

ИНФОРМАТИКА

**В. М. Котов
А. И. Лапо
Ю. А. Быкадоров
Е. Н. Войтехович**

9



ИНФОРМАТИКА

Учебное пособие для 9 класса
учреждений общего среднего образования
с русским языком обучения

*Допущено
Министерством образования
Республики Беларусь*

Минск «Народная асвета» 2019

Правообладатель Народная асвета

УДК 004(075.3=161.1)
ББК 32.81я721
И74

Авторы:

В. М. Котов, А. И. Лапо, Ю. А. Быкадоров, Е. Н. Войтехович

Рецензенты:

кафедра инженерной психологии и эргономики факультета компьютерного проектирования учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (кандидат физико-математических наук, доцент *И. Ф. Киринович*); учитель информатики квалификационной категории «учитель-методист» государственного учреждения образования «Средняя школа № 4 г. Дзержинска» *С. Г. Пузиновская*

ISBN 978-985-03-3113-7

© Оформление. УП «Народная асвета», 2019

Правообладатель Народная асвета

СОДЕРЖАНИЕ

От авторов	6
------------------	---

Глава 1. ИНФОРМАЦИОННЫЕ РЕСУРСЫ СЕТИ ИНТЕРНЕТ

§ 1. Организация сети Интернет	8
1.1. Организация службы WWW. Понятие о веб-сервере	—
1.2. Адресация в сети Интернет	9
§ 2. Национальные информационные ресурсы	11
§ 3. Поиск в сети Интернет	14
§ 4. Облачные технологии. Сервисы совместной работы	18
§ 5. Сетевой этикет и меры безопасности в сети Интернет	24

Глава 2. АЛГОРИТМЫ ОБРАБОТКИ СТРОКОВЫХ ВЕЛИЧИН

§ 6. Основные алгоритмические конструкции и типы данных	28
6.1. Основные алгоритмические конструкции	—
6.2. Вспомогательные алгоритмы	29
6.3. Типы данных	30
6.4. Примеры задач	32
§ 7. Строковые величины	34
7.1. Ввод, вывод, присваивание строковых величин	—
7.2. Сравнение и сложение строковых величин	36
§ 8. Стандартные процедуры и функции для работы со строковыми величинами	42
8.1. Поиск в строке	—
8.2. Копирование, вставка и удаление символов	45
§ 9. Составление алгоритмов обработки строковых величин	49
9.1. Анализ текста на наличие различных символов	—
9.2. Преобразование строк	52
9.3. Преобразование строк в числа и чисел в строки	58

Глава 3. ОБРАБОТКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

§ 10. Понятие электронной таблицы	64
10.1. Назначение и типовые задачи обработки информации в электронных таблицах	—
10.2. Структура электронной таблицы	65
10.3. Типы данных в электронной таблице	66
§ 11. Создание электронной таблицы	69
11.1. Ввод данных	—
11.2. Текст	—
11.3. Числовые данные	70
11.4. Последовательности	—
11.5. Формулы	71
11.6. Редактирование данных и формул	72
§ 12. Редактирование электронной таблицы	75
12.1. Объединение ячеек	—
12.2. Операции с диапазонами ячеек	—

4 Содержание

12.3. Вставка и удаление строк и столбцов	76
12.4. Изменение ширины столбца и высоты строки	—
§ 13. Форматирование электронной таблицы	78
13.1. Форматирование ячеек	—
13.2. Границы и заливка	79
13.3. Форматы чисел	80
13.4. Подготовка таблицы к печати	82
§ 14. Ссылки в формулах	86
§ 15. Использование стандартных функций	90
15.1. Функции в Excel	—
15.2. Категории функций	91
§ 16. Сортировка и фильтрация данных	97
16.1. Сортировка данных	—
16.2. Фильтрация данных	98
§ 17. Построение диаграмм	104
17.1. Вставка диаграммы	—
17.2. Настройка элементов диаграммы	106
§ 18. Использование электронных таблиц для выполнения практических заданий из различных предметных областей	110
18.1. Решение расчетных задач	—
18.2. Ввод прогрессий	111
18.3. Аналитическая графика	112
18.4. Распределение данных и организация поиска	114

Глава 4. КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ МОДЕЛИ

§ 19. Модели и виды моделей	117
19.1. Понятие модели	—
19.2. Основные виды моделей	118
19.3. Виды информационных моделей	—
19.4. Компьютерные информационные модели	119
19.5. Средства реализации компьютерных моделей	120
§ 20. Моделирование и его этапы	121
20.1. Моделирование	—
20.2. Научные методы в моделировании	122
20.3. Этапы компьютерного моделирования	123
§ 21. Моделирование в задаче роста и убывания	124
21.1. Постановка задачи (этап 1)	—
21.2. Выбор плана создания модели (этап 2)	125
21.3. Создание документальной математической модели (этап 3а)	—
21.4. Создание компьютерной модели (этап 3б)	126
21.5. Исследование модели (этап 4)	—
21.6. Получение решения задачи (этап 5)	127
§ 22. Моделирование в задаче определения температурных режимов	—
22.1. Постановка задачи (этап 1)	—
22.2. Выбор плана создания модели (этап 2)	128
22.3. Создание документальной математической модели (этап 3а)	—
22.4. Создание компьютерной модели (этап 3б)	—

22.5. Исследование модели (этап 4)	130
22.6. Получение решения задачи (этап 5)	—
§ 23. Моделирование в задаче выбора положения железнодорожной станции	—
23.1. Постановка задачи (этап 1)	—
23.2. Выбор плана создания модели (этап 2)	131
23.3. Создание документальной математической модели (этап 3а)	—
23.4. Выбор метода решения математической задачи (этап 3б)	132
23.5. Создание компьютерной модели (этап 3в)	—
23.6. Исследование модели (этап 4)	133
23.7. Получение решения задачи (этап 5)	—
§ 24. Моделирование в задаче полета тела, брошенного под углом к горизонту	134
24.1. Постановка задачи (этап 1)	—
24.2. Выбор плана создания модели (этап 2)	—
24.3. Создание документальной математической модели (этап 3а)	—
24.4. Создание компьютерной модели (этап 3б)	135
24.5. Исследование модели (этап 4)	136
24.6. Получение решения задачи (этап 5)	—
§ 25. Моделирование средствами текстового процессора	137
25.1. Моделирование бумажных документов	—
25.2. Компьютерные носители вместо бумаги	—
25.3. Решение логических задач	138
25.4. Постановка задачи (этап 1)	—
25.5. Выбор плана создания модели (этап 2)	139
25.6. Создание компьютерной модели (этап 3)	—
25.7. Исследование модели (этап 4)	—
25.8. Получение решения задачи (этап 5)	—
§ 26. Знакомство с 3D-редактором	141
26.1. Понятие о 3D-графике	—
26.2. Запуск и выбор шаблона	142
26.3. Перемещение камеры	—
26.4. Простейшие объекты	143
26.5. Преобразования объектов	144
26.6. Текстуры и материалы	145
§ 27. Моделирование в задачах проектирования строений	148
27.1. Постановка задачи (этап 1)	—
27.2. Выбор плана создания модели (этап 2)	149
27.3. Создание компьютерной модели (этап 3б)	—
27.4. Исследование модели и получение решения (этапы 4 и 5)	151
§ 28. Моделирование движения мотоцикла	152
28.1. Постановка задачи (этап 1)	—
28.2. Выбор плана создания модели (этап 2)	—
28.3. Изучение готовой компьютерной модели (этап 3а)	153
28.4. Настройка готовой компьютерной модели (этап 3б)	—
28.5. Исследование модели (этап 4)	154
28.6. Получение решения задачи (этап 5)	—
Приложения	155

От авторов

Дорогие девятиклассники! Вы держите в руках учебное пособие по информатике. Эта наука становится важнейшей составляющей всей системы научного познания и определяет пути формирования информационного общества, основанного на знаниях.

Ускорение темпов роста объемов информации привело к необходимости искать новые пути для хранения и обработки информации и способствовало появлению и развитию облачных технологий. В начале XXI в. владение компьютерными информационными технологиями стало таким же важным элементом общей культуры современного человека, как умение грамотно писать и правильно излагать свои мысли. Методы информационного моделирования открывают новые возможности для изучения природы, человеческого общества и др.

Мы, авторы учебного пособия, постарались сделать так, чтобы вы смогли освоить необходимые современному человеку знания и получить соответствующие навыки. Надеемся, что это поможет вам в решении различных практических задач.

Материал параграфов учебного пособия разделен на две колонки. Цвет фона поможет вам разобраться в назначении размещенной на нем информации:



— основные материалы, обязательные для изучения;



— примеры, иллюстрирующие основные материалы;



— определения основных понятий;



— исторические сведения, информация об ученых, внесших вклад в развитие информатики, и другие интересные факты.

В учебном пособии используются следующие условные обозначения:



— вопросы и задания для проверки знаний;



— раздел «Упражнения» содержит задания, при выполнении которых используется компьютер;



— раздел «Упражнения» содержит задания для выполнения в тетради;



— раздел «Упражнения» содержит задания, при выполнении которых может быть использована информация, размещенная на Национальном образовательном портале;

* — упражнение или пример для любознательных.

В тексте некоторых упражнений вам будет предложено открыть файл. Это означает, что упражнение можно выполнить, используя файл, размещенный на Национальном образовательном портале <http://e-vedy.adu.by> («Электронное обучение» → «Электронные образовательные ресурсы» → «Информатика» → «Информатика. 9 класс»). Зайти на портал и скачать файлы к упражнениям можно также с помощью матричного QR-кода:



Имя файла для скачивания содержит номер параграфа и номер упражнения. Например, имя файла `upr4_5` означает, что файл относится к пятому упражнению четвертого параграфа. Также на портале размещены файлы с программами, рассмотренными в примерах. Такие файлы имеют имя `pr6_7.pas` (программа для примера 6.7).

В учебном пособии размещено много иллюстративного материала. Экранные копии предназначены для ознакомления с интерфейсами программ, для указания расположения отдельных элементов. Подробно рассмотреть все структурные элементы окна используемой программы можно на экране компьютера.

Данное учебное пособие имеет электронное приложение, размещенное на Национальном образовательном портале.

Глава 1

ИНФОРМАЦИОННЫЕ РЕСУРСЫ СЕТИ ИНТЕРНЕТ

§ 1. Организация сети Интернет

В наше время Интернет успел прочно войти в жизнь почти каждого человека. Он позволяет произвести расчеты за покупки или коммунальные услуги, не выходя из дома, и др.

В некоторых странах, например в Германии, законодательно закреплено право человека на пользование Интернетом.

Интернет остался бы компьютерной сетью для специалистов, если бы не два значимых изобретения, ставших востребованными среди массового пользователя, — электронная почта и World Wide Web.

История WWW началась в 1989 г., когда Тимоти Джон Бернерс-Ли предложил систему гипертекста.



Т. Д. Бернерс-Ли разработал первый веб-сервер, первый веб-браузер, протокол HTTP и концепцию WWW. Основные положения концепции:

- возможность редактирования информации так же важна, как и ее использование;
- компьютеры могут быть использованы для совместной работы;
- принцип «паутины», а не иерархии;
- ученые в IT-сфере несут не только техническую, но и моральную ответственность.

1.1. Организация службы WWW.

Понятие о веб-сервере

Интернет — всемирная система объединенных компьютерных сетей. Служба сети Интернет, представляющая собой всемирное информационное хранилище, называется World Wide Web (Всемирная паутина), сокращенно — WWW. Объем содержащейся в этом хранилище информации настолько велик, что его трудно оценить.

Всемирная паутина представляет собой: 1) информационные ресурсы, объединенные в единое целое; 2) компьютерные ресурсы, размещенные по всему миру.

WWW имеет структуру, в которой информационные ресурсы связаны посредством ссылок. Всемирная паутина позволяет передавать информацию в виде веб-страниц. Для просмотра содержимого веб-страниц и переходов по гиперссылкам предназначены специальные программы — **браузеры**.

Взаимодействие компьютеров в сети определяется специальными правилами — **протоколами**. Передача данных в сети Интернет описывается набором протоколов, основной из которых — TCP/IP (Transmission Control Protocol/Internet Protocol).

Система WWW построена на протоколе передачи гипертекста — HTTP (HyperText Transfer Protocol). Протокол HTTP предполагает использование клиент-серверной структуры передачи

данных. Задача, которая решается с помощью данного протокола, — обмен данными между клиентским приложением (обычно браузером) и веб-сервером. Расширение протокола HTTP с возможностью шифрования в целях безопасности называется HTTPS.

Веб-сервер — сервер, обеспечивающий обмен данными по протоколу HTTP.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором оно установлено.

Браузер формирует запрос и отправляет его на сервер, после чего веб-сервер обрабатывает данный запрос, формирует ответ и передает его обратно клиенту.

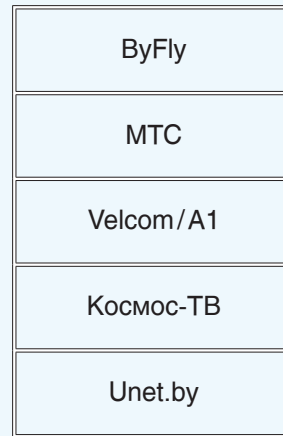
Для подключения клиентского компьютера к Интернету нужно воспользоваться услугами специальной организации — **провайдера** (пример 1.1).

1.2. Адресация в сети Интернет

В Интернете существует единая система адресации, обеспечивающая доступ к интернет-ресурсам. Компьютеры в сети могут иметь постоянные (статические) адреса или получать новый (динамический) адрес при каждом сеансе связи. Компьютеру присваивается уникальный 32-битный адрес из четырех частей, разделенных точками, — **IP-адрес**. Каждая часть представляет собой десятичное число в диапазоне от 0 до 255 (пример 1.2).

Наряду с IP-адресацией существует **система доменных имен** (DNS — Domain Name System) (пример 1.3).

Пример 1.1. Провайдеры, предоставляющие доступ в Интернет в Республике Беларусь.



Возможное количество IP-адресов можно рассчитать по формуле:

$$2^{32} - 1 = 4\,294\,967\,295.$$

Чтобы узнать IP-адрес своего компьютера, достаточно выполнить команду **Пуск → Все программы → Стандартные → Командная строка** и в появившемся окне ввести `ipconfig`.

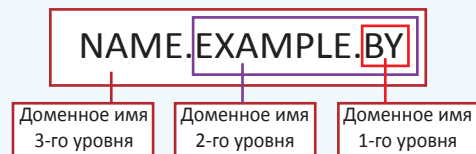
Динамический IP-адрес (назначается автоматически при подключении к сети) компьютера можно узнать, набрав `http://2ip.ru` в адресной строке браузера.

Пример 1.2. IP-адреса.

128.250.33.199

46.53.211.93

Пример 1.3. Структура доменного имени.



Пример 1.4. Домены первого уровня.
Географические

Домен	Страна
by, бел	Беларусь
ru, рф	Россия
de	Германия
uk	Англия
pl	Польша

Административные

Домен	Тип организации
com	Коммерческая
org	Некоммерческая
edu	Образовательная
net	Компьютерная сеть

Пример 1.5. Доменные имена.

nchtdm.by[93.125.99.47] — Национальный центр художественного творчества детей и молодежи.

frpmi.bsu.by[217.21.43.3] — факультет прикладной математики и информатики Белорусского государственного университета.

Чтобы узнать IP-адрес по доменному имени, нужно в окне **Командная строка** ввести nslookup и через пробел — доменное имя.

Пример 1.6. URL-адрес.

<http://luceum.by/komanda/direktor.html>

Протокол Доменное имя Путь Файл

Часто вместо «доменное имя» говорят «домен» (лат. *dominium* — владение).

Доменное имя можно расшифровать, читая его справа налево.

Доменные имена первого уровня обычно отражают принадлежность ресурса. Они могут быть:

- географическими — каждой стране соответствует свой буквенный код;
- административными (трехбуквенными) — позволяют определить род деятельности организации — владельца доменного имени.

(Рассмотрите пример 1.4.)

Записанные далее одно или несколько имен составляют доменные имена второго, иногда третьего и более низких уровней (обычно их 2—4).

Доменное имя второго уровня выбирается организацией и несет информацию о ее названии или услугах, имя домена третьего уровня может означать подразделение этой организации (пример 1.5).

Для доступа к информационным ресурсам используется своя система адресации. Это URL-адрес ресурса (Universal Resource Locator — универсальный указатель ресурса). В URL-адресе можно выделить составляющие (пример 1.6).



1. Что представляет собой WWW?
2. Какую структуру имеет WWW?
3. Что такое протокол?
4. Какой протокол позволяет передавать по сети гипертекст?
5. Что такое веб-сервер?
6. Что представляет собой IP-адрес?
7. Какую структуру имеет доменное имя?
8. Что такое URL-адрес? Для чего он предназначен?



Упражнения

1 Выберите из списка IP-адреса. Запишите их в тетрадь. Объясните, почему остальные записи не могут быть IP-адресами.

1. 345.132.216.111.

4. 189.27.1.256.

2. 111.234.67.1.78.

5. 132.45.108.255.

3. 101.254.231.13.

2 Выберите из списка доменные имена и URL-адреса. Запишите в тетради в две колонки.

1. 101.201.211.11.

4. fir.bsu.by.

2. bortik@tut.by.

5. https://www.google.com/.

3. obrazovanie.by.

6. informatika7.adu.by.

3 Алексей записал IP-адрес своей школы на листке бумаги и положил его в карман джинсов. Мама Алексея случайно постирала джинсы вместе с запиской. После стирки Алексей обнаружил в кармане четыре фрагмента. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

А	Б	В	Г
2.162	4.18	.61	20

4 Выпишите доменное имя из URL-адреса: <http://www.obelarus.net/buls/2016/mm.html>.

5 Восстановите из отдельных частей URL-адрес и запишите его в тетрадь.

catalog/	www	mazurka	.
midi.	by/	.mid	

§ 2. Национальные информационные ресурсы

В настоящее время информационные ресурсы приравниваются к традиционным (материальным, природным и т. д.). Рост количества информационных ресурсов способствует успешному переходу страны к развитому информационному обществу, а обладание сведениями о них является залогом успеха.

Каждый год у нас в стране выходит каталог «Информационные ресурсы и

Наиболее полной универсальной коллекцией национальных и зарубежных документов в нашей стране располагает Национальная библиотека Беларуси. Библиотека формирует национальные электронные информационные ресурсы, предоставляет пользователям доступ к мировым электронным информационным ресурсам ведущих производителей.

Пример 2.1. Национальные информационные ресурсы.

1. <http://www.belarus.by>



2. <http://redbook.minpriroda.gov.by>



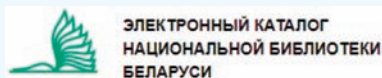
3. <http://edu.gov.by>



4. <http://mir.pravo.by>



5. <http://e-catalog.nlb.by>



6. <http://www.gismeteo.by>



7. <http://museum.by>



8. <http://yandex.by/maps>



системы Беларуси», который постоянно пополняется. Перечислим некоторые из ресурсов каталога:

1. Официальный сайт Республики Беларусь.

2. Красная книга Республики Беларусь (биоинформационная система).

3. Официальный сайт Министерства образования Республики Беларусь (нормативно-правовые документы).

4. Детский правовой сайт (с игровой и информационной составляющей).

5. Электронный каталог Национальной библиотеки Республики Беларусь.

6. Gismeteo (текущая погода и точный прогноз).

7. Музеи Беларуси.

8. Поисково-информационная картографическая служба Яндекса (поиск по карте, информация о пробках, прокладывание маршрутов, панорамы улиц городов).

9. Яндекс Транспорт — мобильное приложение, содержащее данные общественного транспорта более 80 городов. Покажет, как доехать до пункта назначения, где ближайшая остановка, какой автобус довезет до нужного места и где он сейчас едет; сообщит, когда пора готовиться к выходу.

(Рассмотрите пример 2.1.)

Информационные образовательные ресурсы — учебные и исследовательские материалы на любом носителе, цифровом или ином, который находится в общественном достоянии.

Термин *информационные образовательные ресурсы* был впервые введен в 2002 г. на Форуме ЮНЕСКО.

Информационные образовательные ресурсы включают:

- полные учебные курсы;
- учебные материалы;
- модули;
- учебные пособия;
- тесты;
- программное обеспечение и любые другие инструменты или материалы, предоставляющие доступ к знаниям.

В Интернете размещено огромное количество информационных образовательных ресурсов. Доступ к ним обеспечивает пользователей основным и дополнительным учебным материалом, необходимым для обучения в школе, выполнения заданий учителя и самостоятельного обучения.

Приведем примеры образовательных ресурсов:

1. Национальный институт образования — дистанционное обучение (файлы к примерам и упражнениям учебных пособий и другие учебные материалы).

2. ЮНИ-центр-XXI (научно-исследовательский и методический центр преподавателей и учащихся на базе БГУ).

3. Белорусская ассоциация «Конкурс» (интеллектуальные игры-конкурсы по предметам).

4. «Наука — детям» (эксперименты, задания, интеллектуальные игры).

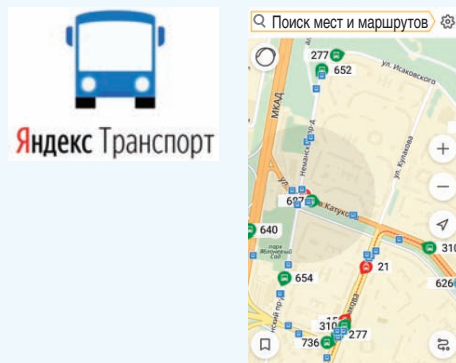
5. Энциклопедия «Хочу все знать» (интересные факты из различных предметных областей).

