из суммы S. После этого значение суммы выводится в надпись Label2.

```
procedure TForm1.CheckBox1Click(Sender: TObject);
begin
 with (Sender as TCheckBox) do begin
 If Checked Then
 S := S + StrToInt(Caption)
 Else S := S - StrToInt(Caption);
 end;
 Label2.Caption := IntToStr(S);
end;
```

- 6) Запустите проект на выполнение. Проверьте работу программы. Решите головоломку. Сохраните проект в папке Упражнение 38.

## Компьютерное тестирование

Выполните тестовое задание 38 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 39. Алгоритмы с повторениями. Цикл с предусловием



**Повторение (цикл)** — это алгоритмическая структура, с помощью которой одна и та же последовательность действий выполняется несколько раз.

Для записи алгоритмов с повторением (циклов) на языке Object Pascal используют *три вида операторов цикла*: с параметром, с предусловием и с постусловием.



Серию повторяющихся при выполнении цикла команд называют **телом цикла**.

Каждое выполнение тела цикла называют **итерацией**.

### Оператор цикла с предусловием While

Оператор повторения While (*цикл «Пока»*) предназначен для организации повторного выполнения серии команд, пока остается истинным условие выполнения цикла.

Синтаксис оператора:

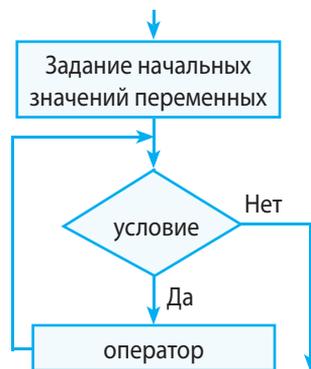
```
While <условие> do <оператор>;
```

Здесь <условие> — логическое выражение, являющееся условием выполнения цикла;

<оператор> — простой или составной оператор, выполняемый при каждой итерации.

Выполнение оператора цикла While начинается с вычисления значения логического выражения — условия повторения цикла. Если условие истинно, то выполняются операторы тела цикла и управление возвращается на проверку условия. Если же условие ложно, то выполняется следующий после оператора While оператор.

Если при первой проверке условие окажется ложным, тело цикла While не выполнится ни разу.



» Для разных исходных значений переменной X определить значение данной переменной после выполнения цикла и количество повторений:

```
While X <= 10 do X := X + 1;
```

| Исходное значение X | Значение X после выполнения цикла | Количество повторений |
|---------------------|-----------------------------------|-----------------------|
| 7                   | 11                                | 4                     |
| 10                  | 11                                | 1                     |
| 15                  | 15                                | 0                     |

» Данный фрагмент программы иллюстрирует бесконечный цикл.  
 Num := 0;

```
While Num < 20 do
 Label1.Caption := IntToStr(Num)
```

В теле цикла переменная Num имеет постоянное значение, поэтому условие Num < 20 (0 < 20) всегда верно и представленный цикл является бесконечным.

**!** Для прерывания бесконечного цикла нужно нажать сочетание клавиш Ctrl + Break.

Чтобы увидеть изменение значений в ходе выполнения цикла, в тело цикла включают вызов метода Application.ProcessMessages.

» Разработать программу-таймер, которая будет последовательно выводить числа от 1 до 100 с интервалом 0,1 с. Пусть переменная A содержит текущее значение выводимого числа. Пока A меньше 100, увеличивать значение A на единицу и выводить значение переменной в Label (рис. 39.1).

Для приостановки работы программы использовать процедуру Sleep.

```
var A: Integer;
begin
 A := 1; // исходное значение A
 While A < 100 do
 begin
 A := A + 1;
 Label1.Caption := IntToStr(A);
 Application.ProcessMessages; // обработка сообщений
 Sleep(100); // остановка на 0,1 с
 end;
 end;
```

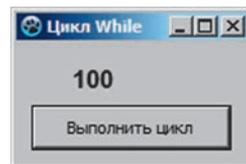


Рис. 39.1

## Алгоритм вычисления суммы $n$ членов числовой последовательности

Рассмотрим алгоритм вычисления  $n$ -го члена некоторой последовательности чисел и суммы  $n$  ее членов:

- 1) задать значения  $A_1$  — первого члена последовательности,  $n$  — количества членов, которые нужно вычислить;
- 2) задать исходные значения счетчика членов последовательности ( $i := 0$ ) и суммы ( $S := 0$ );
- 3) пока номер вычисленного  $i$ -го члена последовательности не достигнет значения  $n$  (заданного количества членов), повторять действия:
  - номер текущего слагаемого увеличить на 1;
  - вычислить значение следующего слагаемого  $A$ ;
  - вычисленное значение  $A$  прибавить к сумме  $S$ .

» Найти сумму 20 элементов последовательности чисел 2,5; 3,0; 3,5  
 { Задаются исходные значения }

```
S := 0; // суммы S
i := 0; // счетчика слагаемых
A := 2.5; // первого слагаемого A
While i < 20 do begin
 S := S + A; // A прибавляется к сумме S
 A := A + 0.5; // вычисляется следующее значение слагаемого A
 i := i + 1
end;
```

» Найти сумму всех элементов последовательности  $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \dots$ , значения которых не меньше 0,01 (рис. 39.2).

```
var i: Integer; A, S: Double;
begin
 { Задаются исходные значения }
 S := 0; // суммы S
 A := 1; // первого слагаемого A
 i := 0; // счетчика слагаемых
 While A > 0.01 do begin
 S := S + A;
 i := i + 1;
 A := 1/i; // вычисление текущего слагаемого A
```

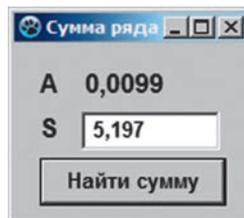


Рис. 39.2

```
Label2.Caption := FormatFloat('0.####', A);
end;
Edit1.Text := FormatFloat('0.####', S);
end;
```

### Вопросы для самопроверки



1. Дайте определение цикла как алгоритмической структуры.
2. Объясните синтаксис и правила выполнения оператора цикла с предусловием.
3. В каком случае тело цикла While не выполнится ни разу?
4. Определите значение переменной S после выполнения операторов:
  - a)  $S := 0; i := 0; \text{While } i < 5 \text{ do } i := i + 1; S := S + i;$
  - б)  $S := 0; i := 5; \text{While } i > 1 \text{ do } S := S + i; i := i - 1; \text{end};$
5. Составьте программу для нахождения суммы чисел, кратных 3, которые расположены в интервале (100; 300).
6. Составьте программу для нахождения суммы десяти элементов последовательности 1,0; 1,2; 1,4; 1,6...

### Упражнение 39



- Разработать проект «Числа Фибоначчи», предназначенный для вычисления  $n$ -го числа Фибоначчи.

Числами Фибоначчи называют числа, которые находят по следующему правилу:  $F_1 = F_2 = 1; F_n = F_{n-1} + F_{n-2}$ .  
 $F_1 = F_2 = 1$ , поэтому вычисление начинается с  $n = 3$ .

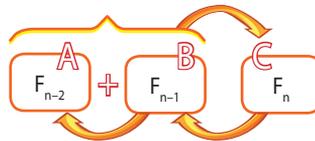


Рис. 1

Чтобы вычислить  $n$ -ое число Фибоначчи, нужно:

- 1) выделить переменные A и B (рис. 1) для сохранения двух текущих чисел Фибоначчи  $F_{n-1}$  и  $F_{n-2}$ ;
  - 2) сумма чисел  $A + B$  занести в переменную C;
  - 3) на следующей итерации цикла:
    - $B = F_{n-1}$  становится  $(n-2)$ -м членом ряда, поэтому  $A := B$ ;
    - $C = F_n$  становится  $(n-1)$ -м членом, поэтому  $B := C$ .
- 1) Создайте новый проект.

2) Разработайте интерфейс программы согласно рис. 2.

3) Опишите переменные, необходимые для реализации алгоритмов выполнения задач:

- $n$  — переменная целого типа для сохранения номера искомого числа Фибоначчи;
- $i$  — переменная целого типа для сохранения номера текущего элемента последовательности;
- $A, B, C$  — переменные целого типа для сохранения  $(i-2)$ -го,  $(i-1)$ -го,  $i$ -го элементов последовательности при каждой итерации цикла.

4) Создайте процедуру обработки события OnClick для кнопки Найти число с номером  $n$ . В программном коде процедуры Button1Click запишите операторы для вычисления  $n$  первых чисел Фибоначчи:

```
n := StrToInt(Edit1.Text);
```

```
A := 1; B := 1; i := 2;
```

```
While i < n do begin
```

```
 i := i + 1;
```

```
 C := A + B;
```

```
 A := B; B := C;
```

```
 Application.ProcessMessages; // обработка сообщений
```

```
 Sleep(500); // остановка на 0,5 с
```

```
 Edit2.Text := IntToStr (C);
```

```
end;
```

5) Запустите проект на выполнение. Проверьте работу программы. Сохраните проект в папке Упражнение 39.

6) Добавьте на форму компонент Edit. Дополните программный код операторами для вычисления суммы  $n$  первых чисел Фибоначчи и вывода результата в созданное текстовое поле (Edit3). Проверьте действие кнопки.

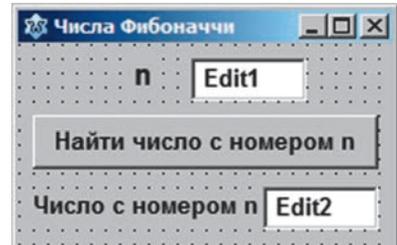


Рис. 2



## Компьютерное тестирование

Выполните тестовое задание 39 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 40. Алгоритмы с повторениями. Цикл с постусловием

### Оператор цикла с постусловием REPEAT

Оператор цикла Repeat используют в тех случаях, когда нужно сначала выполнить некоторую последовательность действий, а потом определить, нужно ли повторять эти действия еще раз.

Синтаксис оператора Repeat:

```
Repeat
<оператор>
Until <условие>;
```

Здесь <условие> — логическое выражение, являющееся условием выхода из цикла;

<оператор> — оператор, выполняемый при каждой итерации (тело цикла).

Порядок выполнения цикла Repeat:

- 1) выполняется <оператор> — указания, которые входят в тело цикла;
- 2) проверяется <условие>. Если условие ложно, то операторы тела цикла повторяются, если же условие истинно — цикл Repeat прекращает свою работу и выполняется следующий оператор программы.

**!** Тело цикла — блок операторов между Repeat и Until — обязательно выполняется хотя бы один раз.



» Вывести четные числа от 2 до 12 (рис. 40.1).

```
var k: Integer;
begin
 k := 0;
 Repeat
 k := k + 2;
 Label1.Caption := IntToStr(k);
 Until k >= 12;
end;
```

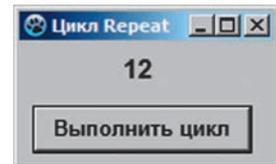


Рис. 40.1

» Определить, сколько первых натуральных чисел нужно сложить, чтобы получить сумму больше 1000 (рис. 40.2).

```
var n, S: Integer;
begin
 n := 0; S := 0;
 Repeat
 n := n + 1; S := S + n;
 Until S > 1000;
 Edit1.Text := IntToStr(n);
end;
```

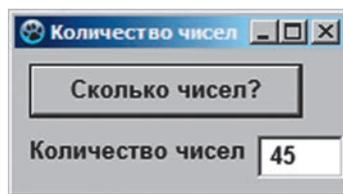


Рис. 40.2

» Найти сумму цифр натурального числа  $n$  (рис. 40.3).

```
var n, d, S: Integer;
begin
 n := StrToInt(Edit1.Text);
 S := 0;
 Repeat
 d := n mod 10; { найдена последняя цифра
 числа }
 S := S + d; // цифра d прибавляется к сумме
 n := n div 10; // отбрасывается последняя цифра числа n
 Until n=0; { Если n=0, цикл прекращает работу, поскольку все циф-
 ры числа рассмотрены }
 Edit2.Text := IntToStr(S);
end;
```

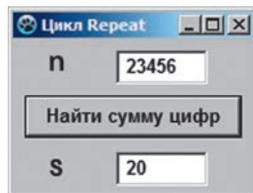


Рис. 40.3

» Посчитать количество слов во введенном пользователем тексте (рис. 40.4).

Будем считать, что слова отделяются исключительно одним пробелом, а первый и последний символы текста (строки  $S$ ) не являются пробелами. Следовательно, количество слов в строке  $S$  на 1 больше, чем количество пробелов. Для ввода текстовой строки используется компонент Мемо  (вкладка Standard на Палитре компонентов): он позволяет вводить многострочный текст с клавиатуры как значение свойства Text.

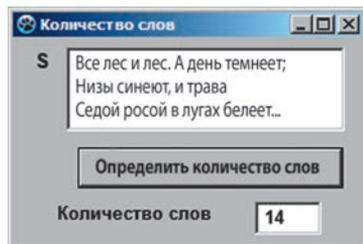


Рис. 40.4

```

var S: String;
 i, count: Integer;
begin
 S := Memo1.Text;
 i := 0;
 If length(S) > 0 Then begin // Если длина строки S не нулевая,
 count := 1; // есть хотя бы одно слово
 Repeat
 i := i + 1; // Проверяется каждый символ строки S
 If S[i] = ' ' // Если найден пропуск,
 Then count := count + 1; // то увеличивается счетчик слов на 1
 Until i >= length(S);
 end
 Else count := 0;
 Edit2.Text := IntToStr(count);
end;
```

### Вопросы для самопроверки



1. Объясните, как выполняется оператор цикла с постусловием.
2. Может ли возникнуть ситуация, при которой тело цикла Repeat не выполнится ни разу?
3. Дан фрагмент программного кода. Дополните таблицу.

| Фрагмент программного кода            | Исходное значение X | Значение X после выполнения цикла | Сколько повторений состоялось |
|---------------------------------------|---------------------|-----------------------------------|-------------------------------|
| Repeat<br>X := X + 1<br>Until X > 10; | 7                   |                                   |                               |
|                                       | 10                  |                                   |                               |
|                                       | 11                  |                                   |                               |

4. Какое значение принимает переменная F после выполнения цикла?  
 i := 1; F := 2;  
 Repeat  
   i := i + 1;    F := F \* i  
 Until i < 6;
5. Объясните назначение оператора цикла:  
 Repeat  
   x := StrToFloat(InputBox('Ввод X', 'Введите положительное число', '0'));  
 Until x > 0;

- Составьте программу для нахождения наименьшего количества первых четных чисел, нужного для того, чтобы получить сумму больше 100.

### Упражнение 40



▶ Организовать ввод последовательности натуральных чисел, которая заканчивается нулем. Найти количество и среднее арифметическое введенных чисел.

- Создайте новый проект.
- Разработайте интерфейс программы согласно рисунку.
- Создайте процедуру обработки события OnClick для кнопки Ввести числа.
- В программном коде процедуры Button1Click запишите операторы:



```
var A, K, S: Integer; Sr: Real;
```

```
begin
```

```
 K := 0; S := 0;
```

```
 Repeat
```

```
 { ввод чисел при помощи функции InputBox }
```

```
 A := StrToInt(InputBox('Ввод чисел', 'A=>', '0'));
```

```
 K := K + 1; // счетчик вводимых чисел
```

```
 S := S + A; // прибавление введенного числа к сумме
```

```
 Until A = 0; // цикл прекращает работу, если A = 0
```

```
 Edit1.Text := IntToStr(K-1); { вывод количества чисел без учета последнего введенного 0 }
```

```
 Sr := S / (K-1); // вычисление среднего арифметического
```

```
 Edit2.Text := FloatToStr(Sr);
```

```
end;
```

- Запустите проект на выполнение. Проверьте работу программы для разных наборов чисел. Сохраните проект в папке Упражнение 40.