6. Свободная энциклопедия «Википедия» (энциклопедия, которую может редактировать каждый).

(Рассмотрите пример 2.2.)

Пример 2.1. Продолжение.

9. <https://mobile.yandex.ru/apps/android/transport>



Пример 2.2. Образовательные ресурсы.

1. <http://e-vedy.adu.by>



2. <http://www.uni.bsu.by>

ЮНИ-центр-XXI

3. <http://www.bakonkurs.by>



4. <http://virtuallab.by>



5. <http://ya-uznay.ru>



6. <http://ru.wikipedia.org/wiki>





1. Какое доменное имя указывает на принадлежность информационных ресурсов нашей стране?
2. Какой информационный ресурс позволит увидеть карту вашего населенного пункта?
3. В каком случае информационные ресурсы могут считаться образовательными?
4. Какие образовательные ресурсы вы использовали ранее?



Упражнения

- 1 На официальном сайте Республики Беларусь в разделе «География» найдите информацию о количестве населения региона, в котором живете. Запишите найденную информацию в тетрадь.
- 2 В биоинформационной системе «Красная книга Республики Беларусь» найдите информацию о журавлеобразных птицах. Выпишите названия птиц в тетрадь.
- 3 На детском правовом сайте из раздела «Вопросы и ответы» скопируйте информацию в текстовый файл. Файл сохраните в папку, указанную учителем.
- 4 Откройте поисково-информационную картографическую службу Яндекс. Выполните следующие задания:
 1. В строке «Поиск мест и адресов» введите «Минск, площадь Победы».
 2. Нажмите на фото площади под строкой поиска. Откроется 3D-панорама.
 3. С помощью направляющих стрелок «пройдите по кругу».
 4. Найдите ближайший к вашему месту жительства город и «побродите» по его улицам.
- 5 Зарегистрируйтесь на портале дистанционного обучения Национального института образования (<http://e-vedy.edu.by>). Перейдите в раздел «Электронное обучение» → «Электронные образовательные ресурсы» → «Информатика. 9 класс». Сохраните файлы раздела в папку, указанную учителем.
- 6 Откройте энциклопедию «Хочу все знать». Перейдите в раздел «Занимательные факты». Найдите ответ на вопрос «Почему бензин нельзя тушить водой?».

§ 3. Поиск в сети Интернет

Пример 3.1. Способы поиска в сети Интернет.

По прямой
ссылке

- самый быстрый
- используется, если известен адрес страницы

Интернет-
серфинг

- используется только в пределах одной предметной области

С помощью
поисковых систем

- наиболее удобный

Существуют три основных способа поиска информации в сети Интернет:

1. По прямой ссылке (URL-адресу страницы).

2. Интернет-серфинг (передвижение по гиперссылкам).

3. С помощью поисковых систем. (Рассмотрите пример 3.1.)

Чтобы получить доступ к ресурсу, достаточно ввести его URL-адрес

(если он известен) в адресную строку любого браузера (пример 3.2).

Второй способ поиска заключается в последовательном переходе по ссылкам с одной веб-страницы на другую до тех пор, пока не будет найдена необходимая информация. Преимущество такого способа — в его увлекательности и возможности освоить большой объем информации с разных сайтов. Недостаток — его длительность.

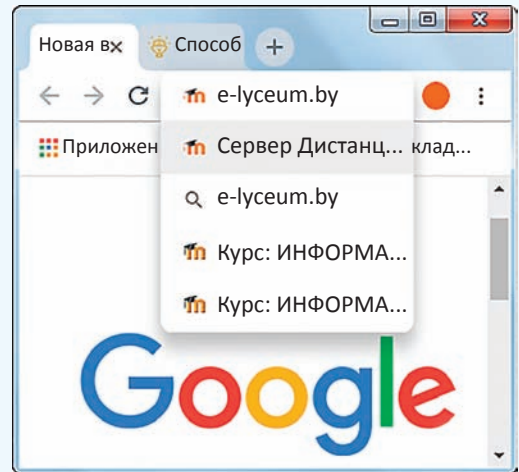
Во Всемирной паутине веб-страницы связаны только гиперссылками, т. е. четкой структуры нет. Единственным указателем на веб-страницу является ее URL-адрес. Тем не менее в случае использования поисковых систем поиск производится по содержанию. Как он осуществляется?

Современные поисковые системы предоставляют пользователю две возможности поиска информации: в поисковых каталогах и с помощью запросов (по ключевым словам).

Поисковые каталоги помогают искать информацию определенной тематики. Информация в таких каталогах структурирована по группам и темам. В каждой теме пользователю предлагаются ссылки на ресурсы, где можно найти нужные сведения (пример 3.3).

Достоинство каталогов в том, что все ресурсы проверяются вручную. Это обеспечивает высокое качество подборки ресурсов и исключает ссылки на случайные ресурсы, не имеющие отношения к теме поиска. К недостаткам относится небольшой охват ресурсов, а также то, что обновление каталогов происходит достаточно медленно, т. е. возможна потеря актуальности информации.

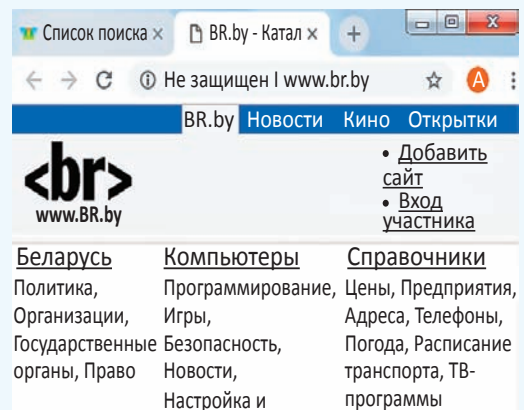
Пример 3.2. Поиск по URL-адресу.



Если текста на веб-странице много, а вас интересует только несколько строчек, то можно воспользоваться поиском на странице. Нажмите **Ctrl + F**, а в поле **Найти** наберите требующуюся вам фразу или слово, после чего нажмите **Enter**.

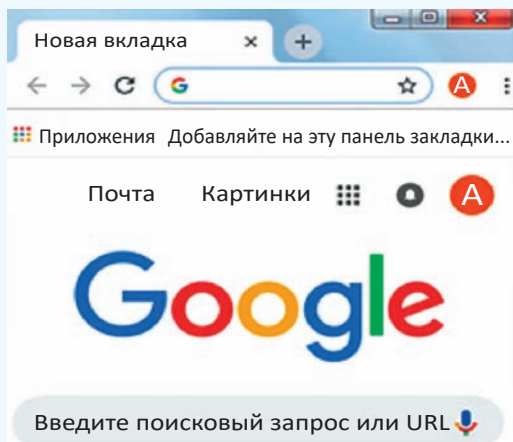
Термин *интернет-серфинг* предложил в 1992 г. американский библиотекарь Джин Армор Полли.

Пример 3.3. Поисковый каталог.

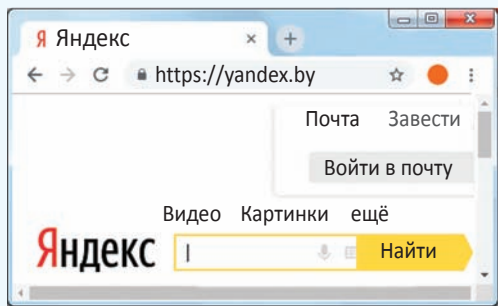


Пример 3.4. Поисковые системы.

<http://google.com>



<http://yandex.by>



Наименование «Google» происходит от слова «гугол» — так в математике называется число из единицы со ста последующими нулями (10 в сотой степени).

Интересное название имеет адресная строка браузера Google Chrome — Omnibox. Она выполняет автодополнение: можно начать вводить часть адреса часто посещаемого сайта, и возникнет подсказка в виде URL-адреса.

Каждый день Google обрабатывает более 3 млрд поисковых запросов. При этом примерно 15 % из этого числа — абсолютно новые, т. е. никогда не встречавшиеся в поисковой системе Google ранее.

Поисковые системы — сайты, в которых при введении запроса в строку поиска пользователь получает ряд ссылок на страницы в Интернете. В поисковых системах работает специальный поисковый робот («паук»). Робот постоянно обходит сеть и заносит в специальную базу данных (индексирует) информацию, собранную с веб-страниц. После получения запроса поиск ведется в базе данных среди индексированных сайтов.

Использование поисковых систем — наиболее удобный способ поиска информации. К известным поисковым системам относятся:

- Google — быстрая и самая большая поисковая система; определяет рейтинг ресурса по количеству ссылок, ведущих к нему;
- Яндекс — мощная поисковая система, производящая анализ информации с учетом особенностей русского языка.

(Рассмотрите пример 3.4.)

Эффективность поиска обычно оценивается по двум параметрам — полноте и точности.

Полнота поиска — отношение числа выданных релевантных ресурсов к общему числу релевантных ресурсов, имеющих в распоряжении поисковой машины. (Информация, соответствующая информационным потребностям, называется релевантной.)

Точность поиска — отношение числа выданных релевантных ресурсов к общему числу выданных ресурсов.

Как бы точно ни были составлены запросы, почти всегда любая поис-

ковая система выдаст список ссылок, где будут ресурсы, не соответствующие информационным потребностям.

Достаточно часто возникает необходимость сохранить найденную в Интернете информацию. В зависимости от потребности можно сохранить всю веб-страницу либо только текст или изображения.

Для сохранения веб-страницы:

1. В контекстном меню веб-страницы выбрать **Сохранить как...**

2. В окне сохранения выбрать папку, куда хотите сохранить страницу, и определить выбор в поле **Тип файла** (пример 3.5).

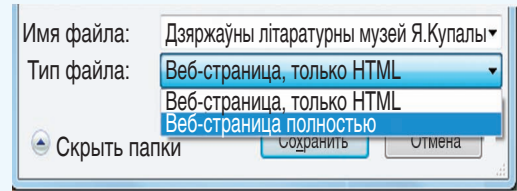
3. Изменить имя файла, если оно слишком длинное.

Только текст (или его фрагмент), расположенный на веб-странице, можно сохранить в текстовый файл с помощью буфера обмена.

Для сохранения изображения (или его копирования в буфер обмена) можно воспользоваться соответствующей командой контекстного меню изображения (пример 3.6).

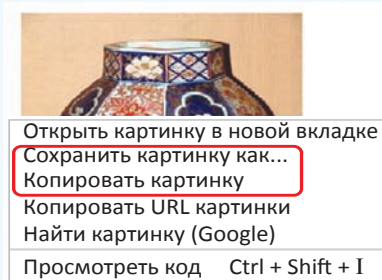
У современных поисковых систем существует еще одна возможность поиска — поиск по картинке. В этом случае достаточно в окне браузера выбрать **Картинки** и просто перетащить файл с изображением в строку поиска браузера или загрузить его с помощью значка  (пример 3.7).

Пример 3.5. Выбор формата сохранения веб-страницы.

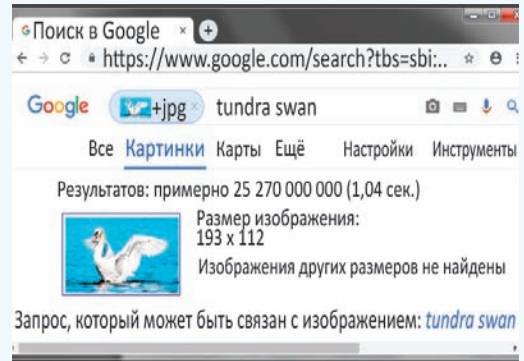


В случае выбора **Веб-страница полностью** изображения, расположенные на странице, будут сохраняться в отдельной папке.

Пример 3.6. Сохранение изображения.



Пример 3.7. Поиск по картинке.



1. Какие способы можно использовать для поиска информации в сети Интернет?
2. Какой способ поиска информации в сети Интернет самый быстрый?
3. Что такое интернет-серфинг?
4. Как организован поиск информации в поисковых каталогах?
5. Каким образом осуществляют поиск информации поисковые системы?
6. Какие поисковые системы вам известны?



Упражнения

- 1 По URL-адресу <http://museum.by> найдите сайт «Музеи Беларуси». С помощью ссылок сайта найдите информацию об одном из музеев вашего региона. Сохраните главную страницу сайта музея (полностью).
- 2 Откройте ресурс свободной энциклопедии «Википедия» (<http://ru.wikipedia.org/wiki>). Перейдите по ссылке в Викицитатник. Перемещаясь по ссылкам, найдите английскую поговорку, русский аналог которой «Дорогу осилит идущий». Сохраните, используя буфер обмена, поговорку и ее аналоги в виде текстового файла.
- 3 Найдите в сети Интернет информацию о количестве пользователей наиболее известных поисковых систем. Составьте таблицу для сравнения результатов.
- 4 Текст параграфа 3 размещен в электронной версии учебного пособия на портале дистанционного обучения Национального института образования. Какие запросы можно составить, чтобы найти его? Выполните следующие задания:
 1. Запишите в тетрадь не менее трех таких запросов.
 2. Воспользовавшись составленными запросами, осуществите поиск с помощью различных поисковых машин.
 3. Сравните результаты поиска. Какие поисковые системы и запросы были эффективнее?
- 5 С помощью поисковой системы найдите изображения оперных театров мира. Сохраните изображения. Создайте текстовый файл, содержащий таблицу с изображениями и текстовыми пояснениями к ним, а также с указанием места нахождения театров.

Фото	Город
------	-------

- 6 Используя поиск по картинкам, определите по изображениям известных ученых-физиков их имена (файлы с изображениями можно найти на портале дистанционного обучения Национального института образования). Запишите в тетради соответствие (номер файла — имя ученого).

§ 4. Облачные технологии. Сервисы совместной работы

Возможно, благодаря применению облачных технологий в будущем мы будем использовать компьютеры, которые по мощности близки к первым их образцам. Такие компьютеры будут представлять собой один лишь экран с микропроцессором, а все расчеты и мощности станут удаленными, т. е. в виртуальных серверах, а именно в облаке.

Одним из перспективных направлений развития современных информационных технологий являются облачные технологии.

Облачные технологии — технологии обработки данных, в которых компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервис.

С ресурсами облачных технологий можно познакомиться в примере 4.1.

Такие технологии основаны на хранении и обработке данных в так называемом облаке — рабочем пространстве в сети Интернет. Пользователь получает в свое распоряжение пространство для хранения данных и (или) программное обеспечение для их обработки.

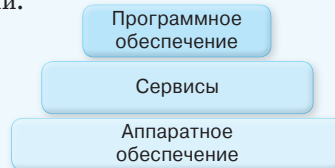
Для того чтобы ваши данные были доступны вам (или вашим друзьям) на любом компьютере (где бы он ни находился), в том числе и на мобильном устройстве, используются облачные хранилища данных. Этот сервис является базовым для других облачных сервисов, поскольку входит в состав почти каждого из них.

Облачное хранилище данных — интернет-хранилище, в котором данные содержатся на многочисленных серверах, распределенных в сети, предоставляемых в пользование клиентам. Некоторые из наиболее популярных сервисов облачных хранилищ данных:

1. Google Drive (<https://drive.google.com>).
2. Mega (<https://mega.nz>).
3. Яндекс.Диск (<https://disk.yandex.by>).
4. Amazon Web Services (<https://aws.amazon.com/ru>). (Рассмотрите пример 4.2.)

Самый известный из облачных сервисов — электронная почта. Пользуется популярностью и разнообразное удаленное (размещенное онлайн) программное обеспечение. Это не только различные приложения (пример 4.3),

Пример 4.1. Ресурсы облачных технологий.



Пример 4.2. Облачные хранилища данных.

Сервис	Описание
Google Drive	Можно хранить 30 типов файлов (15 Гбайт — бесплатно)
MEGA THE PRIVACY COMPANY	Защищает данные за счет шифрования (50 Гбайт — бесплатно в базовом тарифе)
Яндекс.Диск	Облачный сервис (до 10 Гбайт бесплатно, бонусная система)
amazon web services	Широкий спектр облачных сервисов (5 Гбайт в период тестирования — 12 месяцев)

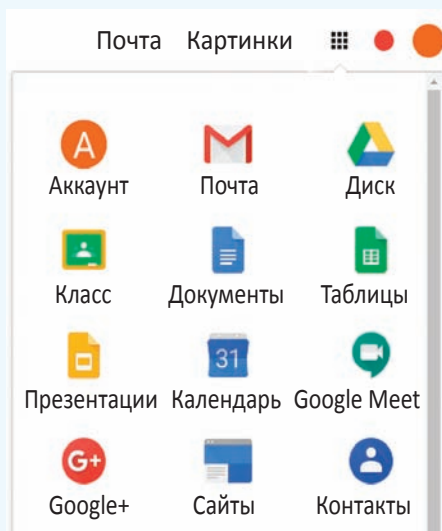
Пример 4.3. Облачные приложения.

Приложение	Описание
Prezi	Создание презентаций (http://prezi.com)
Conceptboard	Виртуальная доска для совместной работы (https://conceptboard.com)
Calc.by Онлайн-калькуляторы	Разнообразные калькуляторы (http://calc.by)
umath.ru	Построение графиков функций, математические вычисления (https://umath.ru)
Calendarum.ru	Создание визиток, календарей, конвертов (http://calendarum.ru)

Пример 4.4. Облачные операционные системы.

Zero PC (http://zero.com)
CloudMe (http://cloudme.com)
AstraNOS (http://astranos.com)

Пример 4.5. Доступ к приложениям Google.



В рамках акции «Беларусь помнит» работает онлайн-сервис видеоткрыток <https://victory.sb.by/>.

Любой пользователь может загрузить фото участника Великой Отечественной войны, его биографию или другую информацию и мгновенно получить видеоролик о данном человеке. Видеоткрытку можно и скачать, и поделиться ею в соцсетях.

но и операционные системы (пример 4.4).

Использование онлайн-приложений расширяет возможности работы с документами.

Часто при работе над каким-либо проектом участники могут находиться далеко друг от друга (в разных городах и даже странах). Это вызывает определенные сложности при необходимости обсуждения и согласования документов. Конечно, существует немало способов быстрого обмена документами в Интернете — можно отправлять файлы по электронной почте, размещать в облачных хранилищах и др. Все эти варианты предполагают, что с одним документом одновременно работает только один пользователь. Такая работа не всегда эффективна, ведь приходится многократно согласовывать содержание текста. Согласование изменений в документах может быть организовано более оперативно — через облачные сервисы, предназначенные для совместной работы над документами, например Google Docs. Доступ к сервисам Google Docs можно получить прямо из окна браузера (пример 4.5).

Интерфейс сервисов Google Docs **Документы** и **Презентации** во многом похож на интерфейс уже известных вам программ Word и PowerPoint (см. Приложение 1, с. 155—157).

Принцип работы таких сервисов следующий: документы хранятся не на локальных компьютерах, а в облаке и доступны только тем пользователям, для которых авторами документов были установлены соот-

ветствующие права (на чтение либо редактирование). Возможности облачного сервиса позволяют работать с документами нескольким пользователям одновременно на любом устройстве с доступом в Интернет.

При использовании сервисов совместной работы исключается какая-либо путаница с разными версиями одного и того же документа, ведь всем пользователям доступен один и тот же вариант этого документа.

Чтобы пользоваться облачными сервисами, необходимо создать **аккаунт** — учетную запись, в которой хранится персональная информация. Аккаунт может быть привязан к адресу электронной почты. Например, при наличии электронной почты на ресурсе Google можно использовать другие сервисы этого ресурса (пример 4.6).

Возможности облачных технологий используются и в борьбе с компьютерными вирусами. Антивирус Panda Cloud автоматически выявляет новые угрозы, используя вычислительную мощь облачных технологий (пример 4.7).

Преимущества облачных технологий:

1. Не требуют больших вычислительных мощностей компьютера.
2. Обеспечивают высокий уровень надежности.
3. Экономически эффективны.

Недостатки облачных технологий:

1. Зависимость от компании, предоставляющей сервис.
2. Необходимость наличия Интернета.
3. Необходимость в дополнительных мерах по обеспечению безопасности.

Пример 4.6. Некоторые из сервисов Google.

Сервис	Назначение
 Документы	Работа с текстовыми документами, аналог MS Word
 Презентации	Работа с презентациями, аналог MS PowerPoint
 Календарь	Планирование встреч и событий. Позволяет задать время встречи, создать повторяющиеся мероприятия, установить напоминания и др.
 Контакты	Синхронизация контактов на всех устройствах, хранение и систематизация контактной информации. Контакты могут содержать имена, адреса электронной почты и др.
 Фото	Хранение, редактирование, обмен фотографиями и видео
 Формы	Создание анкет, викторин, тестов
 Google Keep	Создание и хранение заметок

Пример 4.7. Антивирус Panda Cloud.



Распространяется по принципу «Программное обеспечение + Сервис», т. е. пользователь устанавливает на свой компьютер программу, а часть работы антивируса происходит на серверах Panda Security.



1. Какие технологии называются облачными?
2. Какой облачный сервис является базовым?
3. В чем суть совместной работы над документами? Какие сервисы предоставляют такие возможности?
4. В каком случае пользователь получает доступ к облачным сервисам?
5. Какие облачные сервисы вы используете?

**Упражнения¹**

1 Сохраните результат выполнения упражнения 1 после § 3 на Google Drive в папке «Музеи» и откройте учителю доступ к ней. Для этого выполните следующие действия:

1. Загрузите браузер Google Chrome.
2. С помощью кнопки  разверните список сервисов Google и выберите в нем .
Диск
3. Войдите в аккаунт.
4. Выберите папку  Мой диск.
5. Перетащите файл с результатом выполнения упражнения 1 после § 3 из папки на вашем компьютере в папку «Мой диск».
6. В контекстном меню файла выберите  Совместный доступ...
7. Введите электронный адрес, указанный учителем:

Совместный доступ
Включить доступ по ссылке 

Люди

Введите имена или адреса эл. почты...


Готово

Расширенные

2 Найдите в Интернете сервисы, позволяющие строить графики. С помощью выбранного сервиса постройте графики перечисленных функций.

1. $y = -0,3x^2 + 4x - 0,5$.

3. $y = 4x^3 + 1$.

2. $y = 5\sqrt{3x} - 2$.

4. $y = \frac{0,8}{x + 0,5} + 3$.

После каждого построения создайте ссылку на график. Сохраните все ссылки в текстовом документе.

¹ Перед выполнением упражнений 1—4 после § 4 рекомендуется создать аккаунт в Google.

- 3 Найдите в Интернете сервисы, позволяющие создавать персональные визитки. Создайте свою визитку с помощью выбранного сервиса.
- 4 Создайте в Google **Фото** альбом «Фауна Южной Америки». Перетащите в альбом ( → **Создать альбом**) изображения в соответствии с содержимым файла *upr_4_5.docx*.
- 5 В Google Docs **Документы** создайте совместный проект «Фауна Южной Америки». Каждый участник проекта добавляет информацию в файл *upr_4_5.docx* по образцу:

Ягуар 	Ягуары – хищные млекопитающие семейства кошачьих. Их изображения встречаются на руинах вдоль всего полуострова Юкатан, где они были главной фигурой в религиозных обрядах. Ягуары охотятся днем и ночью. Сегодня эти животные по-прежнему считаются символом власти, силы, красоты и интеллекта.
---	---

Для вставки текста воспользуйтесь буфером обмена. Для вставки изображений выберите кнопку  →  **Добавить из Google Фото**. Используйте изображения из альбома «Фауна Южной Америки», созданного в упражнении 4.

- 6 Используя сервис Google Docs **Презентации**, создайте совместный проект «Асобы ў гісторыі Беларусі». Каждый участник проекта добавляет информацию в файл *upr_4_6.pptx* по образцу:

Еўфрасіння Полацкая Каля 1101–1167 (?) Дачка полацкага князя, інакіня і асветніца перыяду Полацкага княства. Стала манашкай у 12 гадоў і правяла жыццё, дапамагаючы бедным. Пабудавала на свае сродкі дзве царквы ў Полацку, заснавала пад Полацкам жаночы і мужчынскі манастыры, якія сталі цэнтрамі асветы ў Полацкім княстве (там працавалі вучыльні, бібліятэкі, скрыпторый, багадзельня, іканалісная і ювелірная майстэрні). Памерла ў час паломніцтва ў Іерусалім. Яе мошчы перададзены ў Беларусь у 1910 годзе. Прылічана да ліку святых, шануецца як апякунка, нябесная заступніца зямлі беларускай.	
--	--

Информацию о личностях найдите самостоятельно, воспользовавшись поиском в Интернете (при необходимости переведите текст на белорусский язык). Для вставки изображений выберите кнопку  Найдти в Интернете.

7* С помощью сервиса Google **Формы** создайте анкету на одну из перечисленных тем.

1. Здоровый образ жизни.
2. Увлечения и спорт.
3. Моя будущая профессия.
4. Край, в котором я живу.

Разошлите анкету (кнопка **ОТПРАВИТЬ**) одноклассникам на электронную почту.

Примечание. Предполагается работа в группах.

§ 5. Сетевой этикет и меры безопасности в сети Интернет

В Интернете часто используют специальные слова (сленг).

1. Для обозначения нарушений правил сетевого этикета:

Флейм (англ. *flame* — пламя) — неожиданно возникшее активное обсуждение, при развитии которого участники обычно забывают о первоначальной теме, переходят на личности и не могут остановиться.

Флуд (англ. *flood* — наводнение) — сообщения, не несущие никакой полезной информации. Флуд может распространяться с целью троллинга, т. е. из желания кому-либо досадить. Технический флуд представляет собой хакерскую атаку с большим количеством запросов, приводящую к отказу работы сервиса (DDoS-атака).

Спам (англ. *spam*) — сообщения, присылаемые от неизвестных людей или организаций без разрешения. Часто термин *спам* употребляется в значении *почтовый спам* — рассылка электронных писем, содержащих рекламу.

Оффтоп (англ. *off topic* — вне темы) — сетевое сообщение, не имеющее отношения к заранее установленной теме общения. Наиболее неодобряемой формой оффтопа являются рекламные сообщения.

Интернет — мир интересных и полезных возможностей, но в то же время это источник угроз, особенно для детей и молодежи. Агрессия, мошенничество, психологическое давление — опасности, которые могут поджидать в глобальной сети каждый день.

В 6-м классе вы познакомились с сетевым этикетом при работе с электронной почтой, узнали, какие меры безопасности необходимо соблюдать, пользуясь электронным почтовым ящиком. Работая с различными облачными сервисами или общаясь по сети, нужно соблюдать такие же правила, как и при работе с электронной почтой.

Обезличенность при общении в Интернете заставляет пользователей забывать, что они имеют дело не с машиной, а с реальными людьми. Правила сетевого этикета помогают достичь взаимопонимания и обеспечивают безопасность общения. Основные из них:

1. Будьте вежливы и не забывайте об обязательных формулах приветствия, обращения, благодарности.

2. Избегайте бессодержательных бесед, чтобы не тратить свое время и время собеседника.

3. Пишите грамотно. Используйте проверку орфографии. Перед отправкой сообщения перечитайте текст.

4. Без необходимости не пишите на транслите (т. е. не злоупотребляйте использованием букв алфавита другого языка). Не набирайте текст заглавными буквами. Не перегружайте сообщение смайликами.

5. Сохраняйте анонимность при общении с незнакомцами.

Неосмотрительность и халатный подход к обеспечению безопасности в Интернете могут дать возможность преступникам совершить противоправные действия. Сначала преступник получает несанкционированный доступ к учетным записям в социальных сетях, к электронному почтовому ящику, к аккаунтам и др. Получив реквизиты, злоумышленник заходит в учетную запись и осуществляет рассылку контактам владельца взломанной учетной записи сообщения мошеннического характера.

Рекомендации, которые помогут снизить вероятность совершения противоправных действий в Интернете:

1. Для выхода в сеть Интернет используйте устройства, на которых установлены и постоянно обновляются антивирусные программы.

2. При посещении известных сайтов обращайте внимание на их внешний вид: возможно, это поддельная копия.

3. Вводите личную информацию только на веб-сайтах, которые работают с использованием защищенных

Хотлинк (англ. *hotlink*) — включение в веб-страницу файлов-изображений или других ресурсов с чужого сервера. Этот прием используется недобросовестными веб-мастерами. При этом расходуются чужие ресурсы и трафик.

Оверквотинг (англ. *overquoting*) — избыточное цитирование.

2. Для обозначения противоправных действий в Интернете:

Фишинг (англ. *phishing* — *password* + *fishing* — выуживание паролей) — вид мошенничества с целью получения доступа к логинам и паролям пользователей.

Киберсквоттинг (англ. *cybersquatting*) — регистрация доменных имен, содержащих торговую марку, принадлежащую другому лицу, с целью их дальнейшей перепродажи или недобросовестного использования.

Брутфорс (от англ. *brute force* — полный перебор) — метод атаки или взлома путем перебора всех возможных вариантов пароля.

Кардинг (от англ. *carding*) — вид мошенничества с использованием чужой платежной карты или ее реквизитов.

Кликджекинг (англ. *clickjacking*) — механизм обмана пользователей Интернета, позволяющий узнать контакты посетителей сайта еще до того, как они сами разместили их на сайте.

Руткит (англ. *rootkit*) — программа или набор программ для скрытия следов присутствия злоумышленника или вредоносной программы в системе.

Фарминг (англ. *pharming*) — скрытое перенаправление на ложный IP-адрес.

Примеры мошенничества в Интернете:

1. Производится вирусная атака на компьютерные устройства, блокируется браузер или операционная система, а на экране монитора появляется требование оплатить крупный штраф.

2. На электронный почтовый ящик приходит письмо, которое обещает: после покупки обучающего курса можно уже завтра начинать зарабатывать огромные деньги. Помните, что никто не будет рассказывать совершенно незнакомым людям, как заработать огромные деньги, мошенники просто зарабатывают деньги на продаже этих обучающих курсов.

Пример 5.1. Статьи Уголовного кодекса Республики Беларусь, определяющие ответственность за преступления в сети Интернет (<http://kodeksy.by>).

Статья 212. Хищение путем использования компьютерной техники.

Статья 349. Несанкционированный доступ к компьютерной информации.

Статья 350. Модификация компьютерной информации.

Статья 351. Компьютерный саботаж.

Статья 352. Неправомерное завладение компьютерной информацией.

Статья 353. Изготовление либо сбыт специальных средств для получения неправомерного доступа к компьютерной системе или сети.

Статья 354. Разработка, использование либо распространение вредоносных программ.

Статья 355. Нарушение правил эксплуатации компьютерной системы или сети.

Анализ 3 млн созданных людьми восьмизначных паролей показал, что буква «е» была использована в паролях 1,5 млн раз, в то время как буква «f» — только 250 000 раз.

Ресурс для проверки уникальности пароля: <https://exploit.in/passcheck>

протоколов (в браузере рядом с адресом такого сайта отображается значок замка).

4. Не используйте одинаковые логины и пароли на различных сайтах.

5. Не используйте легкие пароли (даты рождения, номера телефонов и т. д.).

6. Остерегайтесь неожиданных или необычных электронных сообщений, даже если вам знаком отправитель; не открывайте вложения и не переходите по ссылкам в таких сообщениях.

7. При поступлении сообщений от знакомых, содержащих просьбы о финансовых операциях или о передаче финансовых реквизитов, обязательно проверяйте данную информацию по другим каналам связи (личная встреча, телефонный звонок, голосовая связь). Постарайтесь установить личность собеседника с помощью контрольных вопросов, ответы на которые могут быть известны только вам двоим.

Законодательством Республики Беларусь определена мера ответственности за следующие преступления в сети Интернет:

1. Несанкционированный доступ к данным.

2. Модификация (изменение) данных без разрешения владельца.

3. Умышленное уничтожение данных, приведение их в непригодное состояние.

4. Разработка, использование и распространение вредоносных программ.

5. Нарушение авторского права.

(Рассмотрите пример 5.1.)



1. Почему нужно соблюдать правила этикета при общении в сети Интернет?
2. Какие правила сетевого этикета не соблюдаются в приведенных ниже сообщениях?

Сегодня четверг. Это хуже, чем суббота, но гораздо лучше, чем понедельник... Но немного хуже, чем пятница. Зато четверг лучше, чем среда. Четверг даже лучше, чем воскресенье, потому что в воскресенье завтра понедельник, а в четверг завтра пятница...

Смайлы 😊😊😊 — это очень удобно 🙌🙌🙌. Но в то же время у них есть обратная сторона 😞. Они берегут эту тайну 🤔🤔. И никто не должен знать о ней 🙄🙄. НИКТО! 🙄🙄🙄

нАриСуЙ Мне:

- 1) УвАжАеШЬ — рОзоВыЙ фОн;
- 2) ОБИдЕлся — КоричНевЫй фОн;
- 3) друЖишЬ — бЕлЫй фОн.

рАзошлИ Это Всем СвОим дРуЗьяМ, И тВоя СтЕнА бУдЕт СупеРКрАсИвая.

3. Какие меры необходимо предпринять, чтобы обезопасить свои учетные записи от действий мошенников?
4. Каким должен быть безопасный пароль?
5. Какие действия в сети Интернет определены законодательством Республики Беларусь как противоправные?



Упражнения

- 1 Приведите примеры, когда соблюдаются и не соблюдаются правила сетевого этикета. Сделайте вывод.
- 2 Выберите из таблицы небезопасные пароли.

sdfghjkl	password	qS8+njiPh	12345678	Yn2004
----------	----------	-----------	----------	--------

Создайте текстовый документ, содержащий пояснение вашего выбора для каждого пароля.

- 3 Подготовьте презентацию на тему «Сетевой этикет».
- 4 С помощью сервиса Google **Формы** создайте анкету на тему «Безопасность в сети Интернет».

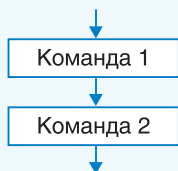
Глава 2

АЛГОРИТМЫ ОБРАБОТКИ СТРОКОВЫХ ВЕЛИЧИН

§ 6. Основные алгоритмические конструкции и типы данных

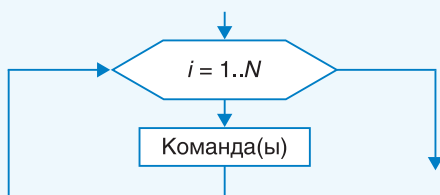
Пример 6.1. Блок-схемы алгоритмических конструкций.

1. Следование

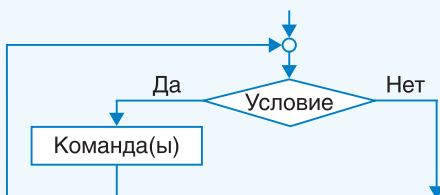


2. Цикл

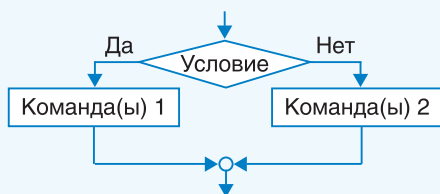
1) цикл с параметром



2) цикл с предусловием



3. Ветвление



6.1. Основные алгоритмические конструкции

Напомним некоторые определения, известные вам из курсов 7-го и 8-го классов.

Алгоритм — конечная последовательность команд, формальное выполнение которых позволяет получить решение задачи для любого допустимого набора исходных данных. Все команды делят на группы:

1. Команды, которые непосредственно выполняются в программе.

2. Команды, изменяющие порядок выполнения других команд.

Любой алгоритм может быть записан с использованием базовых алгоритмических конструкций, а именно: **следование**, **цикл** и **ветвление** (пример 6.1).

Программа представляет собой запись на некотором формальном языке — языке программирования. Командами в языке программирования считают:

- операторы (оператор присваивания, оператор ветвления, оператор цикла и др.);
- вызовы вспомогательных алгоритмов (встроенных в библиотеки и созданных пользователем).

Команды цикла и ветвления управляют порядком выполнения других команд в программе и относятся

к командам управления (управляющим конструкциям) (пример 6.2).

Оператор ветвления — команда, реализующая алгоритмическую конструкцию *ветвление* на языке программирования.

Оператор цикла — команда, реализующая алгоритмическую конструкцию *повторение* на языке программирования.

В Pascal существуют различные возможности управлять тем, сколько раз будет повторяться тело цикла. Может быть задано условие продолжения или окончания работы цикла, а также число повторений тела цикла.

Цикл с предусловием используется в том случае, когда известно условие продолжения работы.

Цикл с параметром используется тогда, когда известно количество повторений.

6.2. Вспомогательные алгоритмы

Вспомогательный алгоритм — алгоритм, который можно использовать в других алгоритмах, указав его имя и, если необходимо, значения параметров.

В языке Pascal используются вспомогательные алгоритмы двух видов: процедуры и функции. Они могут быть с параметрами или без параметров (пример 6.3).

Описание процедур и функций повторяет структуру программы на языке Pascal. Оно может содержать раздел **var** для описания переменных, которые используются только

Пример 6.2. Запись операторов ветвления и цикла на языке Pascal.

Ветвление в полной форме:

```
if <условие> then
begin
    команды 1;
end
else
begin
    команды 2;
end;
```

Ветвление в сокращенной форме

```
if <условие> then
begin
    команды;
end;
```

Цикл с предусловием

```
while <условие> do
begin
    тело цикла;
end;
```

Цикл с параметром

Параметр увеличивается:

```
for var i := N1 to N2 do
begin
    тело цикла;
end;
```

Параметр уменьшается:

```
for var i := N2 downto N1 do
begin
    тело цикла;
end;
```

Пример 6.3. Общий вид процедуры:
procedure <имя>(<список параметров>:тип);

```
var <описание переменных>
begin
    <команды>
end;
```

Общий вид функции:

```
function <имя>(<список параметров>:тип): тип результата;
var <описание переменных>
begin
    <команды>
    <имя> := <значение>;
end;
```

Функция должна содержать команду вида <имя> := <значение>;. Эта команда определяет, что функция должна вернуть в качестве результата.

Пример 6.4. Вызов процедуры и функции.

Вызов процедуры рисования круга:
`Circle(250, 125, 30);`
 вызов функции для вычисления
 квадратного корня и синуса:
`d := sqrt(2) * sin(x);`

Пример 6.5. Целочисленные типы данных в PascalABC.

Тип	Диапазон значений	Размер памяти, байт
shortint	−128..127	1
smallint	−32768..32767	2
integer, longint	−2147483648..2147483647	4
byte	0..255	1
word	0..65535	2
longword, cardinal	0..4294967295	4

Дополнительно в PascalABC определен тип `BigInteger`, который не ограничен диапазоном значений и размером памяти.

Пример 6.6. Вещественные типы данных в PascalABC.

Тип	Диапазон значений	Размер памяти, байт
real (double)	$-1.8 \cdot 10^{308} \dots 1.8 \cdot 10^{308}$	8
single	$-3.4 \cdot 10^{38} \dots 3.4 \cdot 10^{38}$	4
decimal	$-(2^{96} - 1) \dots 2^{96} - 1$	16

Количество значащих цифр в типе `real` составляет 15—16, в типе `single` — 7—8, в типе `decimal` — 28—29.

Тип `real` имеет другое название — `double`. Самое маленькое положительное число типа `real` приблизительно равно $5.0 \cdot 10^{-324}$, для типа `single` оно составляет приблизительно $1.4 \cdot 10^{-45}$.

внутри данной процедуры или данной функции.

Функции, в отличие от процедур, в результате своего выполнения возвращают значение, которое может быть использовано в выражении. Вызов процедуры является отдельной командой (пример 6.4).

6.3. Типы данных

В языке Pascal используются разные типы данных. Они нужны для выполнения различных операций — с каждым типом данных связан свой набор операций.

Для хранения различных типов данных в памяти компьютера отводится разное количество памяти. Вы уже использовали типы данных, которые называют простыми. В таблице примера 6.5 приведены целочисленные типы данных, используемые в PascalABC.

Все целочисленные типы данных, представленные в таблице, можно разделить на две группы:

- **знаковые** (диапазон значений которых содержит как положительные, так и отрицательные числа);
- **беззнаковые** (диапазон значений содержит только неотрицательные числа).

Для каждого знакового типа есть беззнаковый, занимающий столько же памяти.

Вещественные типы данных позволяют хранить число, представленное в стандартном виде: $a \cdot 10^n$, где $1 \leq a < 10$

и n (целое) есть порядок числа, записанного в стандартном виде (пример 6.6).

Значения типа `boolean`, который называют логическим, занимают 1 байт и принимают одно из двух значений, задаваемых константами `true` (истина) и `false` (ложь).

Данные в программу пользователь может вводить с помощью команды `read` (`readln`). Для вывода данных используется команда `write` (`writeln`).

Оператор присваивания используется для того, чтобы задавать значения переменным и вычислять значение выражения.

При использовании в одном операторе присваивания данных разных типов нужно помнить об их совместимости:

- переменной целочисленного типа нельзя присвоить вещественное значение;
- для данных вещественных типов определены операции «+», «-», «*», «/»;
- для данных целочисленных типов определены операции «+», «-», «*», «div», «mod».

Целочисленные типы могут быть преобразованы к вещественным, но не наоборот (пример 6.7).

Все простые типы, кроме вещественного, называются **порядковыми**. Значения только этих типов могут быть параметрами цикла `for`. Для порядковых типов используют функции `ord`, `pred` и `succ`, а также процедуры `inc` и `dec` (пример 6.8).

Пример 6.7. Приведение типов.

```
var a, b, c: real;
    x, y, z: integer;
begin
  a := 1; b := 3; x := 6; y := 4;
  //вычисления с вещественным
  типом
  c := a / b; writeln(c);
  //преобразование к веществен-
  ному
  c := x / y; writeln(c);
  //вычисления с целым типом
  z := x div y; writeln(z);
  //преобразование к веществен-
  ному
  c := x div y; writeln(c);
end.
```

Результат:

Окно вывода

```
0.333333333333333
1.5
1
1
```

Пример 6.8. Процедуры и функции для работы с порядковыми типами.

Функция	Описание
<code>Ord(a)</code>	Порядковый номер значения <code>a</code>
<code>Pred(x)</code>	Значение, предшествующее <code>x</code>
<code>Succ(x)</code>	Значение, следующее за <code>x</code>

Процедура	Описание
<code>Inc(i);</code>	Увеличивает значение переменной <code>i</code> на 1
<code>Inc(i, n)</code>	Увеличивает значение переменной <code>i</code> на <code>n</code>
<code>Dec(i);</code>	Уменьшает значение переменной <code>i</code> на 1
<code>Dec(i, n)</code>	Уменьшает значение переменной <code>i</code> на <code>n</code>

Пример 6.9. Фрагмент программы для работы с порядковыми типами.

```
var a, c: integer;
    b: boolean;
begin
  a := 10; c := 3; b := true;
  writeln(ord(a));
  writeln(pred(b));
  writeln(succ(c));
  inc(b); writeln(b); dec(c, a);
  writeln(c);
end.
```

Окно вывода

```
10
False
4
False
-7
```

Пример 6.10.