- Измените программный код так, чтобы программа определяла порядковый номер  $n$  наименьшего из введенных чисел. Алгоритм нахождения наименьшего (минимального) из введенных чисел и его номера:

- 1) переменной `Min` присвоить значение первого из введенных чисел, а переменной `n` — значение 1;
- 2) каждое следующее число сравнить с `Min`. Если это число меньше `Min` ( $A < \text{Min}$ ), то переменной `Min` присвоить значение данного числа ( $\text{Min} := A$ ), а переменной `n` — его порядковый номер. После ввода  $A = 0$  цикл прекращает работу, `Min` содержит наименьшее число из введенной последовательности, а переменная `n` — порядковый номер этого числа в последовательности.

### Компьютерное тестирование

Выполните тестовое задание 40 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 41. Алгоритмы с повторениями. Цикл с параметром

### Оператор цикла с параметром For

Цикл с параметром (счетчиком) используют в случаях, когда известно количество повторений цикла. Оператор For обеспечивает повторение команд, управляемое параметром цикла.

Синтаксис оператора For:

For <параметр> := NV to KV do <оператор>;

Здесь <параметр> — переменная порядкового типа;

NV и KV — константы, переменные или выражения того же типа, что и параметр;

<оператор> — простой или составной оператор (тело цикла).

Порядок выполнения цикла:

- 1) параметр принимает значение выражения NV;
- 2) пока параметр не превышает значения KV, выполняется <оператор> и значение параметра увеличивается на единицу соответствующего типа.

» Повторить пять раз вывод окна сообщения:

```
For i := 1 to 5 do ShowMessage('i = ' + IntToStr(i));
```

Если конечное значение параметра цикла меньше исходного ( $KV < NV$ ), то цикл не выполнится ни разу.

- » Проанализировать, почему цикл ни разу не выполняется.  
 For A := 10 to 0 do S = S + A;  
 Если при использовании формы to конечное значение параметра цикла меньше исходного, то цикл не выполнится ни разу.

Чтобы реализовать цикл с изменением параметра от большего значения к меньшему, используют форму цикла For со словом downto. Значение параметра с каждой итерацией уменьшается на единицу:

For <параметр> := NV downto KV do <оператор>;

В таком случае параметр каждый раз получает не следующее, а предыдущее значение своего типа.

- » Реализовать обратный отсчет от 10 до 0 в компоненте Label.  
 Чтобы значение счетчика изменялось от большего к меньшему, воспользуемся формой цикла For со словом downto:

```
For A := 10 downto 0 do begin
 Label1.Caption := IntToStr(A);
 Application.ProcessMessages;
 Sleep(1000)
end;
```

- » Найти значение выражения  $A^n$ .  
 В классическом языке Pascal отсутствует стандартная функция возведения в степень. Чтобы вычислить  $St = A^n$ , воспользуемся определением степени:  $St = A \cdot A \cdot \dots \cdot A$ , где количество множителей равно  $n$ .

```
St := 1;
For i := 1 to N do St := St * A;
```

- » Вычислить факториал числа  $n$  (рис. 41.1) по формуле  $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ .

```
var i, n, f: Integer;
begin
 n := StrToInt (Edit1.Text);
 f := 1;
 For i := 2 to n do f := f * i;
 Edit2.Text := IntToStr(f);
end;
```

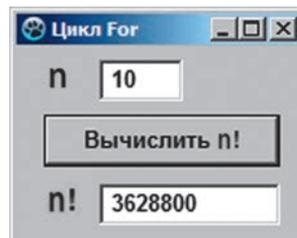


Рис. 41.1

» С клавиатуры вводятся пять действительных чисел, среди которых могут быть как положительные, так и отрицательные числа. Найти среднее арифметическое положительных чисел.

```
var i, k: Integer; a, Sr: Double;
begin
 Sr := 0;
 k := 0; // счетчик положительных чисел
 For i := 1 to 5 do begin
 a := StrToFloat(InputBox('Ввести 5 чисел', IntToStr(i), '0'));
 If a > 0 Then begin
 k := k + 1; Sr := Sr + a;
 end;
 end;
 Sr := Sr / k;
 Edit1.Text := FloatToStr(Sr);
end;
```

## Компонент Список (ListBox)

Для вывода значений переменных в несколько строк удобно использовать компонент ListBox (*Список*).

Компонент ListBox  (вкладка Standard на Палитре компонентов) предназначен для хранения и обработки текстовых данных. Каждая строка ListBox сохраняет строку символов (String).

Доступ к строкам предоставляет свойство Items этого компонента. Для добавления строки в список программным путем предназначен метод Add.

Синтаксис вызова метода Add:  
 ListBox1.Items.Add('Новая\_строка');

» Добавить к списку ListBox значение выражения  $2x$ :  
 ListBox1.Items.Add(IntToStr (2 \* x));

Чтобы очистить содержимое списка, для элемента управления ListBox вызывают метод Clear:

```
ListBox1.Clear;
```

Свойство Count сохраняет количество строк, добавленных в список.

➤ Добавить в список ListBox буквы из заданного диапазона (рис. 41.2):

```
var i, nv, kv: Char;
begin
 nv := Edit1.Text[1];
 kv := Edit2.Text[1];
 For i := nv to kv do ListBox1.Items.Add (i);
end;
```

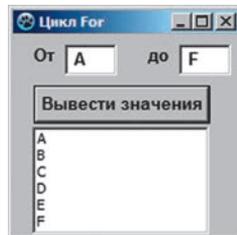


Рис. 41.2

### Вопросы для самопроверки

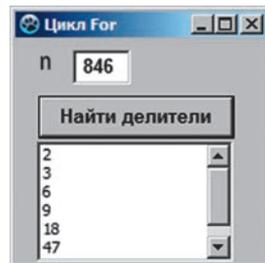


1. Объясните структуру и правила выполнения оператора цикла с параметром. Какого типа может быть параметр цикла?
2. Как изменяется параметр цикла? С каким шагом? В каком диапазоне? Сколько раз выполняется тело цикла?
3. В каком случае применяется форма цикла For со словом downto?
4. Какие преимущества предоставляет использование элемента управления ListBox при выводе значений выражений (переменных)?
5. Чему равно значение переменной S после выполнения цикла:
  - a)  $S := 0$ ; For  $i := 5$  to  $7$  do  $S := S + 1$ ;
  - б)  $S := 0$ ; For  $i := 10$  downto  $6$  do  $S := S + 1$ ;
  - в)  $S := 0$ ; For  $i := 'A'$  to  $'D'$  do  $S := S + 1$ ;
6. Найдите ошибки в данных фрагментах программы:
  - a)  $Y := 0$ ; For  $X := 0.1$  to  $0.9$  do  $Y := Y + \text{Sin}(X)$ ;
  - б)  $K := 81$ ;  $Y := 1$ ; For  $i := 1$  to  $\text{Sqrt}(K)$  do  $Y := Y * 2$ ;
  - в)  $K := 0$ ; For  $i := 9$  to  $1$  do  $K := K * 2$ ;

### Упражнение 41



- Определить все делители числа  $n$  (кроме 1 и самого числа). Если делители отсутствуют, сообщить, что  $n$  — простое число.
- 1) Создайте новый проект. Измените заголовок формы на «Цикл For». Разместите на форме компонент Edit для ввода числа  $n$  согласно рисунку.
  - 2) Добавьте на форму компонент ListBox согласно рисунку.



- 3) Добавьте на форму кнопку Button1. Измените заголовок добавленной кнопки на «Найти делители».
- 4) Проверьте, является ли число  $n$  простым:
  - переберите все числа от 2 до  $(n \div 2)$ . Если число  $i$  является делителем числа  $n$ , то  $i$  добавляется в список ListBox;
  - если после завершения перебора возможных делителей число элементов списка осталось равным 0, то сообщите, что  $n$  — простое число.
- 5) Создайте процедуру обработки события OnClick для кнопки Найти делители. В программном коде данной процедуры запишите операторы, которые реализуют алгоритм, указанный в п. 4:
 

```
var i, n: Integer;
begin
 ListBox1.Clear; // очищение списка
 n := StrToInt(Edit1.Text); // ввод числа n
 For i := 2 to n div 2 do
 If n mod i = 0 Then
 ListBox1.Items.Add(IntToStr(i));
 If ListBox1.Count = 0
 Then ListBox1.Items.Add(IntToStr(n) + ' — простое число');
end;
```
- 6) Запустите проект на выполнение. Проверьте работу программы для разных значений  $n$ . Сохраните проект в папке Упражнение 41.

### Компьютерное тестирование

Выполните тестовое задание 41 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 42. Составление циклических алгоритмов обработки величин



Цикл называется **вложенным**, если он содержится в теле другого цикла.

Такой цикл также называют *внутренним*, а цикл, в котором он содержится, — *внешним*.

Вложенные циклы организованы таким образом: внутренний цикл размещается в теле внешнего цикла (рис. 42.1).

- А — внешний цикл;
- В — внутренний цикл.

Внутренний и внешний циклы могут быть циклами с параметром, с предусловием или с постусловием.

Принцип работы вложенных циклов следующий: при первой итерации внешний цикл вызывает внутренний, который выполняется до своего завершения, после чего управление передается в тело внешнего цикла. При второй итерации внешний цикл снова вызывает внутренний. И так до тех пор, пока не завершится внешний цикл.

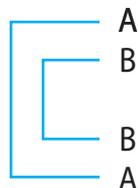


Рис. 42.1

» Напечатать числа в виде таблицы:

```
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
```

Строку из пяти цифр можно сформировать с помощью цикла For: S := ";

```
For i := 1 to 5 do S := S + '5 ' ; // S — строка
```

Чтобы повторить вывод данной строки трижды, вставляем этот цикл внутрь другого:

```
For k := 1 to 3 do // трижды выполняется цикл с параметром k
begin
```

```
 S := ";
```

```
 For i := 1 to 5 do S := S + '5 ' ; // 5 раз выполняется цикл с параметром i
 ListBox1.Items.Add(S);
```

```
end;
```

- ! Параметрами вложенных циклов For должны быть разные переменные.

» Напечатать таблицу умножения (рис. 42.2).

```
var N, K: Integer;
```

```
begin
```

```
 For N := 2 to 9 do begin
```

```
 For K := 1 to 9 do begin
```

```
 ListBox1.Items.Add(IntToStr(N) + ' x ' +
```

```
 IntToStr(K) + ' = ' + IntToStr(N * K));
```

```
 end;
```

```
 ListBox1.Items.Add('-----');
```

```
 end;
```

```
end;
```

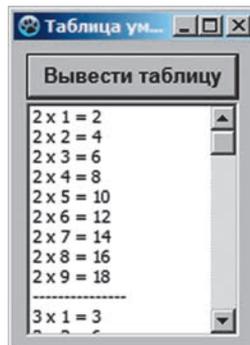


Рис. 42.2

Найти все трехзначные натуральные числа, сумма цифр которых равна их произведению, и определить количество таких чисел (рис. 42.3).

Для решения задачи можно перебрать все возможные сочетания цифр, из которых образуется десятичная запись трехзначного числа, и проверять для каждого сочетания условие задачи.

**var** A, B, C, K: Integer;

**begin**

    K := 0;

**For** A := 1 to 9 do                 // первая цифра

**For** B := 0 to 9 do         // вторая цифра

**For** C := 0 to 9 do     // третья цифра

**If** A + B + C = A \* B \* C **Then**

**begin**

                        K := K + 1;

                        ListBox1.Items.Add(IntToStr(100 \* A + 10 \* B + C));

**end;**

    ListBox1.Items.Add('K=' + IntToStr (K));

**end;**

В задаче использованы вложенные циклы. Обратите внимание на то, что только тело внутреннего цикла с параметром C заключено в операторные скобки **begin...end**, поскольку состоит из двух операторов. Для циклов высших уровней в этом нет потребности, поскольку в каждом из них тело цикла содержит только один оператор: цикл с параметром C является телом цикла с параметром B, а тот, в свою очередь, является телом цикла с параметром A.

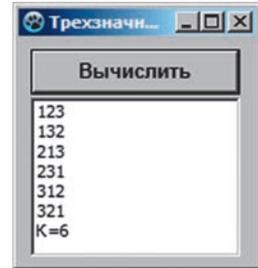


Рис. 42.3

## Вопросы для самопроверки



1. Какие виды циклов вы знаете? В каких случаях применяют каждый из этих циклов?
2. В чем заключается правило вложения циклов?
3. Проанализируйте циклическую конструкцию:

**For** i := 1 to 2 do

**For** j := 1 to 3 do

**For** k := 1 to 3 do

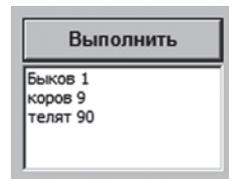
                ListBox1.Items.Add(IntToStr(i) + IntToStr(j) + IntToStr(k));

- а) Назовите тело каждого цикла.  
 б) Сколько раз будет выполнен каждый цикл?  
 в) Какой результат будет выведен после выполнения программы?
4. Какая строка  $S$  будет сформирована в результате выполнения фрагмента программы?
- ```
S := "";
  For i := 0 to 1 do
    For j := 0 to 1 do
      S := S + IntToStr(i) + IntToStr(j);
```
5. Составьте программу для вычисления значения выражения $2k + n$ при всех значениях переменных $n = 1, 2, 3$ и $k = 2, 4, 6, 8$.
6. Составьте программу для вычисления $F = x + x^2 + x^3 + \dots + x^n$ для $x = 0,1; 0,2; \dots; 1,0$ и введенного с клавиатуры значения n .

Упражнение 42



- ▶ Составить программу для решения старинной задачи.
Плата за одного быка — 20 рублей, за корову — 10 рублей, за теленка — 1 рубль. Сколько можно купить быков, коров и телят, если на 200 рублей нужно купить 100 голов скота?
- 1) Создайте новый проект. Измените заголовок формы на «Старинная задача».
 - 2) Разместите на форме компонент `ListBox` согласно рисунку.
 - 3) Добавьте на форму кнопку `Button1`. Измените заголовок добавленной кнопки на «Выполнить».
 - 4) Решение задачи.



Обозначим буквой b количество быков; k — количество коров; t — количество телят. Общее количество голов равно 100: $b + k + t = 100$.

За быков заплатили $20b$ рублей, за коров — $10k$ рублей, за телят — t рублей, то есть $20b + 10k + t = 200$. На 200 рублей можно купить: не более 10 быков, то есть $0 \leq b \leq 10$; не более 20 коров, то есть $0 \leq k \leq 20$; не более 200 телят, то есть $0 \leq t \leq 200$. Таким образом, необходимо перебрать все возможные значения переменных b, k, t и вывести в поле `ListBox` тот набор значений, для которых выполняется условие $(20 * b + 10 * k + t = 200)$ and $(b + k + t = 100)$.

- 5) Создайте процедуру обработки события OnClick для кнопки Выполнить. В программном коде созданной процедуры запишите такие операторы:
- ```
var b, k, t: Integer;
begin
 For b := 0 to 10 do
 For k := 0 to 20 do
 For t := 0 to 200 do
 If (20*b + 10*k + t = 200) and (b + k + t = 100)
 Then begin
 ListBox1.Items.Add('Быков ' + IntToStr(b));
 ListBox1.Items.Add('коров ' + IntToStr(k));
 ListBox1.Items.Add('телят ' + IntToStr(t));
 end;
 end;
 end;
 end;
end;
```
- 6) Запустите проект на выполнение. Проверьте работу программы. Сохраните проект в папке Упражнение 42.

## Компьютерное тестирование



Выполните тестовое задание 42 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## Практическая работа 15



### Составление и выполнение алгоритмов с повторениями и ветвлениями для обработки величин

**Задание:** составить программу для вычисления наибольшего общего делителя (НОД) чисел A и B по алгоритму Евклида.

**Оборудование:** компьютер с установленной средой программирования Lazarus.

### Ход работы

При работе за компьютером соблюдайте правила безопасности.

### I. Размещение элементов управления на форме

- ▶ 1. Создайте новый проект.
- ▶ 2. Разработайте интерфейс программы согласно рис. 1.