V. Программа:

```
uses GraphABC;
const slovo = 'Pascal';
begin
```

```
  SetFontColor(clRed);
  writeln(slovo);
```

end.

VI. Тестирование.

Запустить программу. Результат:



Пример 6.11.

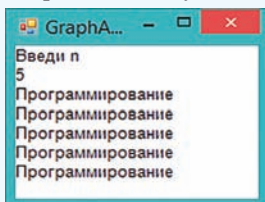
V. Программа:

```
uses GraphABC;
const sl = 'Программирование';
var n: integer;
begin
```

```
  writeln ('Введи n ');
  read(n); writeln (n);
  for var i := 1 to n do
    writeln(sl);
```

end.

VI. Тестирование. Результат при $n = 5$:



и пример 6.9). Все переменные, которые используются в программе, должны быть описаны в разделе **var**. Если данные не изменяются в процессе работы программы, то они могут быть описаны как константы в разделе **const**. Например:

```
const slovo = 'Привет';
      Pi = 3.1416;
```

Для работы с графическими данными используются команды библиотеки GraphABC (см. Приложение 2, с. 159—160).

6.4. Примеры задач

Пример 6.10. Описать слово «Pascal» как константу. Вывести слово на экран красным цветом.

Этапы выполнения задания

I—II. Результат работы не зависит от исходных данных.

III. Алгоритм решения задачи.

1. Установить красный цвет (команда находится в библиотеке GraphABC).

2. Вывести константу.

IV. В программе нет переменных.

Пример 6.11. Написать программу, которая выведет заданное слово на экран n раз. Значение n вводится.

Этапы выполнения задания

I. Исходные данные: переменная n .

II. Результат: n слов.

III. Алгоритм решения задачи.

1. Описываем слово как константу.

2. Вводим значения n .

3. Воспользуемся циклом **for** для вывода слова n раз.

4. Выводим слова в цикле.

IV. Описание переменных: n — integer.

Пример 6.12. Написать программу, которая выведет на экран n раз одно из двух слов. Выбор слова осуществляется случайным образом. Значение n вводится. Посчитайте, сколько раз было выведено каждое слово.

Этапы выполнения задания

I. Исходные данные: переменная n .

II. Результат: n раз выведено одно из двух слов и сообщение о том, сколько раз выведено каждое из слов.

III. Алгоритм решения задачи.

1. Слова описываем как константы.

2. Вводим значения n .

3. Инициализируем нулем переменные $k1$ и $k2$, которые будут подсчитывать, сколько раз выведено каждое слово.

4. Для вывода слов используем цикл **for**.

4.1. Сгенерируем случайное число x на промежутке $[0; 2)$.

4.2. Если $x = 0$, то выведем первое слово и увеличим значение переменной $k1$ на 1.

4.3. Иначе выведем второе слово и увеличим значение переменной $k2$ на 1.

5. Выводим сообщения о количестве слов.

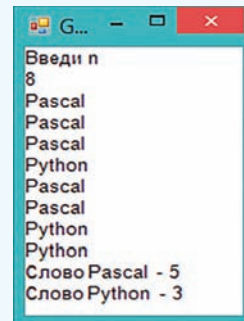
IV. Описание переменных: n , $k1$, $k2$, x — integer.

Пример 6.12.

V. Программа:

```
uses GraphABC;
const sl1 = 'Pascal';
      sl2 = 'Python';
var n, k1, k2, x: integer;
begin
  writeln ('Введи n ');
  read(n);
  writeln (n);
  k1 := 0; k2 := 0;
  for var i := 1 to n do
  begin
    x := random(2);
    if x = 0 then
    begin
      writeln(sl1);
      k1 := k1 + 1;
    end
    else
    begin
      writeln(sl2);
      k2 := k2 + 1;
    end
  end;
  writeln('Слово ',sl1,' - ',k1);
  writeln('Слово ',sl2,' - ',k2);
end.
```

VI. Тестирование. Запустить программу. Результат при $n = 8$:



1. Что такое алгоритм?

2. Назовите основные алгоритмические конструкции.

3. Что понимают под вспомогательным алгоритмом?

4. Чем отличаются различные целочисленные типы данных друг от друга?

5. Что нужно помнить о совместимости типов данных?

6. Какие арифметические операции определены для целочисленных типов данных? Для вещественных типов данных?



Упражнения

- 1 Измените константу в программе примера 6.10 на свое имя. Используя команды графической библиотеки для работы с текстом, измените шрифт, размер символов, фон, начертание букв.
- 2 Измените программу примера 6.11 так, чтобы выполнялись указанные условия.
 1. Каждое слово должно выводиться случайным цветом.
 - 2*. Расстояние между словами должно быть 50 пикселей.
 - 3*. Каждое новое слово должно выводиться шрифтом на 5 пунктов больше, чем предыдущее. Отрегулируйте расстояние между словами так, чтобы слова при выводе не перекрывали друг друга.
- 3 Запустите программу из примера 6.12 несколько раз. Какие результаты получили?
- 4 Измените программу из примера 6.12 так, чтобы случайным образом выбиралось одно из трех слов. Выводите каждое слово своим цветом (например, первое — красным, второе — синим, третье — зеленым).

§ 7. Строковые величины

В первых языках программирования строкового типа данных не было; программист должен был сам строить функции для работы со строками.

В 1962 г. был разработан язык SNOBOL (StriNg Oriented symBolic Language), ориентированный на работу со строками. В конце 60-х гг. XX в. строковые типы данных появились в языках Algol и Fortran.

Две строки, в отличие от двух чисел, нельзя прочитать с помощью одной команды `read`, поскольку пробел для строк не разделитель, а такой же символ, как и все остальные. Необходимо использовать две команды `readln`.

Если использовать две команды `read`, то первая строка будет считана так, как нужно, а вторая строка будет пустой (она не будет вводиться). Это происходит потому, что первая команда `read` считывает данные до нажатия клавиши Enter. Вторая команда `read` прочитает один символ — символ нажатия клавиши Enter.

7.1. Ввод, вывод, присваивание строковых величин

Современные компьютеры способны обрабатывать данные, представленные различными способами: числа, тексты, графику, звуки. Вы уже знаете, как на языке программирования Pascal можно работать с целыми и вещественными числами, выполнять простейшие графические построения. Обработка текстовых данных является сегодня наиболее актуальной — это обработка различных поисковых запросов в Интернете, перевод текстов с одного языка на другой, озвучивание компьютером печатного текста и др.

В языке Pascal для работы с текстовыми данными используется тип **string (строка)**. Строки состоят из набора последовательно расположенных символов и используются для хранения текста. Они могут иметь произ-

вольную длину. Строка, в которой нет ни одного символа, называется **пустой**.

Строка описывается следующим образом:

```
var <имя строки>: string;
```

Для ввода и вывода строки используются те же команды, что и для ввода и вывода чисел: `read` (`readln`) и `write` (`writeln`) (пример 7.1). Ввод данных всегда заканчивается нажатием клавиши `Enter`, которой соответствует специальный символ: `¶`. Команда `read` считывает символы в строку до тех пор, пока не встретится этот специальный символ. Команда `readln` отличается тем, что считывает не только данные в строку, но и символ нажатия `Enter`. Сам символ `¶` к строке не приписывается.

Переменной строкового типа можно присвоить значение с помощью команды присваивания. Значение строковой величины записывается в апострофах. Пустая строка задается следующим образом: `S := ''`;

Запись поясняющего текста при выводе в команде `write` является строковой константой. К символам в строке можно обращаться, используя индекс. Нумерация символов начинается с единицы, `s[i]` соответствует `i`-му символу в строке `s` (пример 7.2).

Пример 7.3. Написать программу, которая спросит имя пользователя и выведет приветствие, обращаясь по имени.

Этапы выполнения задания

I. Исходные данные: `imja` — строка, в которой будет храниться введенное имя.

Пример 7.1. Введем строку `s` и выведем ее значение.

```
var s: string;
begin
  writeln('Введи строку');
  readln(s);
  writeln('Ввели строку: ', s);
end.
```

Результат работы:

```
Окно вывода
Введи строку
Pascal
Ввели строку: Pascal
```

Пример 7.2. Вывести третий символ строки «Информатика».

```
var s: string;
begin
  s := 'Информатика';
  writeln('3-й символ - ', s[3]);
end.
```

Результат работы:

```
Окно вывода
3-й символ - ф
```

Пример 7.3.

V. Программа:

```
var imja: string;
begin
  writeln('Как тебя зовут?');
  readln(imja);
  writeln('Привет, ', imja);
end.
```

VI. Тестирование.

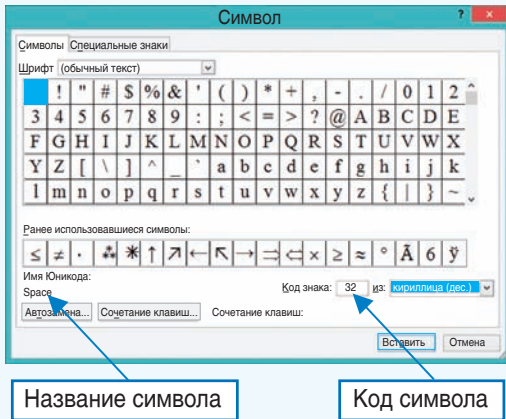
Запустить программу и ввести значение «Катя». Результат:

```
Окно вывода
Как тебя зовут?
Катя
Привет, Катя
```


Пример 7.4. Порядок расположения групп символов в таблице символов.

1. Пробел.
2. Символы цифр.
3. Заглавные латинские буквы.
4. Строчные латинские буквы.
5. Заглавные русские буквы.
6. Строчные русские буквы.

В нижеприведенной таблице символов выделен пробел, имеющий код 32. Далее видно расположение цифр и латинских букв.



Пример 7.5. Примеры сравнения строк.

S1	S2	Результат
'string'	'char'	S1 > S2
'строка'	'Строка'	S1 > S2
'книга'	'Журнал'	S1 > S2
'мебель'	'стол'	S1 < S2
'липа'	'лист'	S1 < S2
'леска'	'лес'	S1 > S2
'112'	'7'	S1 < S2
'123'	'11111111'	S1 > S2
'123'	'123'	S1 = S2

- II. Результат: строка с приветствием.
- III. Алгоритм решения задачи.

1. Ввод имени.

2. Вывод строки с приветствием.

IV. Описание переменных: `imja` — `string`.

7.2. Сравнение и сложение строковых величин

Так же как и для других типов данных, для строк определены свои операции — действия, которые можно выполнять с данными типа `string`. Для строкового типа такими операциями являются сравнение и сложение.

Для сравнения строк нужно уметь сравнивать символы. Все символы записаны в таблице символов (в документе Word эту таблицу вы использовали для вставки символа, отсутствующего на клавиатуре). Каждый символ в таблице имеет свой код (номер), и при сравнении символов сравниваются их коды — номера в таблице символов.

В таблице символов, которую использует PascalABC, 65536 символов (стандарт Юникод — англ. *Unicode*¹). Запомнить, в каком порядке записаны символы, невозможно. Достаточно знать, как расположены группы наиболее часто употребляемых символов. В примере 7.4 приведены некоторые группы символов в порядке их расположения в таблице.

Строки сравниваются посимвольно. Сначала сравниваются первые символы двух строк. Если символы различны, то больше та строка, символ которой имеет больший номер. Если

¹ <https://ru.wikipedia.org/wiki/Юникод> (дата доступа: 05.01.2019).

символы одинаковые, то переходят к сравнению следующих символов. Сравнение заканчивается, когда найдены различные символы или в одной из строк закончились символы — в этом случае больше та строка, в которой символы остались. Если при сравнении символов различия не найдены и строки закончились одновременно, то они равны (пример 7.5). Порядок, в котором меньшая строка предшествует большей, называют **лексикографическим**. Это название он получил по аналогии с размещением по алфавиту в словаре.

При сравнении строк следует помнить, что заглавные и строчные буквы — это разные буквы, поскольку имеют различные номера в таблице символов. Поэтому строки 'мама' и 'Мама' будут различными, а неравенство 'мама' > 'Мама' будет верным (у буквы 'м' код 236, а у 'М' — 204). Для проверки правильности сравнения строк можно воспользоваться программой из примера 7.6.

Пример 7.7. Написать программу, которая спросит имя пользователя, его пол, а затем поздоровается с ним, выводя красным цветом женские имена, а синим — мужские.

Этапы выполнения задания

I. Исходные данные: переменные `imja` (имя пользователя) и `pol` (пол пользователя).

II. Результат: строка с приветствием.

III. Алгоритм решения задачи.

Частично задача совпадает с примером 7.3, поэтому возьмем решение

Пример 7.6. Сравнение строк.

```
var S1,S2: string;
begin
  writeln('Введите первую строку');
  readln(S1);
  writeln('Введите вторую строку');
  readln(S2);
  if S1>S2 then
    writeln('S1 > S2')
  else
    if S1<S2 then
      writeln('S1 < S2')
    else
      writeln('S1 = S2')
end.
```

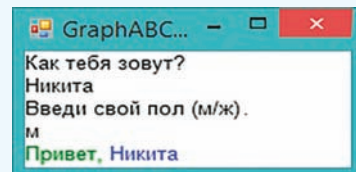
В PascalABC есть тип `char`, который позволяет хранить один символ. Над символами определены операции сравнения «<», «>», «<=», «>=», «=», «<>», которые сравнивают коды символов. Функция `ChrUnicode(n)` возвращает символ с кодом `n` (тип `char`), а функция `OrdUnicode(c)` — код символа (тип `byte`).

Пример 7.7.

```
V. Программа:
uses GraphABC;
var imja, pol: string;
begin
  writeln('Как тебя зовут?');
  readln(imja); writeln(imja);
  writeln('Введи свой пол (м/ж).');
  readln(pol); writeln(pol);
  SetFontColor(clgreen);
  write('Привет, ');
  if pol = 'ж' then
    SetFontColor(clred)
  else
    SetFontColor(clblue);
  writeln(imja);
end.
```

VI. Тестирование.

Результат для имени Никита:



Пример 7.8. Примеры сложения строк.

```
var s1, s2, s3, s4: string;
begin
  s1 := 'Петя';
  s2 := 'Иванов';
  s3 := s1 + ' ' + s2;
  s4 := s2 + ' ' + s1;
  writeln(s3);
  writeln(s4);
end.
```

Результат:

Окно вывода
Петя Иванов
Иванов Петя

Пример 7.9.

V. Программа:

```
var str, gor, s: string;
begin
  writeln('Введите название госу-
дарства');
  readln(str);
  writeln('Введите название
столицы');
  readln(gor);
  s := 'Столица государства '
+ str + ' — город ' + gor;
  writeln(s);
end.
```

VI. Тестирование.

Запустить программу и ввести значения Республика Беларусь и Минск. Результат:

Окно вывода
Введите название государства
Республика Беларусь
Введите название столицы
Минск
Столица государства Республика Беларусь —
город Минск

из этого примера и изменим его. Цвет для текста можно задавать только в графическом окне, поэтому подключим графический режим.

1. Ввод исходных данных.

2. Сравнение переменной `pol` со значением 'ж'. Если результат сравнения — истина, то устанавливаем красный цвет, иначе — синий.

3. Вывод строки.

IV. Описание переменных: `imja`, `pol` — string.

Для строк определена операция *сложение* (конкатенация). Обозначается операция знаком «+». В результате сложения двух строк получается новая строка, в которой после символов первой строки будут записаны символы второй строки. Результат данной операции зависит от порядка слагаемых (пример 7.8).

Пример 7.9. Написать программу, которая просит ввести название государства и его столицу, затем выводит сообщение:

Столица государства ... — город ...

Вместо многоточия должны быть выведены соответствующие значения.

Этапы выполнения задания

I. Исходные данные: переменные `str` (название государства) и `gor` (название города).

II. Результат: переменная `s` (итоговая строка).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Создание итоговой строки.

3. Вывод строки.

IV. Описание переменных: `str`, `gor`, `s` — string.

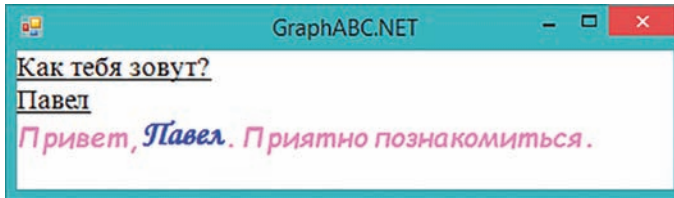


1. Как описываются строковые переменные?
2. Из чего состоят строки?
3. Какие операции возможны над строками?
4. Как упорядочены символы в таблице символов?
5. Как сравниваются строки?
6. Что является результатом сложения двух строк?

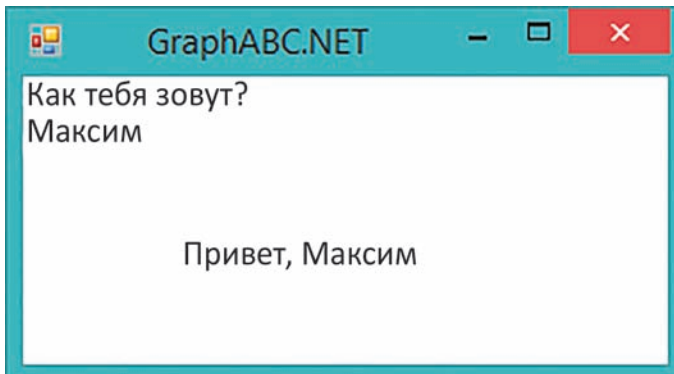


Упражнения

- 1 Внесите указанные изменения в программу из примера 7.3.
 1. После имени пользователя выводить восклицательный знак.
 2. Измените программу так, чтобы выводился текст:
Привет, Ольга. Приятно познакомиться.
 3. Подключите графический режим. Задайте параметры шрифта для текста «Как тебя зовут?» и выводимого текста. Например, так:



- 2 Выполните следующие задания, изменив программу примера 7.3.
 1. Выведите приветствие в графическом окне в точке с координатами (70; 70).



2. Задайте координаты вывода случайным образом.
3. Измените программу так, чтобы координаты месторасположения текста можно было вводить.
4. Задайте параметры шрифта для выводимых сообщений.

3 Используя программу из примера 7.6, проведите исследование по сравнению символов.

1. Совместно заполните таблицы.
 2. Добавьте в каждую таблицу 2—3 строки с другими символами в соответствии с правилом сравнения.
 3. Проверьте некоторые из результатов, используя таблицу символов (можно открыть в Word).
- Сделайте выводы.

Сравнение символов русского алфавита, набранных в одном регистре		
S1	S2	Результат
ф	а	
С	Я	
я	у	

Сравнение символов латинского алфавита, набранных в одном регистре		
S1	S2	Результат
r	z	
W	J	
q	d	

Сравнение символов русского алфавита, набранных в разных регистрах		
S1	S2	Результат
ы	Ы	
а	Я	
я	А	

Сравнение символов латинского алфавита, набранных в разных регистрах		
S1	S2	Результат
Q	q	
Z	a	
A	z	

Сравнение символов цифр		
S1	S2	Результат
1	3	
7	9	
4	4	

Сравнение цифр и букв		
S1	S2	Результат
Q	1	
2	б	
9	ю	

Сравнение русских и латинских букв		
S1	S2	Результат
Q	л	
w	б	
Ч	k	

Сравнение знаков и цифр		
S1	S2	Результат
,	5	
7	!	
9	>	

Сравнение знаков и букв		
S1	S2	Результат
пробел	f	
W	{	
№	Я	

Сравнение знаков		
S1	S2	Результат
*	+	
.	,	
(	)	

4 Используя программу из примера 7.6, сравните строки. Откройте файл с таблицей и запишите результаты.

S1	S2	Результат сравнения
'string'	'String'	
'строка'	'символ'	
'alinéa'	'caractère'	
'Zeile'	'Zeichen'	
'cuerda'	'simbolo'	
'778'	'8'	
'876'	'55555555'	
'2 + 2'	'4'	

5 Какое приветствие выведет программа из примера 7.7, если, указывая пол, пользователь введет символ, отличный от «м» и «ж»? Внесите в программу изменения так, чтобы в этом случае вместо приветствия выводилось сообщение «Ошибка ввода».

6 Напишите программу, которая организует диалог с пользователем по следующему шаблону:

Как тебя зовут?
Ваня
Привет, Ваня. А ты любишь читать? (д/н)
д
Хорошо! Книга – источник знаний.

Как тебя зовут?
Петя
Привет, Петя. А ты любишь читать? (д/н)
н
Плохо, из книг можно узнать много интересного.

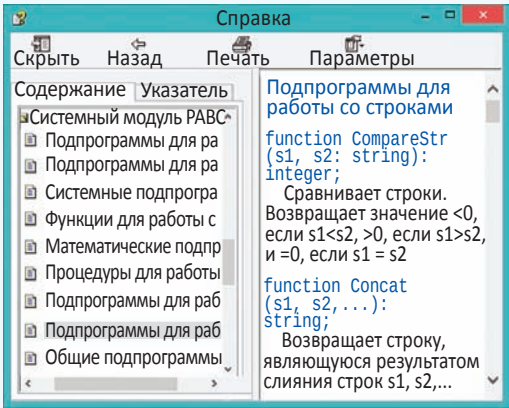
Как тебя зовут?
Леша
Привет, Леша. А ты любишь читать? (д/н)
v
Леша, ты ответил на вопрос некорректно.

Добавьте в программу свои вопросы и сообщения.

7 Напишите программу для решения следующей задачи: пользователь вводит свою фамилию, класс и учебное заведение, программа должна вывести сообщение «Учащийся ... учится в ... классе ГУО “...”». Вместо многоточия должны быть выведены соответствующие значения.

§ 8. Стандартные процедуры и функции для работы со строковыми величинами

Пример 8.1. В справке среды программирования PascalABC.NET в разделе **Системный модуль PascalABC-System → Подпрограммы для работы со строками** можно найти описание функций и процедур:



Пример 8.2. Примеры использования функций.

d := Length ('Компьютер');	d = 9
s := 'Строка'; d := Length(s);	d = 6
s := 'Не слово хозяин хозяину, а хозяин слову хозяин'; s1 := 'хозяин'; d := Length(s); N1 := Pos(s1,s); N2 := LastPos(s1,s); N3 := PosEx(s1,s,15);	d = 46 N1 = 10 N2 = 41 N3 = 17

PascalABC позволяет обращаться к функциям обработки строк и по-другому: если после имени строковой переменной поставить точку, то появится список функций.



8.1. Поиск в строке

Современные компьютерные устройства позволяют достаточно быстро осуществлять поиск в тексте, используя для этого различные алгоритмы. Языки программирования предоставляют широкий набор функций для работы с текстом. Некоторые функции языка программирования Pascal для поиска подстроки (части строки) в другой строке представлены в таблице.

Функция	Описание
Length(s)	Определяет длину строки S (количество символов в строке)
Pos(s1, s)	Определяет позицию подстроки s1 в строке s. Если не найдена — возвращает 0
LastPos(s1, s)	Определяет позицию последнего вхождения подстроки s1 в строке s. Если не найдена — возвращает 0
PosEx(s1,s,from)	Определяет позицию подстроки s1 в строке s, начиная с позиции from. Если не найдена — возвращает 0

Подробное описание функций и процедур для работы со строками можно найти в справочной системе PascalABC.NET (пример 8.1), а также в *Приложении 2* (см. с. 161—162).

В примере 8.2. показано, как применять указанные функции.

Пример 8.3. Написать программу, которая вводит слово, а затем выводит его по одному символу в строке.

Этапы выполнения задания

I. Исходные данные: переменная *s* — исходное слово.

II. Результат: слово на экране, каждый символ в отдельной строке.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Определяем длину слова. Переменной *n* присваиваем значение функции `length(s)`.

3. В цикле **for** выводим по одному символу введенного слова.

IV. Описание переменных: *s* — string, *n* — integer.

Пример 8.4. Написать программу, которая выводит на экран последний символ введенного слова и определяет, встречается ли этот символ в слове еще раз. Если встречается, то программа выводит индекс символа.

Этапы выполнения задания

I. Исходные данные: переменная *s* — введенное слово.

II. Результат: последний символ в слове и соответствующее сообщение — встречается или не встречается.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Определяем длину слова. Переменной *n* присваиваем значение функции `length(s)`.

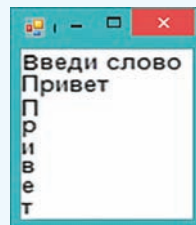
3. Определяем последний символ. Поскольку символы в строке нумеруются с 1, то номер последнего символа совпадает с длиной строки.

Пример 8.3.

V. Программа:

```
uses GraphABC;
var
  s: string; n: integer;
begin
  writeln('Введи слово');
  readln(s); writeln(s);
  n := length(s);
  for var i := 1 to n do
    writeln(s[i]);
  end.
```

VI. Тестирование.



Пример 8.4.

V. Программа:

```
var s: string; n, k: integer;
begin
  writeln('Введи слово');
  readln(s); n := length(s);
  writeln('Последний символ - ',
s[n]);
  k := pos(s[n], s);
  if k = n then
    writeln('Символ один')
  else
    writeln('Символ с индексом ', k)
end.
```

VI. Тестирование. Результаты:

Окно вывода

```
Введи слово
Строка
Последний символ - а
Символ один
```

Окно вывода

```
Введи слово
Информатика
Последний символ - а
Символ на месте 7
```

Один из самых известных и эффективных алгоритмов поиска подстроки в строке — алгоритм Кнута — Морриса — Пратта (КМП-алгоритм). Алгоритм был разработан Д. Кнутом и В. Праттом и (независимо от них) Д. Моррисом. Результаты своей работы они опубликовали совместно в 1977 г. Время работы алгоритма линейно зависит от объема входных данных.



Дональд Кнут
(род. в 1938)



Вон Пратт
(род. в 1944)



Джеймс Моррис
(род. в 1941)

Дональд Эрвин Кнут — американский ученый, автор известной серии книг об основных алгоритмах и методах вычислительной математики.

Вон Рональд Пратт — почетный профессор в Стэнфордском университете, известен своим вкладом в развитие таких областей информатики, как алгоритмы поиска и сортировок, а также тестирование простоты чисел.

Джеймс Хирэм Моррис — американский профессор. Занимался разработкой в области языков программирования и технологическим дизайном программных продуктов.

Пример 8.5. Примеры использования функций.

```
S := 'Информатика';  
S := UpperCase(s);  
После преобразования в строке S  
будет записано ИНФОРМАТИКА
```

```
S := 'Информатика';  
S[1] := LowCase(s[1]);  
После преобразования в строке S  
будет записано информатика
```

4. Определяем позицию последнего символа в слове. Переменной *k* присваиваем значение функции *pos* для последнего символа. Если оно равно длине строки, то символ в слове единственный, иначе в слове есть другой такой же символ.

5. Выводим результат.
IV. Описание переменных: *s* — string, *n*, *k* — integer.

Поиск текстовой информации во всемирной паутине заключается в том, чтобы по запросу пользователя найти документы, содержащие указанные ключевые слова. Для этого различные поисковые системы используют разные алгоритмы. Запрос, который вводит пользователь, может содержать строчные и заглавные буквы. Для осуществления поиска буквы в слове обычно приводят к одному регистру: либо все строчные, либо все заглавные. В Pascal также есть функции преобразования.

Функция	Описание
LowCase(c)	Преобразует один символ (букву) в строчную букву
LowerCase(s)	Преобразует все символы (буквы) строки в строчные буквы
UpCase(c)	Преобразует один символ (букву) в заглавную букву
UpperCase(s)	Преобразует все символы (буквы) строки в заглавные буквы

В примере 8.5 показано применение этих функций.

8.2. Копирование, вставка и удаление символов

При работе с текстом в текстовом редакторе часто приходится пользоваться буфером обмена. Часть текста (подстрока) копируется (вырезается) в буфер обмена, а затем вставляется в другое место в тексте. В языке Pascal реализованы команды для работы с фрагментом текста, которые представлены в таблице.

Команда	Описание
Copy (s,index,count)	Функция копирует часть строки S в другую строку
Delete (s,index,count);	Процедура удаляет символы строки S
Insert (s1,s,index);	Процедура вставляет подстроку s1 в строку S

Во всех командах переменная S обозначает исходную строку, над которой производится операция. Переменная index обозначает позицию символа, начиная с которого выполняют операцию, а переменная count — количество символов. Разберем команды подробнее (пример 8.6).

Команда copy является функцией, и результат ее работы присваивается другой переменной. Строка S при этом не изменяется.

```
s1 := copy(s, index, count);
```

Команды delete и insert являются процедурами, они изменяют строку S. В примере 8.7 показано, как применяются указанные команды.

Пример 8.6. Команды для преобразования строк.

1. Запись `s1 := Copy(s, index, count);` означает, что в строке S выделяют count символов, выделение начинают с символа, индекс которого записан в переменной index. Эти символы копируются в переменную s1 (действие команды сравнимо с копированием фрагмента текста в буфер обмена).

2. Запись `Delete(s, index, count);` означает, что в строке S выделяют count символов, выделение начинают с символа, индекс которого записан в переменной index. Выделенные символы удаляются из строки S. Остальные символы строки сдвигаются влево (действие команды сравнимо с удалением фрагмента текста).

3. Запись `Insert(s1, S, index);` означает, что в строку S вставляют символы строки s1, вставка происходит в позиции index. Остальные символы строки сдвигаются вправо (действие команды сравнимо со вставкой фрагмента текста из буфера обмена).

Пример 8.7. Примеры использования команд.

S := 'Информатика';	
s1 := copy(s, 3, 5);	s1 = 'форма'
Delete(s, 8, 4);	S = 'Информа'
Insert ('ция', s, 8);	S = 'Информация'

Пример 8.8.

V. Программа:

```

var s, p, t: string;
    n1, n2, k: integer;
begin
  writeln('Строка s');
  readln(s);
  writeln('Подстрока p');
  readln(p);
  n1 := length(s);
  n2 := length(p);
  k := 0;
  for var i := 1 to n1 - n2 + 1 do
  begin
    t := copy(s, i, n2);
    if t = p then
      k := k + 1;
  end;
  writeln('Встречается ', k,
  ' раз(-a)');
end.

```

VI. Тестирование.

Запустить программу, ввести строку «Не слово хозяин хозяину, а хозяин слову хозяин слову хозяин» и подстроку «хозяин». Результат:

Окно вывода

```

Строка s
Не слово хозяин хозяину, а хозяин слову хозяин
Подстрока p
хозяин
Встречается 4 раз(-a)

```

Если для той же строки ввести подстроку «хозяйка», то результат будет таким:

Окно вывода

```

Строка s
Не слово хозяин хозяину, а хозяин слову хозяин
Подстрока p
хозяйка
Встречается 0 раз(-a)

```

Пример 8.8. Написать программу, которая определит, сколько раз заданная подстрока встречается в строке.

Этапы выполнения задания

I. Исходные данные: переменная *S* — исходная строка, *p* — исходная подстрока.

II. Результат: *k* — искомое количество.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Инициализируем значение счетчика *k* := 0;

3. Определяем длины *n1* и *n2* для строки *S* и подстроки *p*.

4. В цикле **for** от 1 до разницы в длинах строки *S* и подстроки *p*:

4.1. Выделяем из строки *S* подстроку *t* такой же длины, что и длина *p*, начиная с текущего символа.

4.2. Сравниваем подстроки. Если они равны, то увеличиваем значение счетчика на 1.

5. Выводим результат.

IV. Описание переменных: *S*, *p*, *t* — string, *n1*, *n2*, *k* — integer.

Пример 8.9. Написать программу, которая из слова ТЕСТИРОВАНИЕ получит слово РИСОВАНИЕ, используя процедуры и функции преобразования строк.

Этапы выполнения задания

I. Исходные данные: слово ТЕСТИРОВАНИЕ будем хранить как константу с именем *S*.

II. Результат: полученное слово.

III. Алгоритм решения задачи.

1. Результат не зависит от вводимых данных.

2. В строку s1 запишем шестой символ исходной строки.

3. Скопируем из строки s восемь символов, начиная с позиции 5. Добавим к строке s1. Получим 'РИРОВАНИЕ'.

4. В полученной строке удалим третий символ ('РИОВАНИЕ').

5. Вставим на третье место третий символ исходной строки ('РИСОВАНИЕ').

6. Выведем результат.

IV. Описание переменных: s1 – string.

Пример 8.9.

V. Программа:

```
const s = 'ТЕСТИРОВАНИЕ';
var s1: string;
begin
  s1 := s[6]; s1 := s1 + copy(s,5,8);
  //РИРОВАНИЕ
  delete(s1, 3, 1);
  //РИОВАНИЕ
  insert(s[3], s1, 3);
  //РИСОВАНИЕ
  writeln('Слово =', s1);
end.
```

VI. Тестирование.

Окно вывода

Слово = РИСОВАНИЕ

Для копирования последних 8 символов из строки s можно использовать функцию RightStr: s1 := s1 + RightStr(s,8).

Для копирования первых символов из строки можно использовать функцию LeftStr.

-  1. Что такое длина строки?
 2. С помощью какой функции можно найти длину строки?
 3. Какие функции используются для определения позиции подстроки в строке?
 4. Как скопировать символы из одной строки в другую?
 5. Какая процедура используется для удаления символов из строки?
 6. Какая процедура предназначена для вставки символов в строку?
 7*. Сопоставьте команды преобразования строк с операциями, выполняемыми с помощью буфера обмена.

 Упражнения

- 1 В программу из примера 8.3 внесли следующие изменения:

```
for var i := 1 to n do
begin
  write(s[i]);
  if i mod 2 = 0 then
    writeln;
end;
```

Как теперь выводится слово? Объясните почему.

- 2 Измените программу из примера 8.3 так, как указано ниже.

1. Каждая буква должна выводиться своим цветом (можно использовать случайное задание цветов).

2. Буквы, стоящие на четных местах, должны выводиться одним цветом, а на нечетных — другим.

3 Измените программу из примера 8.4 так, чтобы на экран выводился символ введенного слова, стоящий посередине (для слов с четным количеством букв — символ справа от середины).

1. Проверьте правильность работы своей программы на предложенных примерах. Откройте файл с таблицей и запишите результаты.

Слово	Результат
Школа	о
гимназия	а
форма	
Интернет	

2. Допишите в таблицу два своих примера.

3. Что будет выведено, если ничего не вводить, просто нажать Enter?

4. Проверьте, встречается ли выведенный символ в слове еще раз.

5. Выведите позиции всех символов, совпадающих с символом слова, находящимся посередине.

4 Измените программу из примера 8.4 так, чтобы строчные и заглавные буквы анализировались программой одинаково (например, для слова «Анна» ответ должен быть следующим: «Последний символ — а, символ встретился на месте 1»).

5 Даны два слова. Верно ли, что одно из слов начинается на ту же букву, на которую заканчивается другое? (Первая буква одного из слов может быть заглавной.) Если да, то вывести букву, иначе — соответствующее сообщение.

1. Проверьте правильность работы своей программы на предложенных примерах. Откройте файл с таблицей и запишите результаты.

Слово 1	Слово 2	Результат
array	yellow	у
apple	auto	неверно
Рыба	Арбуз	

2. Допишите в таблицу два своих значения.

3*. Если ответ «верно», указать, принадлежат ли буквы одному регистру.

6 Измените программу из примера 8.8 так, чтобы при $k = 0$ выводилось сообщение 'Подстрока в строке не встречается'.

- 7 Программу из примера 8.8 записали следующим образом:

```
var s, p, t: string;
    k, i: integer;
begin
  writeln('Строка s');
  readln(s);
  writeln('Подстрока p');
  readln(p);
  k := 0; i := 1;
  while PosEx(p, s, i) <> 0 do
  begin
    i := PosEx(p, s, i) + 1; k := k + 1;
  end;
  writeln('Встречается', k, 'раз(-а)');
end.
```

Сравните эту программу и программу из примера 8.8, определив, сколько раз выполнится команда цикла в каждой из программ для перечисленных случаев.

1. Строка $s :=$ «Не слово хозяин хозяину, а хозяин слову хозяин», подстрока $p :=$ «хозяин».
2. Строка $s :=$ «Не слово хозяин хозяину, а хозяин слову хозяин», подстрока $p :=$ «хозяйка».

*Можно ли подобрать тест, для которого количество выполнений цикла будет одинаковым для обеих программ? Если да, то какой?

- 8 Получите из слова «ТЕСТИРОВАНИЕ» указанные слова. Для этого используйте команды: `copy`, `delete`, `insert` и операцию сложения строк.

1. РОСТ.
2. НИВА.
3. ТОВАР.
4. ТОСТЕР.
5. ОТВЕРСТИЕ.
6. Придумайте свое слово.

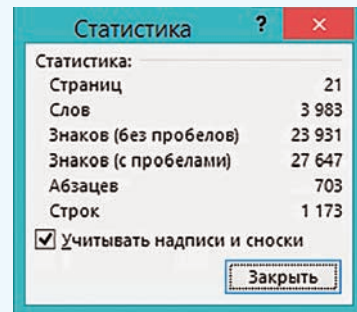
§ 9. Составление алгоритмов обработки строковых величин

9.1. Анализ текста на наличие различных символов

Современные текстовые редакторы позволяют получить статистику по символам и словам в документе (пример 9.1).

Грамотный набор текста предполагает наличие только одного пробела между словами. В правильно набранном тексте количество слов будет на единицу больше, чем количество пробелов.

Пример 9.1. Статистика в документе Word (соответствующая команда на вкладке **Рецензирование**).



Пример 9.2.

V. Программа:

```

var st: string;
    n, k: integer;
begin
  writeln('Введите текст');
  readln(st);
  n := length(st);
  k := 1;
  for var i := 1 to n do
  begin
    if st[i] = ' ' then
      k := k + 1;
  end;
  writeln('В тексте ', k, '
  слов(-o/-a)');
end.

```

VI. Тестирование (для проверки можно скопировать текст абзаца из документа Word).

Окно вывода

Введите текст - st

Современные текстовые редакторы позволяют получить статистику по символам и словам в документе (пример 9.1). Грамотный набор текста предполагает наличие одного пробела между словами. В таком тексте количество слов будет на 1 больше количества пробелов.

В тексте 34 слов (-o/-a)

VII. Анализ результата. Если посмотреть статистику Word для этого абзаца, получим:

Статистика	
Статистика:	
Страниц	1
Слов	34
Знаков (без пробелов)	219
Знаков (с пробелами)	252
Абзацев	1
Строк	7
☒ Учитывать надписи и сноски	
Заккрыть	

Пример 9.2. Написать программу, которая определит количество слов в тексте, если между любыми двумя словами ровно один пробел. Предполагается, что в тексте есть хотя бы одно слово.

Этапы выполнения задания

I. Исходные данные: строка текста st.

II. Результат: количество слов k.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Определяем длину строки.

3. Задаем начальное значение счетчика k = 1 (в тексте есть хотя бы одно слово).

4. С помощью цикла **for** просматриваем каждый символ в строке. Если текущий символ — пробел, то увеличиваем значение счетчика количества слов.

5. Выводим результат.

IV. Описание переменных: st — string, n, k — integer.

Пример 9.3. Написать программу, которая определит, каких знаков препинания в тексте больше — точек или запятых.

Этапы выполнения задания

I. Исходные данные: переменная st (текст).

II. Результат: сообщение о том, каких знаков больше.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Определяем длину строки.

3. Инициализируем два счетчика нулями (для точек и запятых).

4. В цикле **for** проверяем каждый символ строки st.

4.1. Если встретилась точка, то увеличиваем значение счетчика k1 на 1.

4.2. Если встретилась запятая, то увеличиваем значение счетчика k2 на 1.

5. Сравниваем полученные значения счетчиков и выводим результат.

IV. Описание переменных: st – string, n, k1, k2 – integer.

Для лучшего запоминания маленькими детьми гласных и согласных букв их часто окрашивают в разные цвета: гласные — красным, а согласные — синим (пример 9.4).

Пример 9.5. Написать программу, которая выведет в заданном слове согласные буквы синим цветом, а гласные — красным (в слове не встречаются «ь» и «ъ»). Посчитать количество гласных букв во введенном слове.

Этапы выполнения задания

I. Исходные данные: переменная s (слово).

II. Результат: слово, в котором буквы выводятся разными цветами, и сообщение о количестве гласных букв.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Определяем длину строки.

3. Создаем дополнительную строку, в которой хранятся все гласные буквы: g := 'ЕУЕЗОАЫЯЮёуеэоаыяю'.

4. Инициализируем счетчик гласных букв k := 0.

5. В цикле **for** проверяем каждый символ слова s.

5.1. Находим позицию текущего символа из строки s

Пример 9.3.

V. Программа:

```
var St: string;
    n, k1, k2: integer;
begin
    writeln('Введите текст');
    readln(St);
    n := length(St);
    k1 := 0; k2 := 0;
    for var i := 1 to n do
    begin
        if St[i] = '.' then
            k1 := k1 + 1;
        if St[i] = ',' then
            k2 := k2 + 1;
    end;
    if k1 > k2 then
        writeln('Точек больше')
    else
        if k2 > k1 then
            writeln('Запятых больше')
        else
            writeln('Количество запятых
                равно количеству точек');
    end.
```

VI. Тестирование (для проверки можно скопировать текст абзаца из документа Word).

Окно вывода

Введите текст

Для книжных стилей и письменной речи характерны сложные предложения, которые позволяют сделать речь более информативной и выразительной. Если текст состоит в основном из простых предложений, то в нем будут преобладать точки. В сложных предложениях, которые содержат несколько грамматических основ, встречаются запятые. Запятых больше

Пример 9.4. Русский алфавит.

А Б В Г Д Е
Ё Ж З И Й К
Л М Н О П Р
С Т У Ф Х Ц
Ч Ш Щ Ъ Ы Ь
Э Ю Я

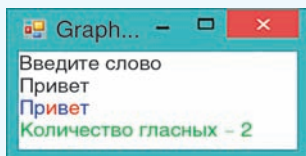
Пример 9.5.

```

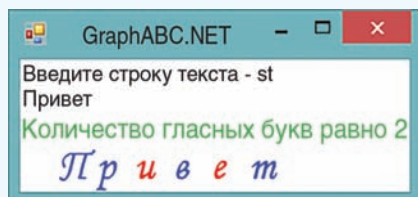
V. Программа:
uses GraphABC;
var s, g: string;
    n, k, p: integer;
begin
  writeln('Введите слово');
  readln(s); writeln(s);
  n := length(s);
  g := 'ЁУЕЗОАЫЯЮёуеэоаыяю';
  for var i := 1 to n do
  begin
    p := pos(s[i], g);
    if p <> 0 then
    begin
      SetFontColor(clRed);
      k := k + 1;
    end
    else
      SetFontColor(clBlue);
      write(s[i]);
    end;
    writeln;
    SetFontColor(clGreen);
    writeln('Количество
гласных - ', k);
  end.