### II. Написание процедуры обработки событий

- ▶ 3. Создайте процедуру обработки события OnClick для кнопки Вычислить NOD, с помощью которой вычисляется НОД чисел A и B: пока  $A \neq B$ , из большего числа вычитать меньшее (рис. 2).

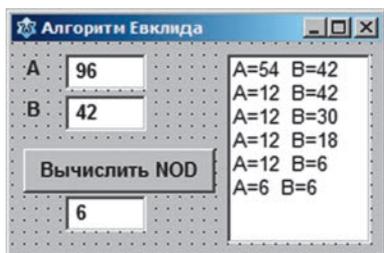


Рис. 1

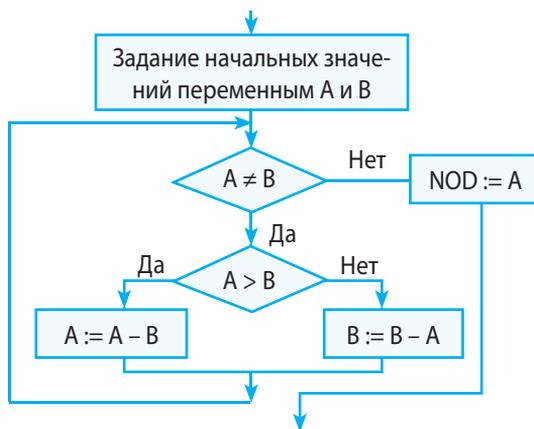


Рис. 2

- ▶ 4. Добавьте на форму текстовое поле Edit4. Измените заголовок кнопки на «Вычислить NOD и NOK».
- ▶ 5. Добавьте в программный код операторы для вычисления значения наименьшего общего кратного (НОК) чисел A и B и вывода данного значения в поле Edit4. Воспользуйтесь формулой:  $NOK(A,B) = A * B / NOD(A, B)$ .

### III. Тестирование проекта

- ▶ 6. Запустите проект на выполнение. Проверьте работу программы для разных наборов чисел. Сохраните проект в папке Практическая работа 15. Завершите работу за компьютером.

**Сделайте вывод:** как составлять и выполнять циклические алгоритмы с ветвлением внутри цикла для решения задач.

# РАЗДЕЛ 9

## ГРАФИЧЕСКОЕ ОТОБРАЖЕНИЕ ДАНЫХ



- § 43. Базовые графические примитивы
  - § 44. Настройка свойств графических примитивов
  - § 45. Создание программ с графическим отображением данных
  - § 46. Графические компоненты
  - § 47. Вывод рисунков из внешних файлов
  - § 48. Разработка программ с графическими компонентами
- Практическая работа 16. Создание программы с графическими компонентами

## ПОВТОРЯЕМ



*Компьютерная графика* — раздел информатики, изучающий технологии обработки графических изображений с помощью компьютерной техники. В зависимости от способа построения графического изображения различают растровые и векторные графические изображения.

*Растровое графическое изображение* состоит из отдельных маленьких прямоугольников — пикселей. Растровый рисунок хранится в памяти компьютера в виде таблицы разноцветных пикселей. Вы уже создавали графические изображения в среде графического редактора Paint.

*Векторное графическое изображение* строится из отдельных базовых объектов — графических примитивов: отрезков, многоугольников, кривых, овалов и т. п. В ходе изучения текстового редактора Word и программы для создания презентаций PowerPoint вы узнали о том, как создавать векторные изображения из геометрических примитивов.

1. Что такое компьютерная графика?
2. Что такое растровое графическое изображение?
3. Что такое векторное графическое изображение?
4. Приведите примеры графических примитивов.
5. Приведите примеры графических редакторов.
6. Опишите принцип построения векторного изображения.



В этом разделе вы ознакомитесь с графической системой координат, узнаете о принципах формирования изображений на экране, научитесь применять графические возможности среды программирования для создания изображений.

## § 43. Базовые графические примитивы

В среде программирования Lazarus существует множество различных средств для работы с компьютерной графикой.

### Графические примитивы

Знакомство с графическими возможностями среды программирования Lazarus начнем с графических примитивов.



**Графические примитивы** — это графические элементы простой формы (линии, окружности, дуги и т. п.), из которых строят более сложные изображения.

Любой векторный рисунок можно рассматривать как совокупность графических примитивов. Чтобы изображение появилось на экране, в тексте программы нужно предусмотреть черчение всех графических примитивов, из которых состоит данное изображение.

### Компонент Изображение (Image)

Для работы с графикой предназначен компонент Image  (*Изображение*), расположенный на вкладке Additional Палитры компонентов. После размещения на форме компонент Image приобретает вид прямоугольной пунктирной рамки (рис. 43.1).

Изменить место расположения на форме (свойства Left, Top), размеры (свойства Width, Height), цвет фона (свойство Color) компонента Image можно в окне Инспектор объектов или запрограммировав изменение значений свойств в ходе выполнения программы.

» Установить в процедуре загрузки формы OnCreate такие значения свойств изображения Image1, чтобы расстояние от него до границ формы со всех сторон составляло 20 пикселей (рис. 43.2).

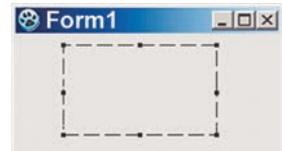


Рис. 43.1

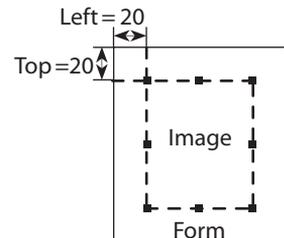


Рис. 43.2

```

procedure TForm1.FormCreate (Sender: TObject);
begin
 Image1.Left := 20;
 Image1.Top := 20;
 Image1.Width := Form1.Width - 40;
 Image1.Height := Form1.Height - 40;
end;

```

## Полотно для рисования

Компонент Image имеет свойство Canvas (*Полотно*). Рисование на полотне осуществляется, когда некоторым его точкам задается цвет. Свойства полотна являются Pen (*Перо*), Brush (*Кисть*) и Font (*Шрифт*). Также свойством полотна является Pixels (*Точки*), которое дает возможность перекрасить каждую точку полотна, то есть каждый пиксель изображения. Координаты конкретной точки указывают в квадратных скобках.

» Вывести точку красного цвета в заданную точку экрана:

```
Image1.Canvas.Pixels [20,50] := clRed;
```

Полотно имеет систему координат *XOY* (рис. 43.3), в которой верхняя левая точка полотна имеет координаты (0,0).

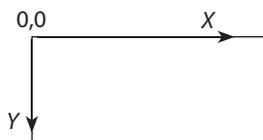


Рис. 43.3

» Закрасить полотно заданным цветом.

```
Image1.Canvas.Brush.Color := clGreen; // устанавливаем зеленый цвет
Image1.Canvas.FillRect (Image1.ClientRect); // закрашиваем
```

Благодаря этим командам полотно становится видимым на экране в виде зеленого прямоугольника.

Цвет можно указать с помощью цветовых констант, которые удобно использовать для работы с основными цветами.

### Цветовые константы

| Значение  | Описание | Значение | Описание     | Значение | Описание    |
|-----------|----------|----------|--------------|----------|-------------|
| clBlack   | Черный   | clGray   | Темно-серый  | clRed    | Красный     |
| clWhite   | Белый    | clLtGray | Светло-серый | clTeal   | Бирюзовый   |
| clBlue    | Синий    | clMaroon | Малиновый    | clPurple | Сиреневый   |
| clGreen   | Зеленый  | clNavy   | Темно-синий  | clSilver | Серебристый |
| clYellow  | Желтый   | clOlive  | Оливковый    | clAqua   | Голубой     |
| clFuchsia | Розовый  | clCream  | Кремовый     | clLime   | Салатный    |

## Графические методы

Полотно предоставляет пространство для создания графических объектов.

Для создания графических примитивов вызывают графические методы полотна Canvas, обеспечивающие рисование определенного графического примитива.

**Отрезок.** Метод `LineTo` чертит отрезок от текущей позиции карандаша до точки с координатами, указанными при вызове метода: `Image1.Canvas.LineTo (250,65);`

Сначала карандаш расположен над точкой  $(0,0)$ . Установить карандаш в другую позицию можно с помощью метода `MoveTo`.

» Нарисовать отрезок черного цвета от точки  $(20,20)$  до точки  $(100,20)$ .  
`Image1.Canvas.Pen.Color := clBlack;`  
`Image1.Canvas.MoveTo (20,20);`  
`Image1.Canvas.LineTo (100,20);`

**Прямоугольник.** Метод `Rectangle` рисует незаполненный прямоугольник, противоположные углы которого совпадают с точками  $(x1,y1)$  и  $(x2,y2)$ .

Стороны прямоугольника параллельны сторонам экрана.

`Image1.Canvas.Rectangle (x1,y1,x2,y2);`

» Нарисовать квадрат с длиной стороны 100 пикселей:  
`Image1.Canvas.Rectangle (20,30,120,130);`

**Эллипс.** Метод `Ellipse` рисует эллипс, вписанный в прямоугольник, который задан координатами двух противоположных вершин  $x1, y1$  и  $x2, y2$  (рис. 43.4).

`Image1.Canvas.Ellipse (x1,y1,x2,y2);`

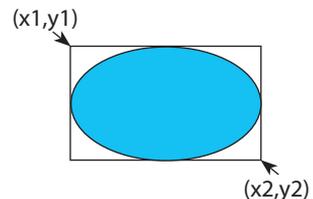


Рис. 43.4

» Нарисовать круг диаметром 100 пикселей.  
`Image1.Canvas.Ellipse (20,30,120,130);`

**Сектор.** Метод `Pie` рисует сектор эллипса.

Эллипс, вписанный в прямоугольник, задан координатами  $(x1,y1,x2,y2)$  (рис. 43.5). Сектор ограничен прямыми, проходящими через центр

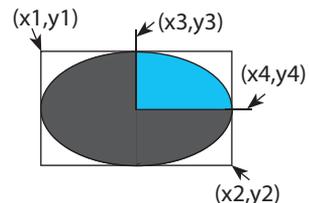


Рис. 43.5

эллипса и точки  $(x_3, y_3)$  и  $(x_4, y_4)$ . Дуга рисуется от точки  $(x_3, y_3)$  до точки  $(x_4, y_4)$  против хода часовой стрелки.

Image1.Canvas.Pie ( $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ )

➤ Нарисовать фигуру по образцу (рис. 43.6).  
 Image1.Canvas.Rectangle (0,0,100,100); // квадрат  
 Image1.Canvas.Pie (0,0,100,100,100,0,0,0); // сектор

**Ломаная линия.** Метод Polyline рисует ломаную линию по точкам ее вершин, которые заданы в виде массива\* (таблицы) Points:

Polyline (Points: array of Tpoint);

➤ Нарисовать ломаную линию по пяти точкам, заданным координатами (рис. 43.7).  
 Image1.Canvas.Polyline ([Point (10,10), Point (30,100), Point (50,10), Point (70,100), Point (90,10)]);

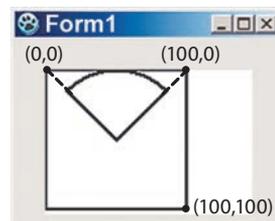


Рис. 43.6

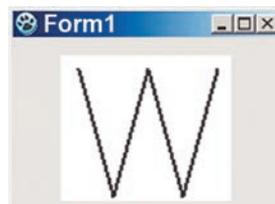


Рис. 43.7

Мы рассмотрели только часть методов для построения графических примитивов, но их достаточно для того, чтобы создавать сложные рисунки. Помните, что их можно применять для всех компонентов, имеющих свойство Canvas.

### Вопросы для самопроверки



1. Что такое графические примитивы?
2. Для чего предназначено полотно Canvas?
3. Какими свойствами обладает полотно Canvas?
4. Какие графические методы применяются к свойству Canvas?
5. Как начертить отрезок на полотне компонента Image?
6. Как нарисовать прямоугольник; эллипс; ломаную линию на полотне Canvas?

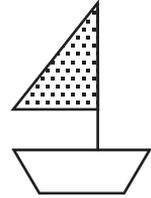
### Упражнение 43



- Составить программу для рисования графических примитивов.
- 1) Создайте новый проект. Разместите на форме четыре командные кнопки, по щелчку которых будет выведен соответствующий графический примитив.

\* Детально с массивами вы ознакомитесь в 9 классе.

- 2) Постройте треугольник с вершинами в точках (100, 100), (200, 200), (100, 200).
- 3) Постройте прямоугольник с вершинами (200, 10), (320, 10), (320, 120), (200, 120).
- 4) Постройте круг радиусом 50 пикселей, центр которого совпадает с центром полотна Image1.
- 5) Запустите проект на выполнение. Проверьте работу программы. Сохраните проект в папке Упражнение 43.
- 6) Добавьте новую кнопку для рисования лодочки (см. рисунок).



### Компьютерное тестирование

Выполните тестовое задание 43 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 44. Настройка свойств графических примитивов

Для рисования рисунков и ввода текста применяют инструменты Pen, Brush и Font. Перо используется для рисования линий, кисть — для закрашивания фигур, а шрифт — для выбора шрифта текста.

### Свойство Перо (Pen)

**Карандаш** (Canvas.Pen) используют для рисования фигур, состоящих из линий. Рассмотрим некоторые свойства пера.

Pen.Color устанавливает цвет пера, то есть цвет, которым будет нарисован контур фигуры. Свойство Color может принимать значения цветовых констант. Также цвет рисунка можно задавать с помощью функции RGBToColor, аргумент которой задает степень насыщенности красного, зеленого и синего цветов:

```
Pen.Color := RGBToColor (r, g, b);
```

Переменные *r*, *g*, *b* могут принимать значения от 0 до 255. Так можно получить более 16 млн различных цветов:

| Цвет    | R(ed) | G(reen) | B(lue) |
|---------|-------|---------|--------|
| Черный  | 0     | 0       | 0      |
| Белый   | 255   | 255     | 255    |
| Красный | 255   | 0       | 0      |
| Зеленый | 0     | 255     | 0      |
| Синий   | 0     | 0       | 255    |

Pen.Style определяет стиль линии и может принимать, в частности, следующие значения:

| Значение | Вид  | Значение | Вид       |
|----------|------|----------|-----------|
| psSolid  | ———— | psDash   | - - - - - |

Pen.Width определяет толщину линии. Значение этого свойства может быть целым числом. Линия толщиной больше 1 может быть только сплошной.

» Нарисовать на форме пирамиду по образцу (рис. 44.1).

```
Form1.Canvas.Pen.Style := psSolid;
Form1.Canvas.Pen.Width := 3;
Form1.Canvas.Polyline ([Point (150,50),
 Point (50,250), Point (250,250),
 Point (150,50), Point (270,170),
 Point (250,250), Point (150,50)]);
Form1.Canvas.Pen.Width := 1;
Form1.Canvas.Pen.Style := psDash;
Form1.Canvas.MoveTo (150,50);
Form1.Canvas.LineTo (130,170);
Form1.Canvas.LineTo (270,170);
Form1.Canvas.MoveTo (130,170);
Form1.Canvas.LineTo (50,250);
```

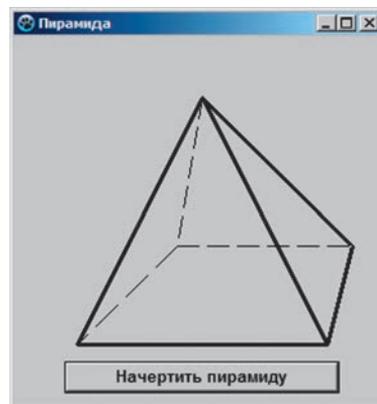


Рис. 44.1

## Свойство Кисть (Brush)

**Кисть** (Canvas.Brush) используют для закрашивания (заливки) замкнутых геометрических фигур. Свойства кисти:

- Brush.Color устанавливает цвет, которым можно закрашивать фигуры;
- Brush.Style определяет стиль заполнения фигуры, который, в частности, может принимать следующие значения:
  - bsSolid — сплошная заливка цветом;
  - bsDiagonal — штриховка справа налево;
  - bsHorizontal — горизонтальные линии и др.