```

VI. Тестирование. Ввести слово «Привет». Результат:



Поскольку текст выводится в графическом окне, то можно задавать не только цвет символов, но также шрифт и размер символов. Использование команды TextOut позволит выводить символы текста в произвольном месте графического окна. Например, так:



в строке g (переменная p). Если значение $p \neq 0$, то текущий символ встретился в строке с гласными буквами и, следовательно, сам является гласной. Если $p = 0$, то символ — согласная.

5.2. Если символ является гласной буквой, то выводим его на экран красным цветом и увеличиваем значение счетчика гласных букв на 1, если символ является согласной буквой, то выводим его синим цветом.

6. Выводим результат.

IV. Описание переменных: s, g — string, n, k, p — integer.

9.2. Преобразование строк

При записи вещественных чисел на уроках математики в качестве разделителя целой и дробной части используется символ «запятая». В языке программирования Pascal разделителем является точка.

Пример 9.6. Написать программу, которая заменит в строке с вещественными числами запятые на точки. Например, из числа 23,5 должно получиться число 23.5.

Этапы выполнения задания

I. Исходные данные: переменная st (введенная строка).

II. Результат: преобразованная строка.

III. Алгоритм решения задачи.

1. Вводим исходные данные.
2. Вычисляем длину строки.
3. В цикле **for** проверяем каждый символ строки st. Если текущий символ текста запятая, то за-

меняем его на точку. Другие символы оставляем без изменения.

4. Выводим результат.

IV. Описание переменных: *st* – string, *n* – integer.

Во всех текстовых редакторах реализована функция «заменить». При выполнении этой команды некоторые символы из строки удаляются, а вместо них вставляются другие символы.

Пример 9.7. Написать программу, которая заменит в тексте каждую цифру 2 словом «два».

Этапы выполнения задания

I. Исходные данные: переменная *st* (введенный текст).

II. Результат: преобразованный текст.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Поскольку удалять мы будем один символ, а вставлять три, то длина строки будет изменяться при обработке строки. Количество повторений цикла заранее не известно, поэтому будем использовать цикл **while**. Счетчик цикла будет изменяться от 1 до длины строки.

3. В цикле **while** проверяем каждый символ текста *st*.

3.1. Если текущий символ текста «2», то удаляем его и вставляем подстроку «два».

3.2. Переходим к следующему символу.

4. Вывод результата.

IV. Описание переменных: *st* – string, *i* – integer.

При правильном наборе компьютерного текста между любыми двумя

Пример 9.6.

V. Программа:

```
var st: string; n: integer;
begin
  writeln('Введите строку');
  readln(st);
  n := length(st);
  for var i := 1 to n do
    if (st[i] = ',') then
      st[i] := '.';
  writeln('Преобразованная строка: ');
  writeln(st);
end.
```

VI. Тестирование.

Окно вывода

```
Введите строку
23,5 34,71 89,234
Преобразованная строка:
23.5 34.71 89.234
```

Пример 9.7.

V. Программа:

```
var st: string;
    i: integer;
begin
  writeln('Введите строку');
  readln(st);
  i := 1;
  while i <= length(st) do
    begin
      if (st[i] = '2') then
        begin
          //замена символов
          delete(st, i, 1);
          insert('два', st, i);
        end;
        //переход к следующему символу
        i := i + 1;
      end;
    writeln('Преобразованная строка: ');
    writeln(st);
  end.
```

VI. Тестирование.

Окно вывода

```
Введите строку
В комнате стояли 2 стула и 2 стола.
Преобразованная строка:
В комнате стояли два стула и два стола.
```

Пример 9.8. Выделение лишних пробелов в Word.

При правильном наборе компьютерного текста между двумя словами должен быть только один пробел.

Пример 9.9.

V. Программа:

```
var st: string;
    i: integer;
begin
  writeln('Введите строку');
  readln(st);
  i := 1;
  while i < length(st) do
  begin
    if (st[i] = ' ') and
       (st[i+1] = ' ') then
      delete(st, i, 1)
    else
      i := i + 1;
  end;
  writeln('Преобразованная
    строка: ');
  writeln(st);
end.
```

VI. Тестирование. Введите текст «При правильном наборе компьютерного текста между любыми двумя словами должен быть только один пробел».

Результат:

Окно вывода

Введите строку
 При правильном наборе компьютерного текста между любыми двумя словами должен быть только один пробел.
 Преобразованная строка:
 При правильном наборе компьютерного текста между любыми двумя словами должен быть только один пробел.

словами должен быть только один пробел. Однако иногда случайно вставляют несколько пробелов. В этом случае Word подчеркивает их голубой волнистой линией (пример 9.8).

Пример 9.9. Написать программу, которая проверяет правильность расстановки пробелов в тексте и, если между словами более одного пробела, удаляет лишние.

Этапы выполнения задания

I. Исходные данные: переменная *st* (введенный текст).

II. Результат: преобразованный текст.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Поскольку длина строки будет изменяться при обработке строки, то количество повторений цикла заранее не известно. Будем использовать цикл **while**.

3. В цикле **while** проверяем соседние символы текста *st*.

3.1. Если оба соседних символа являются пробелами, то удалим один из них. Соседние символы имеют индексы, отличающиеся на один: *i* и *i + 1*. Поскольку в цикле есть обращение к элементу с номером *i + 1*, то условием выполнения цикла будет строгое неравенство $i < \text{length}(st)$.

3.2. Переходим к следующему символу только тогда, когда удаление не проводили.

4. Выводим результат.

IV. Описание переменных: *st* – string, *i* – integer.

Пример 9.10*. Написать программу, которая проверяет правильность расстановки пробелов вокруг тире. Если пробелы пропущены, то вставляет их. Предполагается, что в тексте нет слов, которые пишутся через дефис, двух знаков «-» подряд и лишних пробелов.

Этапы выполнения задания

I. Исходные данные: переменная *st* (введенный текст).

II. Результат: преобразованный текст.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Поскольку длина строки может изменяться при обработке строки, то количество повторений цикла заранее не известно. Будем использовать цикл **while**.

3. В цикле **while** проверяем каждый символ на совпадение с «-». При совпадении проверяем соседние символы.

3.1. Если соседний справа символ (*i* + 1) не пробел, то вставляем пробел.

3.2. Если символ слева (*i* - 1) не пробел, то вставляем пробел и увеличиваем *i* на 1.

3.3. Поскольку в цикле есть обращение к элементу с номером *i* + 1, то условием выполнения цикла будет строгое неравенство *i* < length(*st*).

3.4. Поскольку в цикле есть обращение к элементу с номером *i* - 1, то начальное значение *i* = 2.

Пример 9.10*.

V. Программа:

```
var st: string;
    i: integer;
begin
    writeln('Введите строку');
    readln(st);
    i := 2;
    while i < length(st) do
    begin
        if st[i] = '-' then
        begin
            if st[i+1] <> ' ' then
                insert(' ',st, i+1);
            if st[i-1] <> ' ' then
            begin
                insert(' ',st, i);
                i := i + 1;
            end
        end;
        i := i + 1;
    end;
    writeln('Преобразованная строка: ');
    writeln(st);
end.
```

VI. Тестирование. Введите текст «Жизнь прожить – не поле перейти. Родимая сторона – мать, чужая – мачеха. Окончил дело – гуляй смело».

Результат:

Окно вывода

```
Введите строку
Жизнь прожить – не поле перейти.
Родимая сторона – мать, чужая – мачеха.
Окончил дело – гуляй смело.
Преобразованная строка:
Жизнь прожить – не поле перейти.
Родимая сторона – мать, чужая – мачеха.
Окончил дело – гуляй смело.
```

VII. Анализ результата. В примере встречаются все четыре возможные ситуации: пробелов нет ни слева, ни справа от тире; пробел только слева; пробел только справа; пробелы с двух сторон. В результате выполнения все пробелы расставлены правильно.

Пример 9.11*.

V. Программа:

```

var st, sl: string;
    p: boolean; n: integer;
procedure DelSpace(var s: string);
begin
    while (s<>'') and (s[1]= ' ') do
        delete(s, 1, 1);
end;
function FirstWord(s: string):
    string;

var
    i: integer;
begin
    i := pos(' ', s);
    if i <> 0 then
        FirstWord := copy(s, 1, i - 1)
    else
        FirstWord := s;
end;
function CheckPalindrom
    (s: string): boolean;
var n: integer; f: boolean;
begin
    n := length(s); f := true;
    for var i := 1 to n div 2 do
        if s[i] <> s[n-i + 1] then
            f := false;
    CheckPalindrom := f;
end;
begin
    writeln('Введите st ');
    readln(st); p := false;
    while st <> '' do
        begin
            DelSpace(st);
            sl := FirstWord(st);
            if CheckPalindrom(sl) then
                begin
                    writeln(sl); p := true;
                end;
            n := length(sl);
            delete(st, 1, n);
        end;
        if p = false then
            writeln('Нет палиндромов');
    end.

```

4. Переходим к следующему символу.

5. Вывод результата.

IV. Описание переменных: st – string, i – integer.

Пример 9.11*. Написать программу, которая выведет слова-палиндромы¹ (слова, которые одинаково читаются слева направо и справа налево), входящие в заданный текст. Слова в тексте могут быть разделены одним или несколькими пробелами. Пробелы могут быть в начале и в конце текста.

Этапы выполнения задания

I. Исходные данные: переменная st (введенный текст).

II. Результат: слова-палиндромы.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Будем выделять первое слово из исходного текста, проверять его и затем удалять.

3. Логическая переменная p изменит значение с false на true, если будет выведен палиндром.

4. Поскольку длина строки может изменяться при обработке строки, то количество повторений цикла заранее не известно. Будем использовать цикл **while**.

5. В цикле **while**, до тех пор, пока строка не станет пустой, выполняем следующее.

5.1. Удалим пробелы в начале строки.

5.2. Выделим первое слово из строки.

¹ В мире интересных слов. Палиндромы. <http://www.tramvision.ru/words/pal.htm> (дата доступа: 08.01.2019).

5.3. Проверим выделенное слово и, если оно является палиндромом, то выведем его.

5.4. Удалим слово из строки.

6. Опишем три вспомогательных алгоритма:

6.1. Процедуру DelSpace для удаления пробелов в начале строки. Пробелы удаляются из строки только тогда, когда она не пуста. Параметр данной функции будет изменяться внутри функции и должен остаться измененным после ее завершения, поэтому при описании перед параметром стоит ключевое слово **var**. Если пробелов в начале строки нет, то процедура не изменит исходную строку.

6.2. Функцию FirstWord, которая скопирует из строки первое слово. Если в строке только одно слово (нет пробелов), то оно же и является первым.

6.3. Функцию CheckPalindrom для проверки, является ли слово палиндромом. Будем сравнивать первый символ с последним, второй с предпоследним и т. д. Символ с номером i будет сравниваться с символом с номером $(n - i + 1)$, где n — длина слова.

7. Вывод сообщения «Нет палиндромов» в случае, если значение p осталось false.

IV. Описание переменных: st, sl — string, i, n — integer, p — boolean.

Пример 9.11*. Продолжение.

VI. Тестирование. Введите текст:

«На берегу стоит шалаш из камыша. Для трафаретной печати предназначен мимеограф, или ротатор. Пшеничная лепешка наан является блюдом индийской национальной кухни».

Результат:

Окно вывода

Введите st
На берегу стоит шалаш из камыша.
Для трафаретной печати предназначен мимеограф, или ротатор. Пшеничная лепешка наан является блюдом индийской национальной кухни.
шалаш
или
наан

VII. Анализ результатов. Из введенной строки не вывелось слово «ротатор», которое тоже является палиндромом. Но после этого слова стоит не пробел, а точка. Поэтому для проверки функция CheckPalindrom получает слово «ротатор.», которое не является палиндромом.

Чтобы в качестве палиндромов учитывались слова, после которых стоят знаки препинания, изменим функцию CheckPalindrom:

```
function
CheckPalindrom(s: string):
var n: integer;
    z: string;
    f: boolean;
begin
    n := length(s);
    z := ',.;&!';
    if pos(s[n], z) <> 0 then
    begin
        delete (s, n, 1);
        n := n - 1;
    end;
    f := true;
    for var i := 1 to n div 2 do
        if s[i] <> s[n-i + 1] then
            f := false;
    CheckPalindrom := f;
end;
```

Пример 9.12. Примеры использования команд преобразования типов.

Преобразование числа к строковому представлению:

```
var a1: integer;
    a2: real;
    s1, s2: string;
begin
  a1 := 245; a2 := 3.7;
  s1 := IntToStr(a1);
  s2 := FloatToStr(a2);
  //выполним действия, чтобы
  //убедиться, что преобразование
  //типов произошло
  s1 := s1 + '1';
  writeln(s1);
  writeln(s2[2]);
end.
```

Окно вывода

Результат: 2451.
.

Преобразование строкового представления числа к числовому значению:

```
var a1:integer;
    a2: real;
    s1, s2: string;
begin
  s1 := '245'; s2 := '3.7';
  a1 := StrToInt(s1);
  a2 := StrToFloat(s2);
  //выполним действия, чтобы
  //убедиться, что преобразование
  //типов произошло
  a1 := a1 + 1;
  a2 := a2 + 0.1;
  writeln(a1, ' ', a2);
end.
```

Окно вывода

Результат: 246 3.8

При использовании процедур преобразования Str(v,s) и Val(s,v,er) тип числа определяется его записью.

Преобразование Str(v,s) возможно для любых доступных числовых типов.

9.3. Преобразование строк в числа и чисел в строки

Числовые данные используются для выполнения арифметических операций. Если символы цифр записаны в строковую переменную, то выполнять вычислительные действия с ними нельзя. Но можно преобразовывать строки, содержащие символы цифр, в числа и числа в строки, используя нижеперечисленные команды¹.

Команда	Описание
Функции преобразования числа a к строковому представлению	
FloatToStr(a)	Число a — вещественное
IntToStr(a)	Число a — целое
Функции преобразования строкового представления числа к числовому значению	
StrToFloat(s)	Строка s — запись вещественного числа
StrToInt(s)	Строка s — запись целого числа
Процедуры преобразования типов	
Str(v,s);	Преобразование числа в строку
Val(s,v,er);	Преобразование строки в число

Использование этих команд показано в примере 9.12.

В строковых переменных легко производить такие операции, как удаление, вставка или замена символа. Вставка, замена или удаление цифры

¹ Функции преобразования не работают с типом BigInteger.

из числа производятся сложнее. При необходимости число можно преобразовать в строку, выполнить необходимые действия и преобразовать строку обратно в число.

Пример 9.13. Написать программу, которая проверяет, является ли данный текст записью числа. В непустой текст могут входить только цифры или буквы. Если да, то найти сумму цифр данного числа, иначе вывести соответствующее сообщение.

Этапы выполнения задания

I. Исходные данные: переменная *st* (введенный текст).

II. Результат: сумма цифр или сообщение, что это не число.

III. Алгоритм решения задачи.

1. Вводим исходные данные.
2. Вычисляем длину строки.

3. К введенному тексту нельзя в явном виде применить функции преобразования типа, поскольку длина текста может быть больше 20, а простые числовые типы, содержащие такое количество цифр, Pascal не поддерживает. Поэтому будем пытаться преобразовывать в число каждый введенный символ, считать сумму и количество тех символов, которые удалось преобразовать.

4. Выполним инициализацию переменных: *s* := 0 (сумма цифр числа) и *k* := 0 (количество цифр).

5. В цикле **for** проверяем каждый символ строки *st*. Если текущий символ текста цифра, то преобразуем его в число, добавляем

Пример 9.12. Продолжение.

При использовании процедуры *Val(s,v,er)* сначала проверяется, возможно ли преобразование строковой записи в число в соответствии с типом. Если «да», то выполняется преобразование и переменная *er* получает значение 0 — код успешного преобразования, в противном случае значение *er* — это номер символа, который невозможно преобразовать. Вызов *Val('22.3',v,er)* присвоит переменной *v* значение 22.3, если она описана как *real* (*er* = 0); если она описана как *integer*, то *v* получит значение 22, *er* = 3 (символ '.' не может быть преобразован).

Пример 9.13.

V. Программа:

```
var st: string; n, k, s: integer;
begin
  writeln('Введите текст');
  readln(st);
  n := length(st); k := 0; s := 0;
  for var i := 1 to n do
  begin
    if (st[i] >= '0') and
      (st[i] <= '9') then
    begin
      k := k + 1;
      s := s + StrToInt(st[i]);
    end;
  end;
  if k = n then
    writeln('Сумма цифр =', s)
  else
    writeln('Текст не число');
  end.
```

VI. Тестирование. Введите текст 12345. Результат:

Окно вывода

```
Введите текст
12345
Сумма цифр = 15
```

Введите текст 123BC. Результат:

Окно вывода

```
Введите текст
123BC
Текст не число
```

Проверку того, что символ является цифрой, можно выполнить несколькими способами.

Если воспользоваться процедурой `val`, то фрагмент программы для проверки цифры будет таким:

```
for var i := 1 to n do
begin
  val(st[i], x, c);
  if c = 0 then
  begin
    k := k + 1;
    s := s + x;
  end;
end;
```

Проверить, является ли символ цифрой, можно аналогично тому, как в примере 9.5 выполнялась проверка гласных букв. Для этого нужно создать строку, в которую перечислить все цифры — `g:='0123456789'`;

Пример 9.14*.

V. Программа:

```
var st,sr,s1,s2,s3: string;
    a,b,c,n,r,r1,r2,p: integer;
begin
  writeln('Введите выражение');
  readln(st);
  sr := st + '=';
  //первое число a
  p := pos('(', st);
  s1 := copy (st, 1, p-1);
  a := StrToInt(s1);
  delete(st, 1, p);
  sr := sr + s1 + '*';
  //второе число b
  p := pos('+', st);
  s2 := copy (st, 1, p-1);
  b := StrToInt(s2);
  delete(st, 1, p);
  sr := sr + s2 + '+' + s1 + '*';
  //третье число c
  n := length (st);
```

число `k` сумме и увеличиваем счетчик количества преобразованных цифр. Если выполняется условие `(st[i]>='0') and (st[i]<='9')`, то символ строки является цифрой, поскольку символы цифр в таблице расположены последовательно.

6. Если количество символов, которые удалось преобразовать, равно длине строки, то выводим сумму цифр, иначе выводим соответствующее сообщение.

IV. Описание переменных: `st` — string, `n`, `s`, `k` — integer.

Пример 9.14*. Написать программу, которая раскрывает скобки в числовом выражении и вычисляет его значение. Выражение имеет вид $a(b + c)$ и вводится как строка. Вместо a , b и c — символы цифр, образующие целое число (количество цифр в каждом из них не более девяти). Вывести последовательность преобразований и результат. Например, для выражения $5(7 + 8)$ должны получить: $5(7 + 8) = 5 * 7 + 5 * 8 = 35 + 40 = 75$.

Этапы выполнения задания

I. Исходные данные: переменная `st` (текст).

II. Результат: числовое значение выражения.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Будем последовательно копировать из строки нужные символы и удалять те, которые уже обработали.

2.1. Находим символ «(», символы до него скопируем в переменную *s1* и преобразуем ее в число *a*. Удалим эти символы из строки.

2.2. Находим символ «+», символы до него скопируем в переменную *s2* и преобразуем ее в число *b*. Удалим эти символы из строки.

2.3. Из оставшейся строки скопируем в переменную *s3* все символы, кроме последнего — «)», и преобразуем в число *c*.

3. Текущий результат будем добавлять к новой строке, которой вначале присваивается введенная строка и символ «=». Текущие результаты вычислений ($r1 := a*b$ и $r2 := a*c$) будем преобразовывать в строковый тип и добавлять к строке *sr*.

4. Вычисляем значение выражения.

5. Выводим результат.

IV. Описание переменных: *st*, *sr*, *s1*, *s2*, *s3* — string, *a*, *b*, *c*, *n*, *r*, *r1*, *r2*, *p* — integer.

```
s3 := copy (st, 1, n-1);
c := StrToInt(s3);
sr := sr + s3 + '=';
//вычисление произведений
r1 := a * b;
sr := sr+IntToStr(r1)+'+';
r2 := a * c;
sr := sr+IntToStr(r2)+'=';
//вычисление результата
r := r1 + r2;
sr := sr + IntToStr(r);
writeln(sr);
```

end.

VI. Тестирование. Введите выражение $5(7 + 8)$.

Результат:

Окно вывода

Введите выражение

5 (7+8)

5 (7+8)=5*7+5*8=35+40=75

Введем выражение $12(234 + 802)$.
Получим следующий результат:

Окно вывода

Введите выражение

12 (234+802)

12 (234+802)=12*234+12*802=2808+9624=12432



Упражнения

- 1 Напишите программу, которая определит количество предложений в тексте. Предложение заканчивается одним из трех символов: «.», «?», «!». Предполагается, что в тексте есть хотя бы одно предложение (см. пример 9.2).
- 2 Напишите программу, которая определит количество слов в тексте, если между любыми двумя словами может быть более одного пробела. Предполагается, что в тексте есть хотя бы одно слово (см. пример 9.2).
- 3 Напишите программу, которая определит, каких букв в строке с русским текстом больше: «о» или «О» (см. пример 9.3).