## Вывод текста на полотно

**Шрифт** (Canvas.Font) определяет шрифт, которым будет выведен текст. Можно установить следующие свойства шрифта: размер, начертание и т. п.

» Изменить значение свойств шрифта.

```
Canvas.Font.Size := 20; //размер в пунктах (20)
Canvas.Font.Color := clRed; //цвет (красный)
Canvas.Font.Name := 'Comic Sans MS'; //гарнитура (Comic Sans MS)
Canvas.Font.Style := [fsBold, fsItalic]; // начертание (полужирное, курсив)
или
Canvas.Font.Style := []; //начертание (обычное)
Canvas.Font.Style := [fsBold]; //начертание (полужирное)
```

**Вывод текста на полотно** может осуществляться с помощью метода TextOut (x, y: Integer; const Text: String);

Этот метод выводит строку текста Text на полотно, начиная с позиции с координатами (x, y).

```
Canvas.TextOut (10, 10, 'Вывод текста');
```

» Залить прямоугольники зеленым цветом с использованием разных стилей кисти по образцу (рис. 44.2).

Чтобы рядом с каждым вызовом метода не писать название объекта Canvas, используем оператор присоединения With. При этом в теле оператора к каждой строке, начало которой совпадает с названием свойства или метода объекта Canvas, при компиляции будет добавлено Canvas.

```
With Canvas do begin
```

```
 Font.Size := 16;
 TextOut (10, 10, 'bsSolid'); // вывод названия стиля кисти
 Brush.Color := clGreen; // цвет заливки
 Brush.Style := bsSolid; // стиль заливки
```

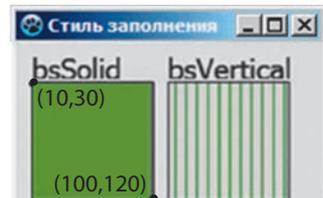


Рис. 44.2

```

Rectangle (10, 30, 100, 120);
Brush.Style := bsClear;
TextOut (110, 10, 'bsVertical');
Brush.Color := clGreen;
Brush.Style := bsVertical; // стиль заливки
Rectangle (110, 30, 200, 120);
end;
```

## Заполнение многоугольников и сложных фигур

При рисовании простых геометрических фигур можно автоматически их закрашивать согласно настройкам кисти. Для любой замкнутой области используют специальный метод FloodFill:

```
Canvas.FloodFill (x, y, Color, FillStyle);
```

Здесь  $x$ ,  $y$  — это координаты любой точки внутри фигуры.

Если  $FillStyle = fsBorder$ , то краска «растекается» от точки с координатами  $(x, y)$  до тех пор, пока не встретит на пути цвет, заданный в параметре  $Color$ . Если  $FillStyle = fsSurface$ , то цвет  $Color$  заменяется цветом, указанным в свойстве  $Brush.Color$ .

➤ Нарисовать зеленый треугольник со сторонами красного цвета (рис 44.3).

```

Form1.Canvas.Pen.Color := clRed;
Form1.Canvas.Polyline ([Point (150,50),
Point (50,250), Point (250,250), Point (150,50)]);
Form1.Canvas.Brush.Color := clGreen;
Form1.Canvas.FloodFill (200, 200, clRed, fsBorder);
```

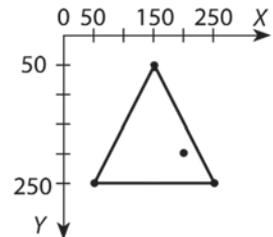


Рис. 44.3

Описанные возможности Lazarus значительно расширяют ваш арсенал художника-программиста. Поэтому обязательно проверьте на практике все приведенные примеры, а потом попробуйте запрограммировать построение собственного рисунка.

## Вопросы для самопроверки



1. Какими свойствами обладает инструмент Pen?
2. Как задать цвет для построения контура графических примитивов?
3. Какими свойствами обладает инструмент Brush?
4. Как закрасить фигуру?
5. Как построить и закрасить зеленым цветом круг радиусом 100, центр которого совпадает с центром формы?

6. Запишите фрагмент программы для построения треугольника с вершинами в точках (100,100), (150,100), (80,70); цвет фона — серый; цвет линий — красный.

### Упражнение 44



- ▶▶ Составить программу для рисования елки по образцу (рис. 1).

- 1) Создайте новый проект. Добавьте на форму кнопку Button1, по щелчку которой будет выполняться рисование на полотне формы.
- 2) Создайте процедуру обработки события OnClick для формы.
- 3) Запишите программный код для рисования елки:



Рис. 1

```
With Form1.Canvas do // работаем с объектом Form1.Canvas
begin
 Pen.Width := 1; // устанавливаем толщину пера
 Pen.Color := clGreen; // устанавливаем цвет пера
 Brush.Color := clGreen; // устанавливаем цвет заливки
 { Рисование елки }
 Polygon ([Point (40,180), Point (180,140), Point (60,140), Point (90,100),
 Point (80,100), Point (100,50), Point (120,100), Point (110,100), Point
 (140,140), Point (120,140), Point (160,180)]);
 Pen.Color := RGBToColor (128,64,0); // устанавливаем цвет пера
 Brush.Color := RGBToColor (128,64,0); // устанавливаем цвет заливки
 { Рисование ствола }
 Rectangle (80,180,120,200);
end;
```

- 4) Запустите проект на выполнение. Добавьте операторы рисования желтого солнышка (рис. 2).
- 5) Проверьте работу программы. Сохраните проект в папке Упражнение 44.
- 6) Наша елка «висит» в воздухе. Добавьте операторы для рисования лесной лужайки, на которой растет елка, и облаков над елкой.



Рис. 2



### Компьютерное тестирование

Выполните тестовое задание 44 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 45. Создание программ с графическим отображением данных

В предыдущем параграфе вы рисовали графические примитивы на форме в результате выполнения процедуры обработчика события OnClick для кнопки. При перекрывании окна формы другими окнами или изменении размеров формы часть изображения может быть потеряна. Существуют способы для восстановления изображения при активизации формы и настройки размеров изображения под текущие размеры формы.

### Перерисовывание формы

Событие OnPaint происходит при необходимости перерисовать форму, например, в случае ее активации, если до этого часть формы была закрыта другими окнами.

Для создания процедуры обработки события OnPaint нужно:

- 1) щелкнуть мышью на форме;
- 2) в окне Инспектор объектов перейти на вкладку События;
- 3) дважды щелкнуть в поле события OnPaint (рис. 45.1).

Созданная процедура TForm1.FormPaint будет выполняться во время загрузки формы и при каждом ее перерисовывании в случае изменения размеров или отображения формы на экране.

```
procedure TForm1.FormPaint (Sender: TObject);
begin
```

```
 Form1.Canvas. Rectangle (200,0,300,100);
 { далее следуют команды для построения рисунка }
```

```
end;
```

Для визуализации процесса изменения размера формы нужно очистить полотно перед перерисовыванием формы нового размера. Для этого в процедуре обработки события OnResize нужно вызвать метод Refresh (). Он очищает полотно и генерирует событие OnPaint:

```
procedure TForm1.FormResize (Sender: TObject);
begin
 Refresh ();
end;
```



Рис. 45.1

» Нарисовать разноцветные линии (отрезки) случайным образом (рис. 45.2).

```
procedure TForm1.FormPaint (Sender: TObject);
const Num_Lines = 100; // количество линий
var
i: Integer;
begin
 Randomize; { инициализация генератора
случайных чисел }
 For i := 0 to Num_Lines do
 begin
 Canvas.Pen.Color := RGBToColor(Random (256), Random (256), Random
(256)); //случайный выбор цвета
 Canvas.LineTo (Random (ClientWidth), Random (ClientHeight)); { рисо-
вание отрезка со случайными координатами концов }
 end;
 end;
procedure TForm1.FormResize (Sender: TObject);
begin
 Refresh ();
end;
```



Рис. 45.2

## Масштабирование изображения

При изменении размеров формы в режиме выполнения желательно, чтобы размеры нашего рисунка подстраивались под эти изменения. Свойства формы ClientWidth и ClientHeight определяют ширину и высоту области, на которой можно рисовать.

Чтобы **рисунок масштабировался вместе с формой**, необходимо при вызове графических методов указывать не абсолютные значения координат, а зависимости их координат от свойств ClientWidth, ClientHeight.

» Масштабировать рисунок по образцу (рис. 45.3, 45.4):

```
procedure TForm1.FormPaint (Sender: TObject);
const x = 10; y = 10;
var dx, dy: Integer;
begin
 dx := Form1.ClientWidth div x;
 dy := Form1.ClientHeight div y;
 with Form1.Canvas do begin
 Font.Size := 20;
```



Рис. 45.3

```

TextOut (0, dy, ' Hello!');
Pen.Color := clYellow;
Pen.Width := 3;
Brush.Color := clYellow;
Ellipse (dx, 2*dy, 3*dx, 5*dy);
MoveTo (dx, 2*dy);
LineTo (3*dx, 5*dy);
MoveTo (3*dx, dy);
LineTo (dx, 5*dy);
MoveTo (0, 4*dy);
LineTo (4*dx, 3*dy);
Brush.Style := bsClear;
end;
end;
procedure TForm1.FormResize (Sender: TObject);
begin
 Refresh ();
end;

```

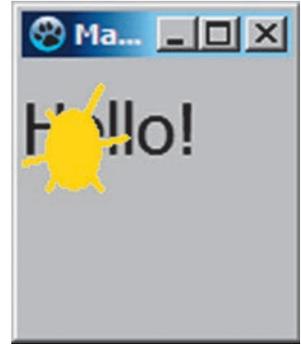


Рис. 45.4

### Вопросы для самопроверки



1. Как создать процедуру обработки события OnPaint для формы?
2. Какое действие выполняет метод Refresh()?
3. Какими свойствами определяются размеры формы?
4. Как запрограммировать масштабирование изображения?
5. Как запрограммировать задание цвета рисования случайным образом?
6. Напишите фрагмент программы для заполнения формы разноцветными точками со случайными координатами.

### Упражнение 45



► Составить программу для рисования в центре формы четырех концентрических эллипсов, покрашенных разными цветами (рис. 1).

- 1) Создайте новый проект. Измените заголовок формы на «Эллипсы».
- 2) Создайте процедуру обработки события OnPaint для формы.
- 3) В программном коде процедуры запишите операторы для рисования четырех эллипсов (рис. 2).



Рис. 1

```

procedure TForm1.FormPaint (Sender: TObject);
const x = 10; y = 10; // условное разбиение формы на ячейки
var dx, dy, i: Integer;
begin
 dx := Form1.ClientWidth div x;
 dy := Form1.ClientHeight div y;
 For i := 1 to 4 do begin
 Canvas.Brush.Color := RGBToColor
 (255, 255 div i, 255 div i);
 Canvas.Ellipse (i*dx, i*dy, (10-i) *dx,
 (10-i) *dy);
 end;
end;
procedure TForm1.FormResize (Sender: TObject);
begin
 Refresh ();
end;

```

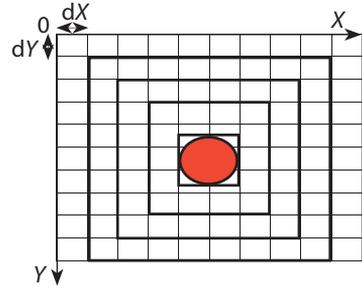


Рис. 2

- 4) Измените цвета заливки эллипсов на оттенки зеленого.
- 5) Запустите проект на выполнение. Проверьте работу программы. Сохраните проект в папке Упражнение 45.
- 6) Измените программный код таким образом, чтобы выполнялось рисование пяти вложенных прямоугольников.



### Компьютерное тестирование

Выполните тестовое задание 45 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 46. Графические компоненты

При конструировании формы для создания визуальных эффектов и изображений используют такие графические компоненты, как изображение Image и автофигура Shape.

### Автофигуры (Shape)

Компонент Shape  (Автофигуры) расположен на вкладке Additional Палитры компонентов и предназначен для вывода геометрических фигур.

Вид фигуры, выводимой этим компонентом, определяется свойством `Shape`. Форму фигуры можно выбрать в списке свойства `Shape` (рис. 46.1) в окне Инспектор объектов (рис. 46.2).

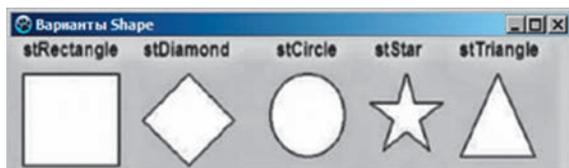


Рис. 46.1

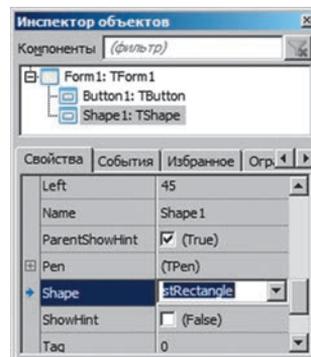


Рис. 46.2

Чтобы задать размеры фигуры, в окне Инспектор объектов устанавливают значения свойств `Height` и `Width`. Для точного размещения фигуры на форме предназначены свойства `Left` и `Top`.

Вы уже знакомы со свойствами `Pen` и `Brush` полотна `Canvas` компонента `Image`. Аналогичными свойствами обладает и компонент `Shape`.

Свойство `Pen` определяет цвет и стиль контура фигуры, свойство `Brush` — стиль и цвет закрашенной фигуры. Значения этих свойств можно задать как в окне Инспектор объектов, так и программным способом.

➤ Изменить значения свойств компонента `Shape` программным способом по образцу (рис. 46.3).

```
With Shape1 do begin
 Shape := stEllipse;
 Pen.Color := clBlue;
 Pen.Width := 3;
 Brush.Color := clRed;
 Brush.Style := bsDiagCross;
end;
```

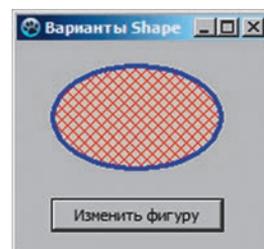


Рис. 46.3

## Перемещение объектов мышью

Рассмотрим алгоритм перемещения объекта с помощью мыши. Для перемещения объекта нужно:

- 1) нажать левую кнопку мыши на объекте (событие `MouseDown`);
- 2) переместить указатель при нажатой кнопке мыши (событие `MouseMove`);
- 3) отпустить кнопку мыши (событие `MouseUp`).

Чтобы запрограммировать перемещение по форме объекта `Shape1`, нужно создать для него процедуры обработки событий: `OnMouseDown` (нажатие кнопки мыши, когда указатель мыши находится в области объекта), `OnMouseMove` (перемещение мыши), `OnMouseUp` (отпускание кнопки мыши). Текст программы будет следующим:

```

var
 Form1: TForm1;
 x1, y1: Integer; // описание глобальных переменных
 flag: Boolean = False; // кнопка мыши не нажата
implementation
 {$R * .lfm}
 procedure TForm1.Shape1MouseDown (Sender: TObject; Button:
 TMouseButton; Shift: TShiftState; x, y: Integer);
 begin
 x1 := x; y1 := y; // запоминание текущих координат указателя мыши
 flag := True; // кнопка мыши нажата
 end;
 procedure TForm1.Shape1MouseMove (Sender: TObject; Shift: TShiftState;
 x, y: Integer);
 begin
 If flag Then // если кнопка мыши нажата
 begin
 Shape1.Left := Shape1.Left + x - x1;
 Shape1.Top := Shape1.Top + y - y1;
 end;
 end;
 procedure TForm1.Shape1MouseUp (Sender: TObject; Button: TMouseButton;
 Shift: TShiftState; x, y: Integer);
 begin
 flag := False; // кнопка мыши не нажата
 end;

```

Рассмотрим работу процедур данного проекта. Если кнопка мыши нажата в момент, когда указатель находится в области объекта `Shape1`, то в процедуре `Shape1MouseDown` переменным `x1`, `y1` будут присвоены значения координат указателя, а переменной `flag` — значение `True`. Если не отпускать кнопку мыши и перетягивать объект, то в процедуре `Shape1MouseMove` происходит пересчет координат объекта `Shape1` относительно системы координат формы. Если кнопку мыши отпустить, то выполнится процедура `Shape1MouseUp`, в которой переменная `flag` примет значение `False`, и перетягивание завершится.