- 4 Напишите программу, которая определит, какой процент составляют буквы «а» во введенном тексте (см. пример 9.3).
- 5 Напишите программу, которая определит, сколько слов в тексте начинается на букву «а».
- 6* Напишите программу, которая определит, какой процент слов в тексте начинается на букву «к». (Слово может начинаться как с прописной, так и со строчной буквы.)
- 7 Дан текст. Напишите программу, которая проверит, правильно ли в нем расставлены круглые скобки. Если нет, то вывести соответствующее сообщение: «Открывающихся скобок больше (меньше), чем закрывающихся»; «Закрывающиеся скобки раньше открывающихся скобок».
- 8 В тексте могут встречаться гласные и согласные буквы, а также символы «ь» и «ъ». Измените программу из примера 9.5 так, чтобы символы «ь» и «ъ» выводились желтым цветом.
- 9 Дано арифметическое выражение, состоящее из цифр, скобок и знаков арифметических действий. Напишите программу, которая выведет цифры синим цветом, а остальные символы — голубым: например, в выражении $2 + (3 - 5) * 7 - 13$ (см. пример 9.5).
- 10 Напишите программу для решения задачи. Задана строка цифр. Вывести четные цифры синим цветом, а нечетные — голубым (например, 128235). Сколько в строке нечетных цифр? (См. примеры 9.3 и 9.5.)
- 11* Вводится текст, слова в котором разделены пробелами, после слов могут стоять точки или запятые. Напишите программу, которая выведет синим цветом те буквы «а», которые являются последними буквами слова, остальные символы текста вывести голубым цветом (например, в скороговорке На дворе — трава, на траве — дрова). Какой процент от общего количества слов составляют слова, заканчивающиеся на букву «а»?
- 12 Напишите программу, которая заменит в заданном тексте каждую букву «а» символом «*» (см. пример 9.6).
- 13 Напишите программу, которая заменит в заданном тексте каждую цифру символом «?» (см. примеры 9.5 и 9.6).
- 14 Напишите программу, которая заменит в заданном тексте из латинских букв все вхождения «х» на «ks» (см. пример 9.7).
- 15 Напишите программу, которая заменит в заданном тексте из латинских букв все вхождения «ing» на «ed» (см. пример 9.7).
- 16* Напишите программу для решения задачи. В заданном тексте заменить все слова A1 на слова A2 (слова в тексте разделены пробелами, слова A1 и A2 вводятся).

- 17 Напишите программу, которая удалит из текста все гласные буквы (см. примеры 9.5 и 9.9).
- 18 Напишите программу, которая удалит из текста все знаки «+», непосредственно за которыми стоит не цифра.
- 19 Напишите программу, которая в заданном тексте после каждой латинской буквы «q» добавит букву «u» (см. пример 9.10).
- 20 Напишите программу, которая в заданном тексте после каждого знака препинания («.», «;», «:», «;») вставит пробел, если его там нет (см. пример 9.10).
- 21 Измените функцию `CheckPalindrom` из примера 9.11 так, чтобы слова, которые начинаются на заглавную букву, тоже считались палиндромами, например «Анна», «Алла».
- 22 Добавьте в программу из примера 9.11 подсчет количества выведенных палиндромов.
- 23* Фразы-палиндромы читаются одинаково слева направо и справа налево без учета пробелов и знаков препинания. Например: «Кулинар, храни лук» или «А роза упала на лапу Азора». Напишите программу, которая определит, является ли фраза палиндромом.
- 24 Напишите программу, которая проверяет, является ли данный текст записью числа. В непустой текст могут входить только цифры или буквы. Если да, то требуется проверить, делится ли данное число на 4, иначе вывести соответствующее сообщение. Для проверки делимости на 4 использовать признак делимости: число делится на 4, если двузначное число, состоящее из последних двух цифр исходного числа, делится на 4 (см. пример 9.13).
- 25 Измените программу из упражнения 20 так, чтобы проверялась делимость на 2, 3, 5, 6, 8, 12 (используйте соответствующие признаки делимости).
- 26 Дан текст. Напишите программу, которая проверит, может ли быть этот текст записью вещественного числа.
- 27 Напишите программу для решения задачи. Строка представляет собой запись следующего вида: « $a \pm b$ ». Найти значение выражения. Вместо знака « $\pm$ » может быть знак «+» или знак «-». Числа a и b являются целыми и состоят не более чем из девяти цифр (см. пример 9.14).
- 28 Напишите программу для решения задачи. Строка представляет собой запись следующего вида: « $(a + b) / c$ ». Выделить из записи числа и найти целочисленное значение выражения и остаток от деления. Числа, входящие в выражение, являются целыми и состоят не более чем из девяти цифр (см. пример 9.14).

Глава 3

ОБРАБОТКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

§ 10. Понятие электронной таблицы

Пример 10.1. Таблица учета товара.

Продукт	Цена, р.	Получено, шт.	Продано, шт.	Осталось, шт.	Выручка, р.
Булочка	1,2	100	62	38	74,4
Йогурт	0,6	75	59	16	35,4
Желе	0,7	80	42	38	29,4
Сок	0,85	200	84	116	71,4

Учет по данной таблице может проводиться несколько раз в день. Для этого каждый раз придется вносить изменения и менять не только исходные данные, но и значения в вычисляемых столбцах. При создании таблицы с помощью текстового редактора расчеты придется выполнять вручную или с помощью калькулятора.

Первые аналоги электронных таблиц описаны профессором Ричардом Маттесичем (работал в Калифорнийском университете в 1961—1964). В них говорилось о компьютерных моделях составления бюджета. Программа была написана на языке Фортран IV.

Первая электронная таблица для ПК Apple создана Дэном Бриклин-ом при обучении в Гарвардской школе экономики. Программа VisiCalc (Visible Calculator) была написана сначала на Бейсике, а затем переписана на Ассемблере. Она успешно продавалась до 1985 г., пока не была вытеснена Lotus 1-2-3¹.

10.1. Назначение и типовые задачи обработки информации в электронных таблицах

Продавцы в магазине подсчитывают количество полученного, проданного и оставшегося товара, выручку от продажи. Форма учета товара может иметь вид, как показано в примере 10.1. Данные для расчетов представлены в виде таблицы. **Таблица** — это данные, структурированные в определенном порядке, разделенные по строкам и столбцам, на пересечении которых образуются ячейки.

Проанализировав данные в столбцах **Осталось** и **Выручка**, можно заметить, что значения в них получаются через значения в других столбцах (**Осталось** = **Получено** – **Продано**, **Выручка** = **Продано** * **Цена**). Такие столбцы называют **вычисляемыми**. Столбцы **Цена**, **Получено** и **Продано** содержат **исходные данные** для расчетов. Вычисляемые значения могут изменяться автоматически при изменении исходных данных, если для вычислений использовать **электронную таблицу**.

Электронная таблица — таблица, представленная в электронном виде, в которой значения в вычисляемых ячейках могут автоматически пересчитываться при изменении значений исходных данных.

¹ <http://eakolesnikov.narod.ru/> Колесников Е. А. «Эволюция электронных таблиц» (дата доступа 05.01.2019).

Электронные таблицы позволяют обрабатывать большие массивы числовых данных (например, статистические данные, результаты экспериментов), производить финансовые и бухгалтерские расчеты и многое другое.

Прикладные программы для работы с электронными таблицами называют **табличными процессорами**.

Часто табличные процессоры называют **электронными таблицами**. Существует достаточно много вариантов табличных процессоров (пример 10.2). Все они позволяют решать типовые задачи обработки информации в электронных таблицах (пример 10.3).

10.2. Структура электронной таблицы

Табличные процессоры во многом похожи друг на друга, поэтому, изучив принципы работы с одним из них, легко освоить работу и с другими. Рассмотрим табличный процессор Microsoft Excel. Файлы, созданные с его помощью, имеют значок  и расширение .xlsx. Окно программы приведено в *Приложении 3* (с. 163).

Новый документ, который автоматически открывается при запуске Excel, называется **книга**. Каждая книга состоит из листов. **Лист** имеет вид таблицы и предоставляет место для хранения данных и их обработки. Все листы одной книги сохраняются в одном файле. Выполнять действия с листами можно с помощью контекстного меню (пример 10.4).

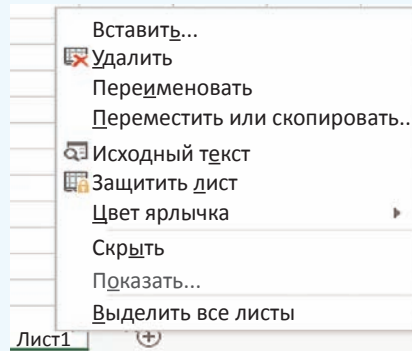
Пример 10.2. Наиболее популярные электронные таблицы.

Название	Логотип	ОС
Microsoft Excel		Windows, Mac OS X
LibreOffice Calc		Windows, Mac OS X, Linux
OpenOffice Calc		Windows, Mac OS X, Linux
Numbers		Mac OS X
Google Документы		Веб-приложение

Пример 10.3. Задачи обработки информации в электронных таблицах.

1. Создание, редактирование и форматирование электронной таблицы.
2. Сохранение электронной таблицы на диске в виде файла.
3. Загрузка электронной таблицы из файла в оперативную память компьютера.
4. Вывод данных из электронной таблицы на бумагу.
5. Графическое представление числовых данных в виде диаграмм.

Пример 10.4. Контекстное меню листа электронной таблицы.



Пример 10.5. Структурные элементы электронной таблицы:

	A	B	C	D
1				
2				
3				
4				
5				

Строка Ячейка Столбец

В электронной таблице на рабочем листе 1 048 576 строк и 16 384 столбцов. Для обозначения столбцов используют буквы латинского алфавита, которых всего 26. Поэтому, начиная с 27 столбца, используются двухбуквенные номера: AA, AB, AC, ..., AZ, BA, BB, ..., BZ, CA, ..., а затем и трехбуквенные: AAA, AAB, Последний столбец имеет наименование XFD.

Количество листов в книге ограничено объемом доступной оперативной памяти (по умолчанию в книге 1 лист).

Пример 10.6. Примеры выделенных диапазонов.

Диапазон B2:D5:

	A	B	C	D	E
1					
2					
3					
4					
5					
6					

Диапазон «A2:B3;D2:D5;A7:D7» состоит из трех выделенных областей:

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					

Диапазоны могут выделяться на разных листах книги. В этом случае перед диапазоном дописывается номер листа: «Лист1!A1:B4;Лист2!D2:F6».

Основные структурные элементы электронной таблицы — **строки, столбцы и ячейки** (пример 10.5). Строки обозначаются числами, столбцы — латинскими буквами. Каждая ячейка получает **адрес**, состоящий из названия столбца и номера строки, на пересечении которых она находится. Одна из ячеек (например, E23 или ML123) выделяется с помощью курсора рамкой. Выделенная ячейка называется **активной**. Именно в ней можно осуществлять ввод информации. В примере 10.5 активной является ячейка C3. Любую ячейку можно сделать активной, переместив на нее курсор с помощью клавиш управления курсором или щелкнув по ней мышью.

Область из нескольких выделенных ячеек образует **диапазон**. Адрес диапазона определяется адресом верхней левой и нижней правой ячеек выделенной области, например: B2:D5. (пример 10.6). Выделять ячейки можно мышью или используя клавишу Shift и клавиши управления курсором.

Диапазон могут образовывать несмежные ячейки таблицы. В этом случае адрес состоит из адресов нескольких прямоугольных областей, разделенных точкой с запятой. Для выделения такого диапазона используют левую клавишу мыши при нажатой клавише Ctrl.

10.3. Типы данных в электронной таблице

Всю информацию, которую вводят в электронную таблицу, размещают в ячейках. В ячейку таблицы можно

поместить данные одного из трех типов: **число**, **текст** или **формулу** (пример 10.7). Тип данных определяет возможности их использования для дальнейшей работы.

Excel автоматически пытается определить тип данных, вводимых в ячейку таблицы. Ввод формул начинается со знака «=». Если данные определились как число, то в ячейке числа выравниваются по правому краю. Все, что не является формулой и числом, является текстом и в ячейке выравнивается по левому краю. По желанию пользователь может установить другое выравнивание в ячейках.

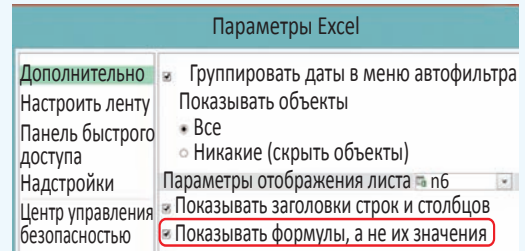
Числа и тексты — исходные данные. Изменяются они путем редактирования соответствующих ячеек. Значения в ячейках, содержащих формулы, изменяются автоматически, как только изменились хотя бы какие-то исходные данные, входящие в формулу. Для этого в формулах используют не сами исходные данные, а ссылки на адреса ячеек, в которых они находятся.

В ячейках с формулами отображаются не сами формулы, а результаты вычислений по ним. Формула в активной ячейке отображается в **строке формул**.

Пример 10.7. Типы данных в электронной таблице.

D3	:	X	✓	f _x	=C3*B3
	A	B	C	D	E
1			Прямоугольник		
2		Длина	Ширина	Площадь	
3		2	7	14	
4					

Для отображения формул в ячейке можно перейти в режим отображения формул с помощью команды **Файл** → **Параметры** → **Дополнительно**, установить флажок **Показывать формулы**, а не их значения в разделе **Параметры отображения листа**.



Для перехода в режим отображения значений нужно сбросить установленный флажок. Переход из одного режима в другой может осуществляться комбинацией клавиш **Ctrl + `** (верхняя левая алфавитно-цифровая клавиша, в русской раскладке — буква «ё»).

D3	:	X	✓	f _x	=C3*B3
	A	B	C	D	
1			Прямоугольник		
2		Длина	Ширина	Площадь	
3		2	7	=C3*B3	
4					



1. Что понимают под электронной таблицей?
2. Перечислите типовые задачи обработки информации в электронных таблицах.
3. Из каких структурных элементов состоит электронная таблица?
4. Как задаются имена для строки, столбца, ячейки?
5. Что такое диапазон ячеек? Как указывается адрес диапазона?
6. Какие типы данных могут использоваться в электронных таблицах?



Упражнения

1 Назовите адрес активной ячейки. Какого типа информация хранится в активной ячейке? Как это можно определить?

1

B2				fx		b	
A		B		C		D	
1		Сложение трех чисел					
2		a		b		c	
3		54		-23		38	
						сумма	
						69	

2

B3		:			<i>f_x</i>	-23
	A	B	C	D		
1	Сложение трех чисел					
2	a	b	c	сумма		
3	54	-23	38	69		

3

D3		:			f_x	=A3+B3+C3	
	A		B		C	D	E
1	Сложение трех чисел						
2	a		b		c	сумма	
3		54		-23		38	69

4

D3

:

✕

✓

fx

=A3+B3+C3

	A	B	C	D
1	Сложение трех чисел			
2	a	b	c	сумма
3	54	-23	38	=A3+B3+C3

2 Рассмотрите таблицы. Назовите адреса ячеек, содержащих текст, числа. Есть ли в таблицах вычисляемые ячейки? Какие? Обоснуйте свой ответ.

1

		A		B		C		D		E	
				Название продукта		Цена		Количество		Стоимость	
1		№									
2		1		Молоко		1,17		1		1,17	
3		2		Хлеб		0,84		2		1,68	
4		3		Шоколад		1,54		2		3,08	
5		4		Квас		1,23		1		1,23	
6		5		Яблоко		0,45		5		2,25	
7		6		Йогурт		0,89		4		3,56	

2

		A		B	
1		Население Беларуси			
2		Год		Численность населения (тыс. чел.)	
3		1897		6673	
4		1939		8912	
5		1950		7745	
6		1960		8190	
7		1970		8992	
8		1980		9658	
9		1990		10189	
10		2000		10002	
11		2010		9500	
12		2018		9491	

3 Откройте файл с таблицами, рассмотренными в упражнениях 1 и 2. Таблицы находятся на листах с соответствующими названиями. Выполните указанные действия.

1. Проверьте правильность своих ответов.
2. Для таблицы из упражнения 1 и для таблицы 1 из упражнения 2 поменяйте исходные данные, проследите за изменениями в вычисляемых ячейках.
3. Поменяйте местами листы с таблицами.

4. Скопируйте лист с таблицей из упражнения 1 и измените его название на «Сложение».

5. Добавьте новый лист и переименуйте его.

4 Подготовьте сообщение на одну из перечисленных тем.

1. «История электронных таблиц».

2. «Облачные электронные таблицы».

§ 11. Создание электронной таблицы

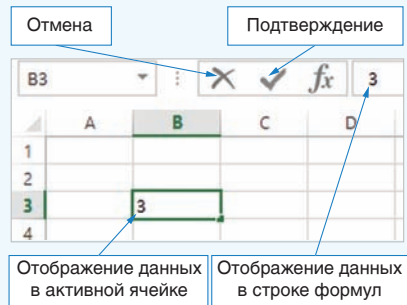
11.1. Ввод данных

Вводить данные в таблицу можно непосредственно в ячейке или в **строке формул** (пример 11.1). Для ввода информации в ячейку необходимо ее активировать, а затем ввести данные с клавиатуры. После внесения данных нужно подтвердить действие — нажать клавишу Enter или активировать другую ячейку. Для подтверждения (или отмены) ввода данных можно использовать кнопку  () в строке формул.

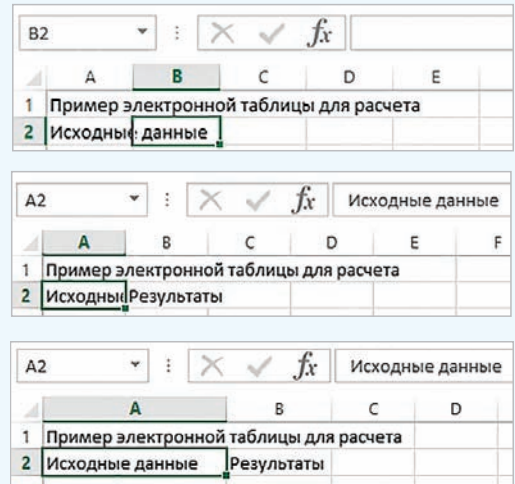
11.2. Текст

При вводе текст может целиком не помещаться в ячейку, тогда он будет зрительно занимать и соседние ячейки, до тех пор пока в них не будет введена информация. В примере 11.2 в ячейку A2 введен текст «Исходные данные». Затем в ячейку B2 введен текст «Результаты». Часть текста из ячейки A2 исчезла с экрана, но в самой ячейке текст полностью сохранился, в чем легко убедиться, если активировать ячейку A2 и посмотреть на строку формул. Если увеличить ширину столбца A, то текст будет виден в ячейке целиком.

Пример 11.1. Ввод данных:



Пример 11.2. Ввод текста.



Для автоподбора ширины столбца достаточно выполнить двойной щелчок мышью по правой границе соответствующего столбца.

Пример 11.3. Числовые данные.

	A	B
1	Числовой вид	235,698
2	Экспоненциальный вид	2,35698E+02

Число 235,698 можно записать как $2,35698 \cdot 10^2$. В этой записи сомножитель 2,35698 является мантиссой, а показатель степени 2 — порядком. Число $2,35698 \cdot 10^2$ в экспоненциальном виде будет записано как 2,35698E + 2.

Иногда после ввода чисел ячейка может оказаться заполненной знаками #:

	A	B
1	Числовой вид	235,698
2	Экспоненциальный вид	#####

Это означает, что ширины столбца недостаточно для отображения числа. Нужно увеличить ширину столбца так, чтобы в ячейке показалось само число.

Одной из причин появления экспоненциальной формы записи числа была необходимость работать с очень большими или очень маленькими числами. Например:

масса Земли приблизительно равна 5 970 000 000 000 000 000 000 000 г,

масса атома водорода составляет 0,000 000 000 000 000 000 000 001 67 г.

В экспоненциальной записи эти числа будут выглядеть так: $5,97 \cdot 10^{27}$ и $1,67 \cdot 10^{-24}$.

Пример 11.4. Обыкновенные дроби.

A1		:			f_x	3,55555555555556	
	A	B	C	D	E	F	
1	35/9						
2	1/2						

Если дробь является обыкновенной, то вначале необходимо ввести 0, затем пробел, а затем дробную часть.

11.3. Числовые данные

В электронных таблицах можно производить вычисления с **целыми** и **действительными** числами. Числовыми данными также являются дата и время.

Действительные числа можно записывать двумя способами: в **числовой** или в **экспоненциальной** форме. Числовая форма предполагает наличие у числа целой и дробной части, разделенных запятой. В экспоненциальной форме число представляется в виде **мантиссы**, умноженной на 10 в степени, равной **порядку** (пример 11.3). При записи таких чисел в ячейку электронной таблицы сначала располагается мантисса, затем буква E (латинская), после нее — порядок. Excel автоматически приводит к экспоненциальному виду очень большие или очень маленькие по значению числа. Если вы введете в ячейку таблицы 10 000 000 000, то это число автоматически преобразуется в форму 1E+10, а число 0,0000000001 — в форму 1E-10.

В ячейку таблицы можно ввести смешанную дробь. Например, для ввода дроби $3\frac{5}{9}$ нужно набрать 3, затем «пробел» и 5/9. В строке формул такое число будет отображаться в виде десятичной дроби (пример 11.4).

11.4. Последовательности

Часто при работе с электронными таблицами приходится работать с последовательностями чисел, дат, тек-

стов. Например, необходимо пронумеровать строки или столбцы, ввести все числа месяца или названия дней недели. Многие такие последовательности табличный процессор Excel распознает автоматически. Достаточно ввести первые элементы последовательности, а затем с помощью маркера заполнения расширить диапазон в строке или столбце. **Маркер заполнения** — маленький квадратик в нижнем правом углу активной ячейки или выделенного диапазона (пример 11.5). При наведении указателя мыши на маркер заполнения указатель принимает вид: **+**.