- Разработать игру «Конструктор», создав процедуры обработки событий OnMouseDown, OnMouseMove, OnMouseUp для компонентов (рис. 46.4).



Рис. 46.4

### Вопросы для самопроверки



1. Для чего предназначен компонент Shape?
2. Какое свойство компонента Shape определяет форму автофигуры?
3. Как изменить размеры; цвет заливки; контур автофигуры?
4. Какие события происходят при нажатии; перемещении; отпущении мыши?
5. Как запрограммировать перемещение объекта с помощью мыши?

### Упражнение 46



- Составить программу, реализующую отображение на форме пяти разных геометрических фигур (рис. 1).
- 1) Создайте новый проект. Измените заголовок формы на «Стили заполнения Shape». Разместите на форме пять компонентов Shape.
  - 2) Придайте фигурам вид: прямоугольника (stRectangle); круга (stCircle); эллипса (stEllipse); квадрата (stSquare); прямоугольника со скругленными углами (stRoundRect).
  - 3) Измените контур фигур, используя свойства Pen.Color, Pen.Width, Pen.Style.
  - 4) Залейте фигуры нужными цветом и стилем, используя свойства Brush.Color, Brush.Style.
  - 5) Проверьте работу программы. Сохраните проект в папке Упражнение 46.



Рис. 1



- 6) Создайте процедуры обработки событий OnMouseDown, OnMouseMove, OnMouseUp для любой автофигуры.
- 7) С помощью компонентов Share создайте на форме флаги Украины, Франции, Польши (рис. 2).



Рис. 2

## Компьютерное тестирование

Выполните тестовое задание 46 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 47. Вывод рисунков из внешних файлов

Для создания проекта можно использовать изображения, содержащиеся во внешних графических файлах. Изображение, содержащееся в файле, добавляется на форму с помощью компонента Image.

### Свойства компонента Image

| Свойство      | Что определяет                                                                                                                                                        |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Picture       | Изображение, которое выводится из внешнего файла в область компонента                                                                                                 |
| Width, Height | Ширина и длина компонента. Если размеры компонента меньше, чем размеры изображения, и значение свойств Autosize и Stretch равно False, то выводится часть изображения |
| AutoSize      | Признак автоматического изменения размера компонента в соответствии с размером изображения (имеет тип Boolean)                                                        |
| Stretch       | Признак автоматического масштабирования изображения в соответствии с размером компонента при условии, что Autosize = False (имеет тип Boolean)                        |
| Visible       | Признак отображения компонента, и, соответственно, изображения, на форме (имеет тип Boolean)                                                                          |

Для вывода изображения в область компонента Image нужно:

- 1) в окне Инспектор объектов выбрать свойство Picture и щелкнуть кнопку ;
- 2) щелкнуть кнопку Загрузить окна Диалог загрузки изображения;
- 3) выбрать файл в диалоговом окне выбора файла.



Рис. 47.1

Если размеры изображения больше размеров компонента, то свойству `Stretch` нужно присвоить значение `True`. Размеры рисунка изменятся в соответствии с реальными размерами компонента `Image` (рис. 47.2).



Рис. 47.2

## Загрузка изображения в ходе выполнения проекта

Чтобы вывести изображение в область компонента `Image` во время работы проекта, нужно к свойству `Picture` применить метод `LoadFromFile`, указав в качестве параметра имя файла с необходимым изображением.

» `Image1.Picture.LoadFromFile ('c:\Photo\portret.bmp')` — загружает изображение из графического файла `portret.bmp`, содержащегося в каталоге `Photo` корневого каталога диска `C:`, и выводит его в область объекта `Image1`.

» Если графический файл содержится в текущем каталоге (в папке проекта), достаточно указать только его название.  
`Image1.Picture.LoadFromFile ('Parrot.jpg');`

Метод `LoadFromFile` позволяет выводить изображения разных графических форматов, но для загрузки `jpg`-файлов необходимо включить в блок `uses` стандартный модуль `JPEG`:

```
uses JPEG;
```

Очистить содержимое объекта `Image1` можно следующим способом:  
`Image1.Picture := nil;`

## Пиктографическая кнопка (BitBtn)

Компонент `BitBtn`  (*Пиктографическая кнопка*) расположен на вкладке `Additional` Палитры компонентов.

Компонент `BitBtn` предоставляет возможность создавать кнопку, которая на своей поверхности может содержать не только надпись, но и изображение, задаваемое свойством `Glyph`. При щелчке кнопки в строке `Glyph` в окне Инспектор объектов открывается окно `Picture Editor`, как и при задании свойства `Picture` для компонента `Image`.

Возможности масштабирования рисунка для компонента `BitBtn` нет, поэтому нужно его размеры согласовывать с размерами созданной кнопки.

Свойство Kind компонента BitBtn дает возможность выбрать вид кнопки из стандартного набора вариантов.

Щелчок любой пиктографической кнопки возвращает в программу результат (название этой кнопки, например Yes, No, Cancel), который определяет дальнейшее выполнение программы. Отличие имеет кнопка Close: ее щелчок приводит к завершению работы программы.

## Выбор цвета (ColorDialog)

Компонент ColorDialog



(вкладка Dialogs на Палитре компонентов) позволяет во время работы программы вызвать диалоговое окно выбора цвета (рис. 47.3).

В окне можно выбирать цвет из базовой палитры или, щелкнув кнопку Определить цвет, раскрыть дополнительную панель (см. рис. 47.3). Это позволяет выбрать цвет, отличный от базовых.



Рис. 47.3

Выбор в программе цвета по расширенной панели осуществляется с помощью метода Execute. Этот метод срабатывает в момент щелчка кнопки ОК в окне выбора цвета.

» Задать цвет фона автофигуры Shape1 с помощью оператора:  
`If ColorDialog1.Execute Then Shape1.Color := ColorDialog1.Color;`

## Вопросы для самопроверки



1. Как добавить на форму изображение из внешнего файла?
2. Опишите основные свойства компонента Image.
3. Почему для свойства Stretch компонента Image следует устанавливать значение True?
4. Запишите оператор, реализующий добавление на форму изображения из файла P1.bmp, который содержится на диске C: в папке Photo.
5. Чем отличаются компоненты Button и BitBtn?
6. Запишите оператор, который задает цвету пера значение, выбранное в диалоговом окне выбора цвета.

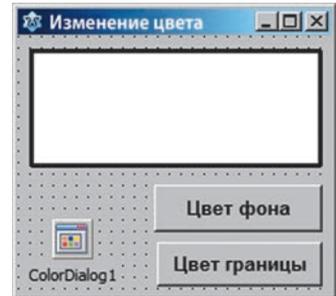
## Упражнение 47



▶ Составить программу, в которой автофигуре задается цвет с помощью компонента ColorDialog.

- 1) Создайте новый проект. Измените заголовков формы на «Изменение цвета».
- 2) Разместите на форме компоненты Shape и ColorDialog согласно рисунку. В окне Инспектор объектов измените значение толщины границы прямоугольника (объекта Shape1). Для этого в списке параметров свойства Pen установите для параметра Width значение 3 пикселя.
- 3) Разместите на форме два компонента Button и измените заголовки созданных кнопок на «Цвет фона» и «Цвет границы».
- 4) Создайте процедуру обработки события OnClick для кнопки Цвет границы. В программном коде созданной процедуры запишите операторы для отображения контура автофигуры Shape1 цветом, выбранным в диалоговом окне выбора цвета:
 

```
procedure TForm1.Button2Click (Sender: TObject);
begin
 If ColorDialog1.Execute Then Shape1.Pen.Color := ColorDialog1.Color;
end;
```
- 5) Создайте процедуру обработки события OnClick для кнопки Цвет фона. Добавьте операторы для изменения цвета заливки автофигуры Shape1 на цвет, выбранный в диалоговом окне выбора цвета. Запустите проект на выполнение. Сохраните проект в папке Упражнение 47.
- 6) Добавьте в процедуру Button1Click оператор для изменения стиля заливки автофигуры. Добавьте в процедуру Button2Click оператор для увеличения ширины границы прямоугольника.



## Компьютерное тестирование

Выполните тестовое задание 47 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 48. Разработка программ с графическими компонентами

Из предыдущего параграфа вы узнали, как в проекте загрузить изображение из графического файла. Также в проекте можно открывать и сохранять файлы.

### Работа с файлами изображений

Компоненты `OpenPictureDialog`  и `SavePictureDialog`  (вкладка `Dialogs` на Палитре компонентов) позволяют во время работы программы выбирать в диалоговом окне графический файл для его открытия или указывать название файла для сохранения изображения.

В диалоговом окне компонента `OpenPictureDialog` есть область предварительного просмотра файлов. Значение свойства `Filter` установлено для показа графических файлов с расширениями `.bmp`, `.ico`, `.emf`, `.wmf`. Чтобы в фильтре была возможность выбора файлов с расширением `.jpg`, нужно добавить модуль `JPEG` в блок `uses`.

При выполнении проекта реализация диалога с пользователем для открытия или сохранения файла осуществляется с помощью метода `Execute`. Метод срабатывает по щелчку кнопки `Открыть` или `Сохранить` в окне выбора файла. Если пользователь щелкнет кнопку `Открыть` (`Сохранить`), диалог завершится, метод `Execute` вернет значение `True` и название выбранного файла сохранится в свойстве `FileName` компонента `OpenPictureDialog` (`SavePictureDialog`). Если же пользователь откажется от диалога (щелкнет кнопку `Отменить` или нажмет клавишу `Esc`), то метод `Execute` вернет значение `False`.

В программе нужно предусмотреть соответствующий компонент, с помощью которого начнется диалог пользователя с программой об открытии (сохранении) файла.

Для открытия файла в процедуре обработки события для соответствующего компонента (например, кнопки) записывают оператор:

```
If OpenPictureDialog1.Execute Then
 Image1.Picture.LoadFromFile (OpenPictureDialog1.FileName);
```

Для сохранения изображения в файле в процедуре обработки события для соответствующего компонента нужно записать оператор:

```
If SavePictureDialog1.Execute Then
 Image1.Picture.SaveToFile (SavePictureDialog1.FileName);
```

## Главное меню (MainMenu)

Компонент MainMenu  (вкладка Standard на Палитре компонентов) предназначен для добавления в программу главного меню. Редактор меню вызывается двойным щелчком компонента MainMenu.

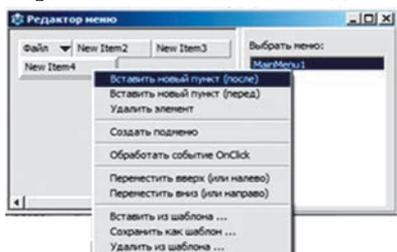


Рис. 48.1

В окне Редактор меню автоматически создается пункт меню верхнего уровня New Item1. Для добавления пунктов меню и подменю удобно пользоваться командой контекстного меню (рис. 48.1). Меню состоит из отдельных пунктов, значение свойства Caption каждого из которых нужно задать в окне Инспектор объектов.

Каждая команда меню — это объект со своими свойствами и методами. Чтобы меню выполняло свои функции, нужно создать процедуру обработки события OnClick для каждого пункта меню.

### Вопросы для самопроверки



1. Какой компонент позволяет открывать графический файл с помощью диалогового окна в ходе выполнения проекта?
2. Какой компонент позволяет сохранять графический объект с помощью диалогового окна в ходе выполнения проекта?
3. Объясните назначение метода Execute.
4. Объясните назначение компонента MainMenu.
5. Как добавлять пункты в главное меню?
6. Как создать процедуру обработки события для пунктов меню?

### Упражнение 48



- ▶ Составить программу, в которой реализованы основные функции графического редактора.
- 1) Создайте новый проект. Измените заголовок формы на «Графический редактор». Разместите на форме компоненты согласно рис. 1: Image; OpenPictureDialog; SavePictureDialog; MainMenu; ColorDialog; три компонента BitBtn.  
Пункты меню назовите так, как показано на рис. 1–3.
  - 2) Создайте процедуру обработки события OnCreate для формы. В программном коде процедуры запишите операторы:

```

procedure TForm1.FormCreate (Sender: TObject);
begin
 Image1.Canvas.Brush.Color := clWhite;
 Image1.Canvas.FillRect (ClientRect);
end;

```

- 3) В редакторе меню создайте пункты меню **Файл** и **Рисунок** и соответствующие пункты подменю по образцу (рис. 2, 3).

- 4) Создайте процедуры обработки событий для пунктов **Открыть** и **Сохранить** меню **Файл**.

```

procedure TForm1.MenuItem2Click (Sender: TObject);
begin
 If OpenPictureDialog1.Execute Then
 Image1.Picture.LoadFromFile
 (OpenPictureDialog1.FileName);
end;

```

```

procedure TForm1.MenuItem3Click (Sender: TObject);
begin
 If SavePictureDialog1.Execute Then
 Image1.Picture.SaveToFile (SavePictureDialog1.
 FileName);
end;

```

- 5) Самостоятельно создайте процедуры обработки событий для пункта **Выход** меню **Файл** и пункта **Очистить** меню **Рисунок**.

- 6) Добавьте на пиктографические кнопки **BitBtn1**, **BitBtn2**, отвечающие за выбор инструмента для рисования, и на кнопку **BitBtn3**, отвечающую за выбор цвета, соответствующие изображения: карандаш (перо), заливка (кисть), цветовая палитра.

- 7) В блок описания глобальных переменных добавьте описание переменных, отвечающих за выбор инструментов для рисования:

```

var Form1: TForm1;
 pen1, brush1: Boolean; cl: TColor;
 flag: Boolean; // вспомогательная переменная

```

- 8) Создайте процедуры обработки событий для кнопок **BitBtn1**, **BitBtn2**.

```

procedure TForm1.BitBtn1Click (Sender: TObject);
begin
 pen1 := True; brush1 := False;
end;

```



Рис. 1



Рис. 2



Рис. 3

```

procedure TForm1.BitBtn2Click (Sender: TObject);
begin
 pen1 := False; brush1 := True;
end;

```

- 9) Создайте процедуру обработки событий для кнопки BitBtn3, реализующую выбор цвета с помощью компонента ColorDialog.
- 10) Запрограммируйте возможность рисования с помощью мыши (рис. 4). По щелчку мышью в области компонента Image1 в процедуре Image1MouseDown проверяется, какой инструмент выбран. Если выбран инструмент Карандаш (pen1 = True), то переменной flag задается значение True, рисование линии начинается с текущей координаты (x, y) указателя. Если выбран инструмент Заливка (brush1 = True), то происходит заливка от точки (x, y) выбранным цветом (FillStyle = fsSurface).

Если не отпускать кнопку мыши и передвигать указатель по поверхности Image1, то в процедуре TForm1.Image1MouseMove происходит рисование линии по текущим координатам указателя мыши. Если кнопку мыши отпустить, то выполнится процедура TForm1.Image1MouseUp, в которой flag примет значение False, и рисование линии завершится.

```

procedure TForm1.Image1MouseDown (Sender: TObject; Button:
TMouseButton; Shift: TShiftState; x, y: Integer);
begin

```

```

 if pen1 Then
 begin
 Image1.Canvas.Pen.Color := cl;
 flag := True;
 Image1.Canvas.MoveTo (x, y);
 end;

```

```

 if brush1 Then
 begin
 Image1.Canvas.Brush.Color := cl;
 Image1.Canvas.FloodFill (x, y, Image1.Canvas.Pixels [x, y], fsSurface);
 end;

```

```

end;
procedure TForm1.Image1MouseMove (Sender: TObject; Shift: TShiftState;
x, y: Integer);
begin
 if flag Then Image1.Canvas.LineTo(x, y);
end;

```



Рис. 4

```
procedure TForm1.Image1MouseUp (Sender: TObject; Button:
 TMouseButton; Shift: TShiftState; x, y: Integer);
```

```
begin
 flag := False;
end;
```

- 11) Проверьте работу программы. Сохраните проект в папке Упражнение 48.

Созданный графический редактор позволяет открывать графические файлы, создавать и редактировать изображения, сохранять их.



### Компьютерное тестирование

Выполните тестовое задание 48 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## Практическая работа 16

### Создание программы с графическими компонентами

**Задание:** составить программу для просмотра рисунков.

**Оборудование:** компьютер с установленной средой программирования Lazarus.

### Ход работы

*При работе за компьютером соблюдайте правила безопасности.*

1. Найдите четыре рисунка с петриковской росписью и в графическом редакторе установите их размеры  $80 \times 80$  пикселей. Создайте папку Практическая работа 16 и сохраните в ней обработанные графические файлы.

#### I. Размещение элементов управления на форме

2. Создайте новый проект и сохраните его в папке Практическая работа 16.
3. Измените заголовок формы на «Элементы петриковской росписи». Добавьте на форму четыре компонента BitBtn (рис. 1), задайте для свойств Width и Height значение 80 пикселей в соответствии с размерами подготовленных изображений. Очистите содержимое поля свойства Caption в окне Инспектор объектов, на каждую из кнопок установите изображения рисунков (свойство Glyph).



Рис. 1

- ▶ 4. Добавьте на форму компонент Panel  (Панель) из вкладки Standard Палитры компонентов. Панель используют в качестве контейнера для других компонентов. Свойства BevelOuter и BevelInner отвечают за стиль оформления внешней и внутренней рамок панели соответственно. Выберите для этих свойств значение *bvLowered* (Рамка утоплена). Свойству BorderWidth (Ширина рамки) задайте значение 2.
- ▶ 5. Разместите компонент Image на компоненте Panel. Свойству Stretch компонента Image задайте значение True. Свойству Align задайте значение *alClient* для того, чтобы компонент Image занимал все доступное пространство компонента Panel.
- ▶ 6. Добавьте на форму кнопки Увеличить рисунок и Уменьшить рисунок.
- ▶ 7. Добавьте на форму пятый компонент BitBtn. Свойство Kind кнопки измените на «Close».

## II. Создание программного кода

- ▶ 8. Запрограммируйте для каждой кнопки с рисунком открытие в области компонента Image соответствующего файла (рис. 2). Вместо названия Рисунок1.bmp напишите название того файла, изображение из которого вы поместили на данную кнопку:  
Image1.Picture.LoadFromFile ('Рисунок1.bmp');
- ▶ 9. Создайте процедуру обработки события OnClick для кнопки Увеличить рисунок.  
В программном коде созданной процедуры запишите операторы:  
Panel1.Height := Panel1.Height + 5;  
Panel1.Width := Panel1.Width + 5;  
Аналогично запрограммируйте уменьшение размеров объекта Panel1 на 5 пикселей.
- ▶ 10. Добавьте на форму компонент Label. Добавьте в процедуры обработки событий для кнопок BitBtn1–BitBtn4 операторы вывода названия файла.



Рис. 2

## III. Тестирование программы

- ▶ 11. Проверьте работу программы.
- ▶ 12. Сохраните проект. Завершите работу за компьютером.

**Сделайте вывод:** как добавлять изображение на форму; как использовать пиктографические кнопки для разработки наглядного интерфейса.

# РАЗДЕЛ 10

## РЕШЕНИЕ КОМПЕТЕНТНОСТНЫХ ЗАДАЧ. ВЫПОЛНЕНИЕ ИНДИВИДУАЛЬНЫХ И ГРУППОВЫХ УЧЕБНЫХ ПРОЕКТОВ



§ 49. Решение компетентностных задач

§ 50. Выполнение учебных проектов

## ПОВТОРЯЕМ



В современном информационном обществе все большее значение приобретают такие компетенции, как умение находить нужную информацию, критически ее обрабатывать, делать выводы и умозаключения, используя для работы новейшие информационные технологии.

Компетентность в переводе с латинского *competentia* означает круг вопросов, в которых человек хорошо осведомлен, имеет знания и опыт. Из курса информатики 5–7 классов вы знаете, что для решения компетентностных задач нужно выполнить ряд заданий по предложенному плану как под руководством учителя, так и самостоятельно. Ваша работа будет более эффективной при условии активного использования интернет-ресурсов для поиска информации и оценки результатов выполнения заданий.

1. Какое значение имеет информация в современном обществе?
2. Какие проекты вам приходилось выполнять?
3. Какие информационные технологии вы использовали при работе над проектами?
4. Как осуществлять поиск материалов в Интернете?
5. Каким образом можно представить результаты выполненной работы?



В этом разделе вы узнаете, как спланировать деятельность при выполнении учебного проекта, разработать стратегию решения задач из разных сфер человеческой деятельности, какие информационные технологии удобно использовать, научитесь оценивать результаты собственной деятельности

## § 49. Решение компетентностных задач

Человеку в информационном обществе необходимо не только ориентироваться в потоке информации и уметь ее анализировать, а и организовывать жизнь таким образом, чтобы при наименьших затратах времени и средств получать от своей деятельности наибольшую отдачу. Будем этому учиться, решая компетентностные задачи.

**Компетентностные задачи** по информатике — это проблемные задачи из различных сфер человеческой деятельности, которые решаются с помощью средств ИКТ.

Для решения задачи вам нужно будет провести поиск и отбор нужной информации как в печатных источниках (учебники, справочники), так и в интернет-ресурсах (электронные энциклопедии, интернет-издания).

### Этапы решения задачи

|                                         |                                                                                                                                                                                                                              |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Постановка задачи                    | Содержательный анализ формулировки задачи: какие данные есть в условии; какие данные, в каком виде должны быть получены; какие данные нужно дополнительно найти                                                              |
| 2. Поиск информации                     | Поиск информационных материалов, направленный на конкретизацию содержания описанной ситуации. Формирование списка печатных источников и интернет-ресурсов                                                                    |
| 3. Построение информационной модели     | Информационная модель — это описание существенных для поставленной задачи свойств и закономерностей поведения объектов. Информационные модели могут быть представлены в виде текстов, рисунков, таблиц, схем, формул и т. п. |
| 4. Определение средств обработки данных | Отбор средств обработки данных (текстовый процессор, графический редактор, редактор презентаций, табличный процессор, среда программирования), необходимых для поэтапного решения задачи                                     |
| 5. Обработка данных                     | Поэтапное выполнение задачи средствами ИКТ                                                                                                                                                                                   |
| 6. Представление результатов            | Выбор формы представления результатов решения задачи. Оформление и презентация полученных результатов                                                                                                                        |
| 7. Анализ результатов                   | Оценка полноты и достоверности результатов решения задачи. Формулировка ответа на проблемный вопрос задачи                                                                                                                   |

Рассмотрим задачу.

Семья из трех человек планирует путешествие по маршруту Киев — Львов — Одесса — Киев. Нужно определить, путешествие каким видом транспорта — поездом или автомобилем, например Toyota Prius 1.8 (рис. 49.1), — будет более дешевым, и обосновать выбор.



Рис. 49.1

1. Постановка задачи. Определить данные, содержащиеся в условии задачи. (*Количество человек; маршрут; транспортные средства.*)  
Какие данные должны быть получены? (*Стоимость путешествия на поезде и автомобиле для последующего их сравнения.*)  
Какие данные нужно дополнительно найти в других источниках? (*Длина маршрута; цена на горючее; цена билетов на поезд.*)
2. Поиск информации. Определить ключевые слова для поиска нужной информации в Интернете. Провести поиск и сохранить URL-адреса использованных сайтов.
3. Построение информационной модели. После анализа данных построить информационную модель задачи.

Для задач, требующих вычислений, целесообразно составить математическую модель задачи. *Математическая модель* — это вид информационной модели, в которой связи различных параметров объекта представлены посредством математических формул и понятий.

**Математическая модель задачи:**

*Дано:*  $VK$  — стоимость одного билета на поезд, грн;

$VB$  — стоимость 1 л бензина, грн;

$K$  — количество горючего на 100 км пути, л;

$S_1, S_2, S_3$  — расстояния между городами, км.

*Найти:*  $VP$  — стоимость путешествия на поезде, грн;

$VA$  — стоимость путешествия на автомобиле, грн;

Сравнить значения  $VP$  и  $VA$ .

*Связь между данными и результатом:*

$$VP = VK * 3; VA = (S_1 + S_2 + S_3) / 100 * K;$$

Если  $VP < VA$ , то ответ: «дешевле ехать на поезде», если  $VP > VA$ , то ответ: «дешевле ехать на автомобиле», если  $VP = VA$ , то ответ: «можно выбрать любой вариант».

4. Определение средств обработки данных. Вычисления удобно выполнять в табличном процессоре Excel. Обобщить результаты решения задачи можно в текстовом процессоре Word или в редакторе презентаций PowerPoint.
5. Обработка данных. Создать электронную таблицу, в которой: указать расстояния между городами, норму затрат и цену горючего, количество билетов и цены на них; рассчитать сумму, которую израсходует семья на путешествие: на поезде; на автомобиле. Построить диаграмму для сравнения полученных результатов.
6. Представление результатов. Для обсуждения будущего путешествия создать отчет о проведенной работе, в который включить:

- результаты вычислений;
  - диаграмму, построенную по полученным результатам;
  - описание исторических мест, которые следует посетить (рис. 49.2);
  - список использованных материалов. Для материалов, найденных в Интернете, укажите URL-адреса сайтов.
7. Анализ результатов. На основании отчета сделать выводы о наиболее дешевом путешествии.

Таким образом, для решения компетентностной задачи нужно проанализировать предложенную ситуацию, поставить цель, спланировать результат, разработать алгоритм решения задачи, выбрать оптимальную для решения конкретной задачи информационную технологию, проанализировать результат и сделать выводы.



Рис. 49.2

### Вопросы для самопроверки



1. Как вы понимаете термин «компетентностная задача»?
2. Из каких этапов состоит процесс решения задачи?
3. Что такое информационная модель?
4. Почему при решении различных задач могут быть составлены разные информационные модели?
5. Как можно представить информационную модель задачи?
6. Что влияет на выбор информационной технологии для обработки данных?

### Упражнение 49



- ▶▶ Решить задачу о выборе транспорта для вашего путешествия по городам Украины.
- 1) Найдите данные, необходимые для решения задачи: длина маршрута Киев — Львов — Одесса — Киев; цена на горячее; цена проезда одного человека по указанному маршруту. В текстовом редакторе создайте перечень интернет-ресурсов, на которых найдены необходимые информационные материалы.

- 2) Создайте электронную таблицу для расчета стоимости путешествия согласно математической модели задачи:

|   | A                                                                                                                            | B                                          | C                               | D                      | E                                     | F                             | G                                   |
|---|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------|------------------------|---------------------------------------|-------------------------------|-------------------------------------|
|   |                                                                                                                              | стоимость билетов на поезд на 1 чел., грн. | стоимость поездки поездом, грн. | цена 1 л бензина, грн. | количество горючего на 100 км пути, л | расстояние между городами, км | стоимость поездки автомобилем, грн. |
| 1 |                                                                                                                              |                                            |                                 |                        |                                       |                               |                                     |
| 2 | Киев-Львов                                                                                                                   |                                            | =B2*3                           |                        |                                       |                               | =F2/100*E2*D2                       |
| 3 | Львов-Одесса                                                                                                                 |                                            | =B3*3                           |                        |                                       |                               | =F3/100*E3*D3                       |
| 4 | Одесса-Киев                                                                                                                  |                                            | =B4*3                           |                        |                                       |                               | =F4/100*E4*D4                       |
| 5 |                                                                                                                              |                                            | =СУММ(C2:C4)                    |                        |                                       |                               | =СУММ(G2:G4)                        |
| 6 | <b>Ответ:</b> =ЕСЛИ(C5<G5;"дешевле ехать на поезде";ЕСЛИ(C5>G5;"дешевле ехать на автомобиле";"можно выбрать любой вариант")) |                                            |                                 |                        |                                       |                               |                                     |

- 3) Внесите в таблицу необходимые для решения задачи данные.  
 4) Постройте диаграмму, на которой отобразите стоимости поездки заданным маршрутом на автомобиле и поезде.  
 5) Создайте отчет о результатах решения задачи.  
 6) Сделайте выводы о путешествии. Сохраните отчет в файле Упражнение 49.

### Задачи для самостоятельной работы



1. Таня обратилась к вам за помощью. Девочка хочет пригласить на свой день рождения 9 друзей в детское кафе. Нужно: 1) подготовить приглашения на праздник; 2) написать сценарий праздника; 3) создать поздравительный видеоклип; 4) составить меню; 5) вычислить стоимость проведения праздника.
2. Учитель математики обратился к вам с просьбой создать программу для нахождения корней квадратного уравнения  $ax^2 + bx + c = 0$ . Нужно: 1) создать программу (рис. 1); 2) создать электронную таблицу для проверки правильности работы программы по образцу (рис. 2); 3) в текстовом редакторе оформить отчет о выполнении задачи по плану: постановка задачи; математическая модель задачи; блок-схема алгоритма; программный код; результаты тестирования программы.

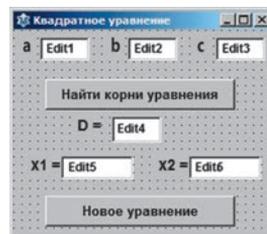


Рис. 1

|   | A | B | C | D             | E                                                             |           |
|---|---|---|---|---------------|---------------------------------------------------------------|-----------|
| 1 | a | b | c | D             | x1                                                            | x2        |
| 2 |   |   |   | =B2*2-4*A2*C2 | =ЕСЛИ(D2>0;(-B2+Сqrt(D2))/2/A2;IF(D2=0;B2/2/A2;"нет корней")) | =ЕСЛИ(D2> |

Рис. 2

3. Ваши знакомые после окончания школы планируют сдавать ВНО по истории Украины. При подготовке у них возникли проблемы с распознаванием исторических памятников по

изображениям. Нужно создать справочник в виде таблицы, содержащей название объекта, его изображение и краткую историческую справку. Перечень исторических объектов:

- Софийский собор в Киеве (I пол. XI в.);
- Спасо-Преображенский собор в Чернигове (1036 г.);
- Армянский собор во Львове (1363–1370);
- Замок Любарта, или Луцкий замок (II пол. XIV в.);
- Хотинская крепость (II пол. XIII в. — I пол. XIV в.).

### Компьютерное тестирование



Выполните тестовое задание 49 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 50. Выполнение учебных проектов

Успех в современном мире во многом определяется способностью человека планировать свою жизнь: определять ближнюю и дальнюю перспективу, находить и использовать необходимые ресурсы, намечать план действий и оценивать достижение поставленных целей.

Проект представляет собой самостоятельное решение определенной проблемы с презентацией результатов работы.

**Учебный проект** — совместная учебно-познавательная, исследовательская, творческая или игровая деятельность учащихся-партнеров, имеющих общую цель, согласованные методы и способы деятельности, направленные на достижение общего результата в решении определенной проблемы.

Выполнение проекта на уроках информатики позволяет не только овладеть компьютерными технологиями, но и научиться приемам самостоятельной работы начиная с выбора темы и постановки целей до разработки алгоритма и создания готового электронного продукта.

### Какие бывают проекты

Рассмотрим некоторые виды учебных проектов.

- **Практико-ориентированный проект** нацелен на социальные интересы участников проекта или внешнего заказчика.
- **Исследовательский проект** по структуре напоминает научное исследование. Он включает обоснование актуальности выбранной темы, определение задач исследования, выдвижение гипотезы

с дальнейшей ее проверкой, обсуждение полученных результатов. При этом используются методы современной науки: лабораторный эксперимент, моделирование, социологический опрос и другие.

- **Информационный проект** направлен на сбор информации о каком-либо объекте или явлении с целью ее анализа, обобщения и представления перед широкой аудиторией (публикация в СМИ, в частности в Интернете).
- **Творческий проект** предусматривает максимально свободный и нетрадиционный подход к представлению результатов. Это могут быть литературные альманахи, театральные спектакли, спортивные игры, видеофильмы и т. п.
- **Ролево-игровой проект.** Участники берут на себя социальные роли либо роли литературных или исторических персонажей с целью моделирования различных ситуаций социальных или деловых отношений, предусмотренных условиями игры.

**Работа над проектами проходит в несколько этапов:**

1. Подготовка к работе над проектом.
2. Выбор темы.
3. Постановка цели и задач проекта.
4. Поиск информации для проекта.
5. Выполнение задач проекта.
6. Подготовка к защите проекта.
7. Презентация (защита) проекта.
8. Анализ результатов проектной работы.

### Проект «Роль всемирной сети Интернет в нашей жизни»

Рассмотрим этапы работы над проектом на примере проекта «Роль всемирной сети Интернет в нашей жизни».

**Цель проекта:**

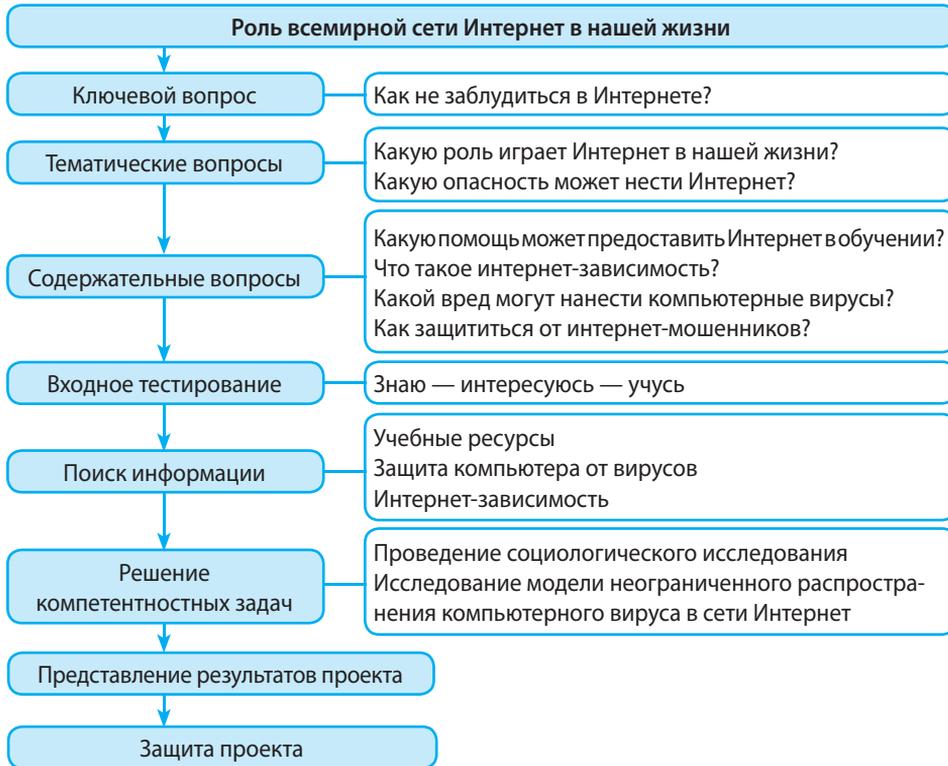
- исследовать, какое место занимает Интернет в нашей жизни;
- узнать, какие возможности и опасности несет в себе сеть;
- подготовить рекомендации для активных пользователей Интернета.

**Продолжительность проекта:** 4 урока.

В ходе выполнения проекта вы проанализируете пользу и возможный вред Интернета и найдете ответ на вопрос: как не заблудиться во всемирной сети. Вначале нужно сформулировать основные тематические вопросы проекта. Входное тестирование покажет, существуют ли у вас проблемы в пользовании Интернетом, а если существуют, то

какие именно. Далее вы объединитесь в группы для работы над заинтересовавшим вас вопросом. После окончания работы следует обобщить и оформить полученные результаты.

Для организации выполнения проекта целесообразно создать информационную модель проекта в форме карты знаний:



Реализация проекта предусматривает решение ряда компетентностных задач, которые вы рассмотрите в упражнении 50.

На этапе представления результатов проекта группа учащихся должна создать отчет, содержащий следующие сведения:

- цель проекта;
- ключевой и тематические вопросы проекта;
- задачи, которые решались в ходе выполнения проекта;
- список URL-адресов сайтов, на которых найдены необходимые информационные материалы;
- результаты выполнения задач проекта;

- выводы, сделанные участниками проекта на основании результатов выполнения задач;
- анализ достигнутых результатов проекта.

Завершающим этапом работы над проектом является его публичная защита, в ходе которой происходит презентация результатов выполнения проекта. Во время защиты проекта нужно оценить уровень оформления и подачи результатов исследования каждой группой по заранее разработанным критериям оценивания.

Работа над проектом позволит вам попробовать свои силы, применить полученные знания и умения и продемонстрировать достигнутый результат.

### Вопросы для самопроверки



1. Что такое учебный проект?
2. Какими бывают проекты?
3. Из каких этапов состоит работа над проектом?
4. Какие сведения должен содержать отчет о результатах выполнения задач проекта?
5. Какую помощь вы можете получить в Интернете при работе над проектом?
6. Какие средства защиты от интернет-опасностей вы используете?

### Упражнение 50



- ▶ Решить компетентностные задачи проекта.
1. **Исследование модели неограниченного распространения компьютерного вируса в сети Интернет**
    - 1) Ученик получил электронное письмо, к которому был прикреплен файл, содержащий компьютерный вирус ILOVEYOU. Когда ученик открыл файл, вирус удалил все графические и звуковые файлы на диске, а потом, используя адресную книгу почтовой программы, разослал такие же письма всем знакомым ученика. Вирус ILOVEYOU заразил огромное количество компьютеров по всему миру за один день. Определите, сколько компьютеров заразил компьютер ученика через 24 часа, если средний коэффициент распространения вируса равен двум адресам в час с каждого зараженного компьютера. Наличие средств антивирусной защиты не учитывать. Динамику распространения компьютерного вируса можно описать с помощью математической модели.

*Дано:*  $K_0$  — количество зараженных компьютеров в начальный момент времени ( $K_0 = 1$ );  
 $R$  — коэффициент распространения вируса ( $R = 2$ );  
 $T$  — общее время распространения вируса ( $T = 24$ );  
 $dt$  — период распространения вируса ( $dt = 1$ ).

*Найти:*  $K$  — количество зараженных компьютеров за время  $T$ .

*Связь между данными и результатом:*

$$N = T / dt; K_{i+1} = K_i + K_i * R, i = 1..N,$$

где  $N$  — количество периодов распространения вируса за время  $T$ ;  
 $K_i, K_{i+1}$  — число зараженных компьютеров в  $i$ -й и  $(i+1)$ -й периоды после начала распространения вируса.

- 2) Разработайте интерфейс проекта согласно рисунку.
- 3) В программном коде процедуры Button1Click запишите операторы:

```
var K, R, T, dt, N, i: Integer;
begin
 K := StrToInt(Edit1.Text);
 R := StrToInt(Edit2.Text);
 T := StrToInt(Edit3.Text);
 dt := StrToInt(Edit4.Text);
 N := Round(T / dt);
 For i := 1 to N do K := K + K * R;
 Edit5.Text := IntToStr(K);
end;
```



- 4) Проверьте работу программы для исходных значений задачи. Сделайте выводы. Сохраните проект в папке Упражнение 50.
- ## II. Проведение социологического исследования
- 5) Подготовьте опросник в текстовом процессоре или с использованием технологий Веб 2.0, раскрыв следующие вопросы:
    - С кем вы общаетесь в Интернете?
    - Сколько времени в день вы проводите в сети Интернет?
    - Какие опасности могут подстергать в Интернете?
    - Какие сайты вы посещаете?
    - Сообщаете ли вы информацию о себе в Интернете?
    - Что полезного есть в Интернете для обучения?
  - 6) По результатам опроса постройте соответствующие диаграммы в Excel. Сохраните файл в папке Упражнение 50.

### Компьютерное тестирование

Выполните тестовое задание 50 с автоматической проверкой на сайте [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## Учебные проекты

### 1. Тема. Цифровая информация в нашей жизни.

**Предметно-содержательная сфера:** информатика.

**Тип проекта:** исследовательский.

**Цель:** исследовать основные идеи, теоретические и практические принципы, лежащие в основе современных цифровых технологий.

**Ключевой вопрос:** как информатика помогает современному человеку?

**Тематические вопросы**

#### 1. Как закодировать звук?

*Содержательные вопросы:*

- В чем разница между аналоговой и цифровой информацией?
- Как аналоговую информацию преобразовать в цифровую и наоборот?
- Какие устройства и программные средства предназначены для обработки цифрового звука?
- Какие существуют форматы звуковых файлов и в чем их отличие?

#### 2. Почему существуют разные палитры цветов?

*Содержательные вопросы:*

- Что такое RGB-палитра и CMYK-палитра, где их используют?
- Существуют ли другие палитры цветов?
- Какие устройства и программные средства предназначены для обработки изображений?
- Какие существуют форматы графических файлов и в чем их отличие?

#### 3. Как научить компьютер грамоте?

*Содержательные вопросы:*

- Что такое таблица кодов?
- Какие таблицы кодов существуют для кодировки символов?
- Какие проблемы возникают при кодировании национальных алфавитов?
- В чем отличие разных таблиц кодов и как различаются текстовые файлы, содержащие данные в разных кодировках?

#### 4. Как оценить информацию?

*Содержательные вопросы:*

- Как оценить качество информации?
- Как оценить количество информации?
- Какие существуют единицы измерения данных?
- Как теория информации помогает решать практические задачи?

**Форма подачи результатов проекта:** презентация, буклет, видеоматериалы, вики-статья.

## 2. Тема. Выдающиеся имена и открытия в истории информатики.

**Предметно-содержательная сфера:** информатика.

**Тип проекта:** информационный.

**Цель:** исследовать историю развития вычислительной техники, роль выдающихся личностей в сфере информационных технологий на развитие информатики.

**Ключевой вопрос:** к чему приводят открытия?

**Тематические вопросы:**

1. В каких направлениях развивается информатика?

*Содержательные вопросы:*

- Какие направления включает информатика?
- Как развиваются основные составляющие информационных систем: hardware, software, brainware?

2. Можно ли заставить компьютер думать?

*Содержательные вопросы:*

- Какие открытия были сделаны в сфере информатики?
- Что такое «искусственный интеллект»?

3. С какими именами связано становление информатики как науки?

*Содержательные вопросы:*

- Кого называют основателем науки информатики?
- С чьими именами связано развитие информационно-коммуникационных технологий?

**Форма подачи результатов проекта:** презентация, буклет, видеоматериалы, вики-статья.

## 3. Тема. Выбор компьютера.

**Предметно-содержательная область:** информатика.

**Тип проекта:** ролево-игровой.

**Цель:** сделать осознанный выбор модели компьютера в соответствии с исходными данными; смоделировать ситуацию практического применения знаний о характеристиках составляющих компьютера; формировать культуру поведения в ситуации продавец — покупатель.

**Ключевой вопрос:** Как выбрать лучший компьютер?

**Тематический вопрос:** Как определить, исходя из потребностей пользователя, наиболее подходящую конфигурацию компьютера?

*Содержательные вопросы:*

- Какие технические характеристики составляющих компьютера интересуют пользователя?
- Как рассчитать стоимость выбранной комплектации?

**Форма подачи результатов проекта:** рекламная презентация, буклет.

## КОМПЬЮТЕРНЫЙ СЛОВАРЬ

**Адаптер (контроллер)** — устройство, предназначенное для подключения к компьютеру периферийного устройства и управления его работой.

**Алгоритм** — последовательность команд для исполнителя, четко определяющая, какие действия и в каком порядке следует выполнить для решения определенной задачи.

**Аналоговый сигнал** — непрерывный сигнал, который может принимать любые значения из определенного диапазона.

**Аппаратное обеспечение** — совокупность устройств и приборов, обеспечивающих функционирование информационной системы.

**Архиватор** — программа, осуществляющая упаковку файлов в архив путем сжатия данных.

**Архитектура компьютера** — описание строения компьютера и принципов работы его составляющих.

**Аудиоредактор** — программа для обработки звуковых (аудио) файлов.

**Байт** — набор из 8 последовательных битов, который воспринимается компьютером как единое целое.

**Бит** — наименьшая единица информации; обозначается двоичной цифрой 1 или 0.

**Видеоредактор** — программа для создания, редактирования и монтажа видеофайлов из отдельных видеофрагментов.

**Глобальная переменная** — переменная, доступ к которой возможен из любой точки программы.

**Двоичное кодирование** — представление данных с помощью двоичных чисел.

**Декодирование** — преобразование закодированного сообщения в форму, понятную получателю.

**Дискретный сигнал** — сигнал, который может принимать только значения из определенного набора.

**Звуковая карта** — устройство компьютера, предназначенное для преобразования аналогового электрического сигнала в дискретный и наоборот.

**Импорт файла** — открытие в среде прикладной программы файла, созданного в другой программной среде.

**Инсталляция** — процесс установки программного обеспечения на компьютер пользователя (обратный процесс — деинсталляция).

**Интерфейс пользователя** — средства взаимодействия пользователя с программой.

**Код** — набор символов или сигналов и правил их использования для кодирования сообщений.

**Кодирование** — преобразование сообщения в удобный для передачи, хранения и обработки вид.

**Колонтитул** — структурный элемент документа, расположенный в верхнем или нижнем поле страницы.

**Компилятор** — специальная программа, преобразующая текст программы, написанной языком программирования высокого уровня, в программу, представленную на машинном языке и предназначенную для выполнения на компьютере.

**Конвертер** — программа, предназначенная для перекодирования медиафайлов из одного формата в другой.

**Локальная переменная** — переменная, доступ к которой возможен только внутри процедуры, в которой она объявлена.

**Мультимедиа** (от латин. *multum* — много и *medium* — среда) — в основном интерактивные (диалоговые) системы, использующие данные разных типов.

**Операционная система** — система программ и данных, с помощью которой организуется работа компьютера.

**Память внешняя (внешнее запоминающее устройство)** — память на внешних носителях данных (диски, флеш-память).

**Память внутренняя** — быстродействующая память, расположенная на системной плате компьютера; делится на оперативную и постоянную.

**Память оперативная** — внутренняя память компьютера, предназначенная для хранения программ и данных, которые находятся в работе в данный момент.

**Память постоянная (постоянное запоминающее устройство)** — память, предназначенная для хранения программ и данных, обеспечивающих начало работы компьютера после включения.

**Переменная** — величина, значение которой может изменяться в процессе выполнения программы.

**Периферийное оборудование** — внешние устройства компьютера.

**Программирование** — процесс создания компьютерных программ.

**Программное обеспечение** — совокупность программ, установленных на компьютере и предназначенных для обеспечения его работы и решения задач пользователя.

**Редактирование документа** — исправление ошибок, внесение изменений в документ.

**Система программирования** — совокупность программ и других средств, обеспечивающих создание компьютерных программ на определенном языке программирования.

**Ссылка на ячейку** — адрес ячейки, к которому добавляется имя листа, если ссылка осуществляется на ячейку, расположенную на другом листе.

**Стиль** — способ форматирования структурного элемента текстового документа.

**Условное форматирование** — автоматическое изменение формата ячейки на заданный, если значение данных в ней соответствует определенному условию.

**Форматирование документа** — изменение внешнего вида документа.

**Шаблон документа** — документ, который хранится в файле особого формата и используется как основа для создания новых документов.

**Электронная таблица** — структура данных, представленных в виде таблицы, обеспечивающая автоматические вычисления по введенным в ячейки формулам.

**Язык программирования** — искусственный язык, представляющий собой систему обозначений и правил для записи алгоритмов в форме, пригодной для дальнейшего их выполнения на компьютере.

## Предметный указатель

### А

Адаптер 24  
Архив 30  
Архивация данных 30  
Архитектура компьютера 21  
Аудиоадаптер 22, 24, 63

### В

Видеоадаптер 22, 24  
Видеоэффект 70  
Внешнее запоминающее устройство 24  
Выражение 82

### Г

Гиперссылка 51  
Графические примитивы 211

### Д

Двоичное кодирование 10  
Деинсталляция программы 32  
Декодирование 8  
Диаграмма 103  
Диапазон ячеек 77  
Длина строки 163

### З

Захват  
— аудио 61  
— видео 63

### И

Импорт 66  
Инсталляция 32  
Интерфейс среды программирования 110, 111

### К

Код 8  
— символа 10  
— ASCII 11  
— KOI8-R 11  
— Юникод 11

Кодек 59

Кодирование 8  
Колонки 44  
Колонтитулы 50  
Комментарий 131

Компилятор 114

Компонент 109

Конвертер 64

Константа 139

Контроллер 22, 24

Конфигурация компьютера 26

### Л

Лицензия 31, 32

### М

Масштабирование 221  
Метод 109, 126  
Мультимедийное устройство 25

### Н

Непечатаемый знак 45  
Носитель данных 30

### О

Обработчик событий 119, 120  
Объект 109, 111  
Объектно-ориентированное программирование 109  
Оператор  
— ветвления 175  
— выбора 182  
— полного ветвления 176  
— присвоения 140  
— составной 176  
— цикла с параметром 199  
— цикла с постусловием 195  
— цикла с предусловием 190

Операции

— логические 157  
— отношения 157

## П

Память

- внутренняя 24
  - компьютера 23
  - оперативная 24
  - постоянная 24
- Переключатель 186
- Преобразование типов 143, 158, 167

Полотно 212

Приемник 7

Принтер 25

Программное обеспечение 28

- прикладное 30
- системное 29
- служебное 30

Проект 113

Проектор 26

Промежуточные итоги 101

Процедура обработки  
событий 117, 119

Процессор 22

## Р

Разархивация 31

Раздел 49

Разрыв раздела 49

Раскадровка 65

Расширенный фильтр 100

Редактор кода 117, 118

Режим структуры 50

## С

Селектор 182

Системная плата 22

Событие 117, 126

Событийно-ориентированное  
программирование 117

Содержание 51

Сортировка данных 94

Список 37

— многоуровневый 38

— маркированный 38

— нумерованный 38

Среда программирования 109

Стиль 48

Структура документа 49

Ссылка на ячейку

— абсолютная 78

— относительная 78

— смешанная 78

## Т

Таблица 41

Тип величины 139

## У

Условие

— простое 157

— составное 157, 158

## Ф

Флажок 187, 188

Форматирование

— данных 96

— носителя данных 30

— списка 39

— условное 96

Форма 110

Формат

— аудиофайла 60

— видеофайла 59

— файла текстового документа 37

Формула 45

Функция 82, 83

— логическая 82

— математическая 86

— статистическая 87

## Ш

Шаблон документа 48

## Э

Элемент управления 114, 125, 126

## Я

Язык программирования 108

# Содержание

## Раздел 1. Кодирование данных

|                                                  |    |
|--------------------------------------------------|----|
| § 1. Кодирование и декодирование сообщений ..... | 7  |
| § 2. Двоичное кодирование .....                  | 10 |
| Практическая работа 1 .....                      | 13 |

## Раздел 2. Аппаратно-программное обеспечение компьютера

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| § 3. История обработки информационных объектов .....                                  | 17 |
| § 4. Архитектура компьютера. Технические характеристики устройств<br>компьютера ..... | 21 |
| Практическая работа 2 .....                                                           | 27 |
| § 5. Программное обеспечение компьютера .....                                         | 28 |
| Практическая работа 3 .....                                                           | 34 |

## Раздел 3. Обработка текстовых данных

|                                                                        |    |
|------------------------------------------------------------------------|----|
| § 6. Списки в текстовом документе .....                                | 37 |
| § 7. Таблицы в текстовом документе .....                               | 41 |
| § 8. Обработка текстового документа, содержащего различные объекты ..  | 44 |
| § 9. Оформление документов. Структура сложного текстового документа .. | 48 |
| Практическая работа 4 .....                                            | 54 |
| Практическая работа 5 .....                                            | 55 |

## Раздел 4. Обработка объектов мультимедиа

|                                                                        |    |
|------------------------------------------------------------------------|----|
| § 10. Форматы видео- и аудиофайлов .....                               | 59 |
| § 11. Программное обеспечение для обработки объектов мультимедиа ..... | 61 |
| § 12. Создание аудио- и видеофрагментов .....                          | 65 |
| § 13. Обработка, сохранение и размещение видеоклипа .....              | 69 |
| Практическая работа 6 .....                                            | 73 |
| Практическая работа 7 .....                                            | 74 |

## Раздел 5. Технологии обработки числовых данных в среде табличного процессора

|                                                          |     |
|----------------------------------------------------------|-----|
| § 14. Типы ссылок на ячейки в Excel .....                | 77  |
| Практическая работа 8 .....                              | 81  |
| § 15. Логические функции .....                           | 82  |
| § 16. Математические и статистические функции .....      | 86  |
| § 17. Параметры страницы. Печать таблицы .....           | 89  |
| § 18. Упорядочение данных. Условное форматирование ..... | 94  |
| Практическая работа 9 .....                              | 98  |
| § 19. Расширенные фильтры. Промежуточные итоги .....     | 100 |
| § 20. Диаграммы и графики .....                          | 103 |
| Практическая работа 10 .....                             | 106 |

## Раздел 6. Основы событийно- и объектно-ориентированного программирования

|                                                   |     |
|---------------------------------------------------|-----|
| § 21. Знакомство со средой программирования ..... | 109 |
| § 22. Создание простейшего проекта .....          | 113 |

|                                                        |     |
|--------------------------------------------------------|-----|
| § 23. Создание программы . . . . .                     | 116 |
| § 24. Основные компоненты программы . . . . .          | 121 |
| § 25. Свойства и методы элементов управления . . . . . | 125 |
| § 26. Отладка программного кода . . . . .              | 128 |
| Практическая работа 11 . . . . .                       | 133 |
| Практическая работа 12 . . . . .                       | 135 |

### **Раздел 7. Алгоритмы работы с объектами и величинами**

|                                                             |     |
|-------------------------------------------------------------|-----|
| § 27. Величины и их свойства . . . . .                      | 139 |
| § 28. Элементы управления для ввода данных . . . . .        | 142 |
| Практическая работа 13 . . . . .                            | 147 |
| § 29. Величины целых типов . . . . .                        | 149 |
| § 30. Величины вещественных типов . . . . .                 | 153 |
| § 31. Величины логического типа . . . . .                   | 157 |
| § 32. Величины символьного типа . . . . .                   | 160 |
| § 33. Величины строкового типа . . . . .                    | 163 |
| § 34. Преобразование величин одного типа в другой . . . . . | 167 |
| Практическая работа 14 . . . . .                            | 171 |

### **Раздел 8. Алгоритмы с повторениями и ветвлениями**

|                                                                      |     |
|----------------------------------------------------------------------|-----|
| § 35. Алгоритмы с ветвлениями . . . . .                              | 175 |
| § 36. Составление программ с ветвлениями . . . . .                   | 178 |
| § 37. Поливариантное ветвление . . . . .                             | 182 |
| § 38. Компоненты выбора . . . . .                                    | 186 |
| § 39. Алгоритмы с повторениями. Цикл с предусловием . . . . .        | 190 |
| § 40. Алгоритмы с повторениями. Цикл с постусловием . . . . .        | 195 |
| § 41. Алгоритмы с повторениями. Цикл с параметром . . . . .          | 199 |
| § 42. Составление циклических алгоритмов обработки величин . . . . . | 203 |
| Практическая работа 15 . . . . .                                     | 207 |

### **Раздел 9. Графическое отображение данных**

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| § 43. Базовые графические примитивы . . . . .                       | 211 |
| § 44. Настройка свойств графических примитивов . . . . .            | 215 |
| § 45. Создание программ с графическим отображением данных . . . . . | 220 |
| § 46. Графические компоненты . . . . .                              | 223 |
| § 47. Вывод рисунков из внешних файлов . . . . .                    | 227 |
| § 48. Разработка программ с графическими компонентами . . . . .     | 231 |
| Практическая работа 16 . . . . .                                    | 235 |

### **Раздел 10. Решение компетентностных задач.**

#### **Выполнение индивидуальных и групповых учебных проектов**

|                                                |     |
|------------------------------------------------|-----|
| § 49. Решение компетентностных задач . . . . . | 239 |
| § 50. Выполнение учебных проектов . . . . .    | 243 |
| Учебные проекты . . . . .                      | 248 |

|                                |     |
|--------------------------------|-----|
| Компьютерный словарь . . . . . | 250 |
| Предметный указатель . . . . . | 252 |

## Сведения о состоянии учебника

| №<br>п/п | Фамилия и имя<br>ученика / ученицы | Учебный<br>год | Состояние учебника |                 |
|----------|------------------------------------|----------------|--------------------|-----------------|
|          |                                    |                | в начале<br>года   | в конце<br>года |
| 1        |                                    |                |                    |                 |
| 2        |                                    |                |                    |                 |
| 3        |                                    |                |                    |                 |
| 4        |                                    |                |                    |                 |
| 5        |                                    |                |                    |                 |

Навчальне видання

*БОНДАРЕНКО Олена Олександрівна*  
*ЛАСТОВЕЦЬКИЙ Василь Васильович*  
*ПИЛИПЧУК Олександр Павлович*  
*ШЕСТОПАЛОВ Євген Анатолійович*

### «ІНФОРМАТИКА»

**підручник для 8 класу загальноосвітніх навчальних закладів  
з навчанням російською мовою  
(російською мовою)**

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів. Продаж заборонено

Провідний редактор *І. Л. Морєва*. Редактори *Л. А. Каюда, Ю. М. Миронова*.  
Художник *В. Д. Хорошенко*. Художнє оформлення *О. С. Юхтман*.  
Технічний редактор *В. І. Труфєн*. Комп'ютерна верстка *О. В. Сміян, С. В. Яшиш*.  
Коректор *Н. В. Красна*.

Підписано до друку 15.07.2016. Формат 70×90/16.

Папір офсетний. Гарнітура Шкільна. Друк офсетний.

Ум. друк. арк. 18,67. Обл.-вид. арк. 24,27. Наклад 3016 прим. Зам. № 160725.

ТОВ Видавництво «Ранок».

Свідоцтво ДК № 3322 від 26.11.2008. 61071 Харків, вул. Кібальчича, 27, к. 135.

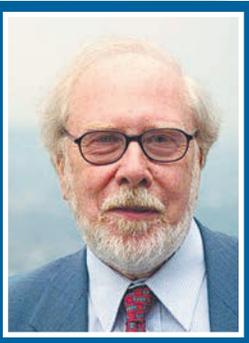
Адреса редакції: 61145 Харків, вул. Космічна, 21-а, 7 поверх.

E-mail: office@ranok.com.ua. Тел. (057) 701-11-22, 719-48-65, тел./факс (057) 719-58-67.

**www.ranok.com.ua**

Під час підготовки видання були використані матеріали із сайтів:  
freepik.com, freeimages.com, morguefile.com, pixabay.com, uk.wikipedia.org

Надруковано у друкарні ФОП Садковий В.Л.  
м. Харків, вул. Киргизька, 21. Тел. (057) 357-12-64, email: druk\_isp@i.ua



**Н. Вирт:**

*Ключ к тайнам компьютеров — в гармонии математики, инженерии и программирования.*

Язык программирования **Pascal** был разработан в 1971 г. Никлаусом Виртом, профессором компьютерных наук Швейцарской высшей технической школы (Цюрих), как язык для обучения будущих программистов. Язык назван в честь выдающегося французского математика, физика, литератора и философа Блеза Паскаля (1623–1662 гг.) — изобретателя механического калькулятора (паскалины).

**Delphi** — объектно-ориентированная среда для визуальной разработки программ, основанная на языке Object Pascal. Свое название среда Delphi получила благодаря поговорке, известной в Древней Греции: «Хочешь поговорить с оракулом — иди в Дельфы». Свободным аналогом среды программирования Delphi является **Lazarus**, основанный на компиляторе Free Pascal.

В мире насчитывается около 8500 языков программирования, но востребованных не так уж и много. Если вы собираетесь в будущем работать в сфере IT-технологий, вам следует ознакомиться с перечнем самых популярных современных языков программирования.



**Блез Паскаль**



**Паскалина, созданная Б. Паскалем в 1642 г.**



**Символ IDE Lazarus — гепард**



**Java** — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems. Название язык получил в честь кофе марки Java. На языке Java работает около 3 млрд мобильных телефонов. В мире насчитывается около 9 млн Java-программистов.



**C/C++** — это универсальный выбор для разработки настольного программного обеспечения, компьютерных игр, а также приложений, требующих большого объема памяти для работы.



**C# (Си Шарп)** — популярный язык для разработки приложений для Windows. Для разработки игр Unity 3D использует C# как один из основных языков.



**Objective C** пригодится в том случае, если вы собираетесь заняться разработкой приложений для Apple Mac OS X, айфонов и айпадов.



**PHP** — инструмент для создания современных веб-приложений. На PHP разработано большинство сайтов, ориентированных на большой объем данных.



**Python.** Веб-приложения, статистика, анализ данных, созданные для пользователя интерфейсы — для каждой задачи в Python найдется соответствующий фреймворк (программное обеспечение, облегчающее разработку и объединение разных компонентов в большой программный проект).



**JavaScript** используется любым современным сайтом. Это ключевой язык для создания интерактивных сайтов или построения предназначенных для пользователя интерфейсов.



**Visual Basic.NET** — это объектно-ориентированный язык, позволяющий создавать широкий спектр программного обеспечения: графические и игровые программы, сложные высоконагруженные веб-сервисы.



**Ruby on Rails** становится популярным среди компаний-новичков, так как является более простым по сравнению с Java или Visual Basic.NET для быстрой разработки веб-приложений.

**Марк Цукерберг,**  
руководитель компании Facebook Inc.:

*Я считаю, что в будущем все, а не только программисты, будут связаны с элементами программирования.*



# Классификация персональных компьютеров (ПК)

## НАСТОЛЬНЫЕ



**Десктоп** — системный блок в отдельном корпусе



**Неттоп** — портативный системный блок может крепиться к монитору



**Моноблок** — системный блок встроен в корпус монитора

## КАРМАННЫЕ



**Коммуникатор** — ПК с функцией мобильного телефона



**Карманный** — миниатюрный ПК с клавиатурой или без нее



**Смартфон** — мобильный телефон с функциями компьютера

## ПОРТАТИВНЫЕ



**Ноутбук** — все составляющие в одном корпусе



**Нетбук** — упрощенный и уменьшенный ноутбук



**Планшет** — плоский ПК без клавиатуры, с сенсорным экраном



**Планшетный ноутбук** — ноутбук с сенсорным экраном



В Украине созданы мощные центры суперкомпьютерных вычислений в Национальном техническом университете «Киевский политехнический институт» и Институте кибернетики им. В. М. Глушкова НАН Украины.