Для ввода последовательности нужно выполнить следующие действия:

1. Ввести начало последовательности (обычно достаточно заполнить одну, две или три ячейки).
2. Выделить маркер заполнения.
3. Переместить маркер заполнения, удерживая нажатой левую кнопку мыши.

По мере заполнения ячеек будут появляться подсказки с данными, которые будут введены в последнюю выделенную ячейку (пример 11.6, 11.7).

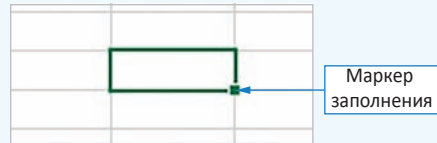
11.5. Формулы

Все вычисления в Excel производятся по формулам.

Формула — запись выражения, по которому выполняются вычисления.

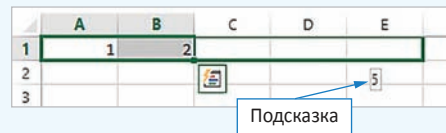
В формулы могут входить числа, адреса ячеек и диапазонов, функции, знаки арифметических операций,

Пример 11.5. Маркер заполнения.



Пример 11.6. Ввести последовательность чисел от 1 до 10 в первую строку электронной таблицы:

1. Ввести число 1 в ячейку A1.
2. Ввести число 2 в ячейку B2.
3. Выделить диапазон A1:B2.
4. Используя маркер заполнения, растянуть диапазон до ячейки J1 (в подсказке появится число 10).



Результат:

	A	B	C	D	E	F	G	H	I	J
1	1	2								
2										

Пример 11.7. Ввести последовательность месяцев года:

1. Ввести в ячейку A1 слово «январь».
2. Выделить ячейку A1 и протянуть маркер заполнения вправо, например до E1.

	A	B	C	D
1	январь			
2				
3				

Диапазон A1:E1 автоматически заполнится названиями месяцев от января до мая.

Продолжить заполнять строку названиями месяцев всего года.

	A	B	C	D	E
1	январь	февраль	март	апрель	май

Пример 11.8. Запись формул в Excel.

Формула	Запись в Excel
$(A_1 - 2A_2)^2$	<code>=(A1-2*A2)^2</code>
$\frac{B_1}{A_1 - 1}$	<code>=B1/(A1-1)</code>

Excel поддерживает и другой стиль ссылок: R1C2 — строка (row) 1, столбец (column) 2. Для переключения к такому стилю нужно выполнить: **Файл** → **Параметры** → **Формулы**, раздел **Работа с формулами**, установить флажок **Стиль формул R1C1**.

Работа с формулами
☒ **Стиль ссылок R1C1**
Пример 11.9. Ввод формул в Excel.

A2	:	X	✓	f_x	=B2/A2
	A	B	C	D	E
1	A	B	Расчет		
2	61	13	=B2/A2		

Для ввода ссылки удобно пользоваться мышью. Если при вводе формулы щелкнуть левой клавишей мыши по ячейке, то в формулу будет введен адрес этой ячейки. Аналогично можно выбрать адрес ячейки с помощью клавиш управления курсором. При этом ячейка и ссылка на нее будут подсвечены одним цветом. На рисунке отмечен момент ввода ссылки A2.

Пример 11.10. Редактирование данных в Excel.

	A	B
1		
2		
3		3

Текстовый курсор в ячейке

X	✓	f_x	3
---	---	-------	---

Текстовый курсор в строке формул

скобки (пример 11.8). Для обозначения арифметических операций используют знаки: +, -, *, /. Для возведения в степень используется знак ^. Порядок выполнения действий при расчетах по формуле:

1. Вычисление значений функций и выполнение действий в скобках.
2. Возведение в степень.
3. Умножение и деление.
4. Сложение и вычитание.

Адрес ячейки, входящий в формулу, называют **ссылкой**.

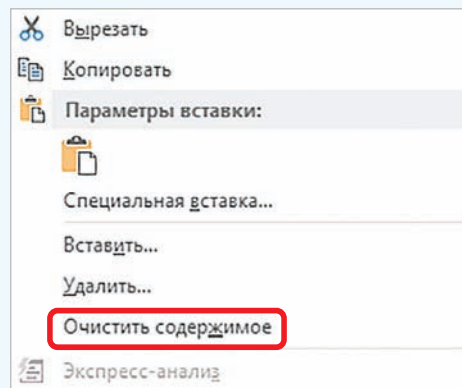
Для записи ссылок могут использоваться как прописные (большие), так и строчные (малые) латинские буквы. Ввод формулы начинается со знака «=» и заканчивается нажатием клавиши Enter или щелчком мыши по кнопке ☒ в строке формул (пример 11.9).

11.6. Редактирование данных и формул

Редактировать введенную в ячейку информацию можно двумя способами: непосредственно в ячейке и используя строку формул. Для этого в активной ячейке нужно нажать клавишу F2 или выполнить двойной щелчок мышью по ячейке. В ячейке появится текстовый курсор, используя который можно изменять данные в ячейке. Для редактирования информации с помощью строки формул нужно сделать ячейку активной, а затем щелкнуть левой клавишей мыши по строке формул (пример 11.10).

Для замены информации в ячейке достаточно сделать ее активной, а затем ввести новую информацию. Для удаления всей информации из ячейки также нужно сделать ее активной, а затем нажать клавишу Delete или выбрать команду **Очистить содержимое** в контекстном меню ячейки (пример 11.11). Если требуется изменить введенную формулу, то ее редактируют так же, как данные в ячейке.

Пример 11.11. Контекстное меню ячейки.



1. Как ввести целое число?
2. Как ввести дробь в ячейку электронной таблицы?
3. Чем отличаются числовой и экспоненциальный форматы числовых данных?
4. Из каких элементов состоит формула в электронной таблице?
5. Как ввести формулу в ячейку электронной таблицы?
6. Что такое ссылка?
7. Как ввести последовательности в ячейки электронной таблицы?
8. Как изменить данные в ячейке электронной таблицы?



Упражнения

1. Создайте электронную таблицу, изображенную на рисунке.

В ячейках A3, A7, A11, B11 находятся числа. Введите следующие формулы:

- в ячейку B3: $=4*A3$
- в ячейку C3: $=A3*A3$
- в ячейку B7: $=6*A7*A7$
- в ячейку C7: $=A7*A7*A7$
- в ячейку C11: $=(A11^2+B11^2)^{0,5}$
- в ячейку D11: $=A11+B11+C11$
- в ячейку E11: $=A11*B11/2$

Измените величину стороны квадрата и проследите за тем, как будут изменяться значения его периметра и площади. Сохраните работу.

	A	B	C	D	E
1	Квадрат				
2	Сторона	Периметр	Площадь		
3	3	12	9		
4					
5	Куб				
6	Ребро	Площадь	Объем		
7	3	54	27		
8					
9	Прямоугольный треугольник				
10	Катет	Катет	Гипотенуза	Периметр	Площадь
11	3	4	5	12	6

2 Электронная таблица создана для решения следующей задачи: «В растворе массой 500 г содержится нитрат натрия NaNO_3 массой 25 г. Определить массовую долю нитрата натрия в этом растворе».

Введите в ячейку A6 формулу для вычисления массовой доли нитрата натрия в этом растворе:

$\omega(\text{вещества}) = \frac{m(\text{вещества})}{m(\text{раствора})}$. Чтобы выразить полученную величину в процентах, ее нужно умножить на 100.

3 Преобразуйте таблицу из предыдущего упражнения для решения следующей задачи: «В воде массой 40 г растворили сахар массой 10 г. Определить массовую долю сахара в этом растворе».

4 Создайте электронную таблицу для решения следующей задачи: «При спокойном дыхании в легкие человека поступает около 500 см^3 воздуха. Состав вдыхаемого воздуха распределяется следующим образом: O_2 — 20,94 %, CO_2 — 0,03 %, N_2 — 79,03 %. Определить, сколько кубических сантиметров каждого газа вдыхает человек».

5 Путем ввода последовательностей создайте таблицу:

	A	B	C	D	E
1	Понедельник	Вторник	Среда	Четверг	Пятница
2					

6* Путем ввода последовательностей создайте следующую таблицу сложения:

	A	B	C	D	E	F	G	H	I	J	K
1	+	1	2	3	4	5	6	7	8	9	10
2	1	2	3	4	5	6	7	8	9	10	11
3	2	3	4	5	6	7	8	9	10	11	12
4	3	4	5	6	7	8	9	10	11	12	13
5	4	5	6	7	8	9	10	11	12	13	14
6	5	6	7	8	9	10	11	12	13	14	15
7	6	7	8	9	10	11	12	13	14	15	16
8	7	8	9	10	11	12	13	14	15	16	17
9	8	9	10	11	12	13	14	15	16	17	18
10	9	10	11	12	13	14	15	16	17	18	19
11	10	11	12	13	14	15	16	17	18	19	20

§ 12. Редактирование электронной таблицы

12.1. Объединение ячеек

Несколько рядом стоящих выделенных ячеек можно объединить в одну. Для этого после выделения диапазона объединяемых ячеек можно:

1. Воспользоваться кнопкой **Объединить и поместить в центре**  на вкладке **Главная** (пример 12.1).

2. Выполнить команду **Формат ячеек...** Команду можно выбрать в контекстном меню ячейки или на вкладке **Главная** в разделе **Выравнивание** (кнопка ). На вкладке **Выравнивание** поставить флажок **Объединение ячеек** (пример 12.2).

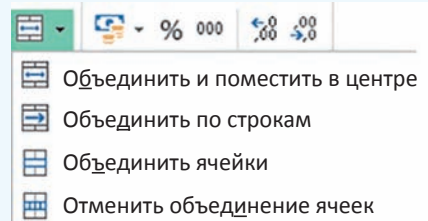
12.2. Операции с диапазонами ячеек

Выделенные диапазоны ячеек можно очищать, вырезать и копировать в буфер обмена, а также вставлять из буфера обмена.

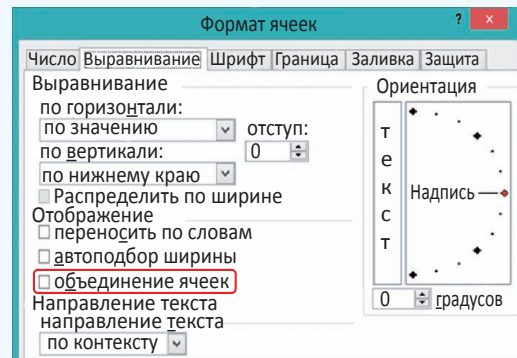
Очистка диапазона ячеек производится так же, как и очистка содержимого ячейки: выделить диапазон и нажать клавишу Delete. Можно воспользоваться командой **Очистить содержимое** из контекстного меню.

Команды **Копировать**, **Вырезать**, **Вставить** можно выбрать на ленте **Главная** (пример 12.3), из контекстного меню или использовать быстрые клавиши: **CTL + C** (копировать), **CTL + X** (вырезать), **CTL + V** (вставить).

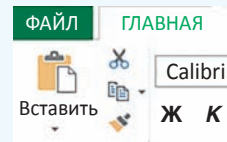
Пример 12.1. Кнопка **Объединить и поместить в центре** с выпадающим списком.



Пример 12.2. Вкладка **Выравнивание** окна **Формат ячеек**.

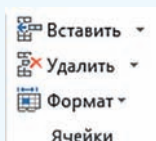


Пример 12.3. Команды для вырезания, копирования и вставки на вкладке **Главная**.

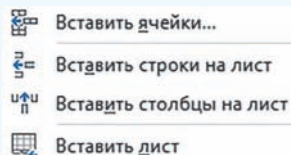


Для перемещения выделенного диапазона ячеек с помощью мыши нужно установить указатель мыши на границу выделенной области так, чтобы он приобрел вид . Затем следует переместить область, удерживая нажатой левую клавишу мыши. Копирование диапазона можно осуществить при нажатой клавише Ctrl аналогично перемещению.

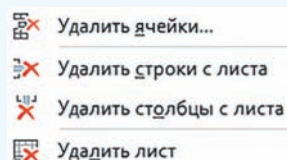
Пример 12.4. Команды вставки и удаления на вкладке **Главная**.



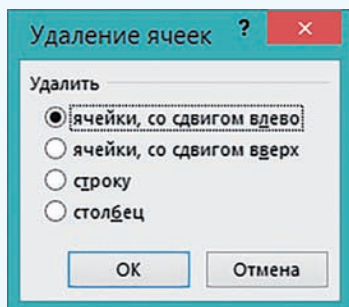
Выпадающий список **Вставить**.



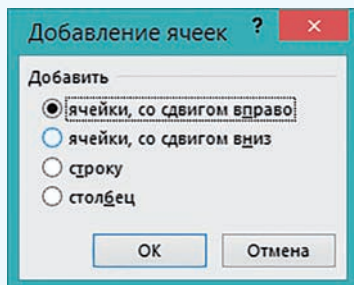
Выпадающий список **Удалить**.



Пример 12.5. Меню удаления ячеек.



Пример 12.6. Меню вставки ячеек.



12.3. Вставка и удаление строк и столбцов

Для управления вставкой или удалением ячеек можно использовать раздел **Ячейки** на вкладке **Главная** (пример 12.4) или команды контекстного меню.

Чтобы удалить строку (столбец), нужно ее (его) выделить, а затем выполнить команду **Удалить**. Для выделения строки (столбца) следует кликнуть по ее (его) заголовку. Удаление отдельных ячеек также выполняется с помощью команды **Удалить**. При удалении выделенных ячеек открывается окно **Удаление ячеек**, в котором пользователь должен уточнить, куда сдвигать остальные ячейки таблицы: влево или вверх (пример 12.5).

Удалять строки, столбцы и ячейки нельзя с помощью клавиши Delete, так как эта клавиша предназначена только для очистки содержимого столбца, строки или ячейки.

Для вставки строки (столбца) нужно выполнить команду **Вставить**. При вставке отдельных ячеек пользователь должен указать, куда сдвигать остальные ячейки: вправо или вниз (пример 12.6). Если требуется вставить несколько строк (столбцов), то в таблице нужно выделить столько строк (столбцов), сколько их следует вставить. Все строки (столбцы) вставятся перед первой выделенной строкой (столбцом).

12.4. Изменение ширины столбца и высоты строки

Изменение ширины столбца или высоты строки можно осуществлять разными способами. Если нужно

установить точное значение ширины столбца (высоты строки), то выполняют команду **Формат** в разделе **Ячейки** вкладки **Главная** (пример 12.7). Затем выбирают соответствующую команду для ввода точного значения ширины (высоты).

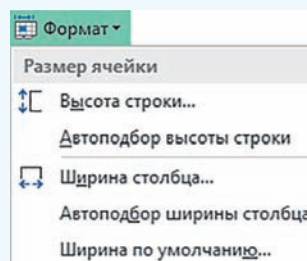
Если требуется установить такую ширину столбца (высоту строки), чтобы в ячейке отображалось все ее содержимое, нужно выполнить команду **Автоподбор ширины столбца** или **Автоподбор высоты строки**.

Изменять ширину столбца или высоту строки можно также с помощью мыши (пример 12.8).

Чтобы изменить ширину (высоту) нескольких столбцов (строк), их предварительно выделяют, а затем производят одно из вышеописанных действий.

Команда **Ширина по умолчанию** позволяет задать ширину для всех столбцов электронной таблицы.

Пример 12.7. Команда **Формат** на вкладке **Главная**.



Пример 12.8. Чтобы изменить ширину столбца или высоту строки с помощью мыши, нужно установить указатель мыши на правую границу заголовка столбца (нижнюю границу заголовка строки). Указатель мыши должен принять вид двусторонней стрелки:  / . После этого следует растянуть столбец (строку) до нужной ширины (высоты). Если по правой границе столбца (нижней границе строки) выполнить двойной щелчок, то будет произведен автоматический подбор ширины (высоты).



1. Как объединить ячейки таблицы?
2. Какие действия можно выполнять над выделенными диапазонами ячеек?
3. Как удалить строку, столбец, ячейку электронной таблицы?
4. Как можно вставить строку, столбец, ячейку в электронную таблицу?



Упражнения

1 Откройте таблицу, созданную в упражнении 1 после § 11. Произведите в таблице указанные изменения.

1. Формулу в ячейке C3 замените на формулу $A3^2$.
2. Формулу в ячейке C7 замените на формулу $A7^3$.
3. Перед строкой 5 вставьте еще 4 строки, добавьте информацию для вычисления периметра и площади прямоугольника со сторонами 8 и 5, вписав формулы самостоятельно.
4. Добавьте столбец перед первым и пронумеруйте в нем фигуры.
5. Подберите подходящую ширину первого столбца.

6. Добавьте строку перед первой, впишите в ней заголовок таблицы: «Геометрические фигуры». Разместите заголовок по центру, объединив ячейки A1:F1.

7. Объедините ячейки для размещения названий геометрических фигур.

	A	B	C	D	E	F	
1	Геометрические фигуры						
2	1	Квадрат					
3		Сторона	Периметр	Площадь			
4		3	12	9			
5							
6	2	Прямоугольник					
7		Длина	Высота	Периметр	Площадь		
8		8	5	26	40		
9							
10	3	Куб					
11		Ребро	Площадь	Объем			
12		3	54	27			
13							
14	4	Прямоугольный треугольник					
15		Катет	Катет	Гипотенуза	Периметр	Площадь	
16		3	4	5	12	6	
17							

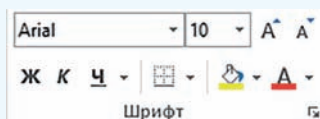
2 Создайте таблицу с вашим расписанием уроков на неделю. Подберите подходящие высоту строк и ширину столбцов, объедините нужные ячейки.

3 Создайте таблицу для подсчета своих расходов на питание в школьной столовой по дням недели. Подберите подходящие высоту строк и ширину столбцов, объедините нужные ячейки.

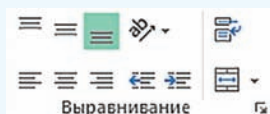
§ 13. Форматирование электронной таблицы

Пример 13.1. Команды форматирования на вкладке Главная.

Раздел **Шрифт**:



Раздел **Выравнивание**:



13.1. Форматирование ячеек

К данным в ячейках электронной таблицы можно применять форматирование. Команды, предназначенные для форматирования, расположены на вкладке Главная в разделах **Шрифт** и **Выравнивание** (пример 13.1). Эти же команды использовались при форматировании текста в Word. Здесь собраны наиболее часто используемые команды.

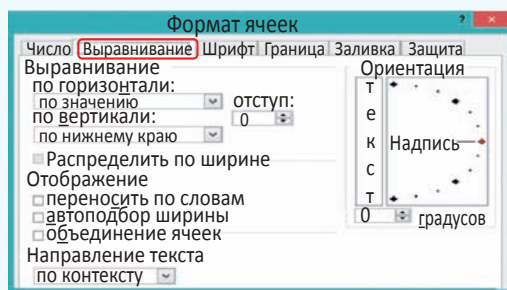
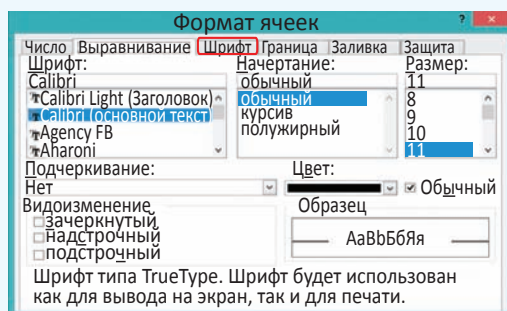
Дополнительные команды можно выбрать в окне **Формат ячеек** (пример 13.2). Открыть окно можно, выполнив соответствующую команду из контекстного меню ячейки или воспользовавшись кнопкой  в разделах **Выравнивание** или **Шрифт** вкладки **Главная**. На вкладке **Шрифт** устанавливаются необходимые параметры шрифта. На вкладке **Выравнивание** можно установить выравнивание содержимого ячейки по горизонтали и вертикали (выбрав значения из соответствующих выпадающих списков). Флажок **Переносить по словам** (соответствует кнопке ) позволит расположить текст в ячейке в несколько строк. Ориентацию содержимого ячейки можно изменить, повернув маркер **Надпись** на заданное количество градусов.

13.2. Границы и заливка

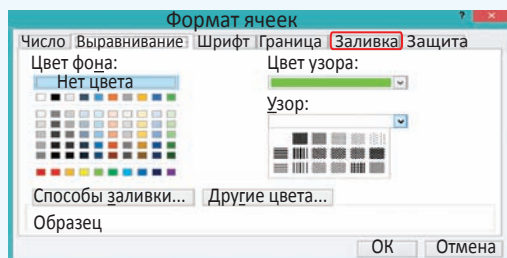
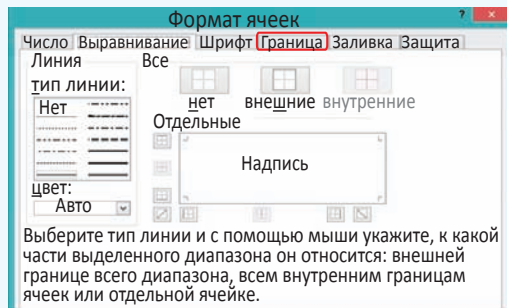
Лист электронной таблицы на экране разбит на ячейки, однако при печати границы ячеек не выводятся, если они не установлены. В этом легко убедиться, выполнив предварительный просмотр. На вкладке **Граница** окна **Формат ячеек** можно определить внешний вид линии, цвет и расположение границы ячейки (пример 13.3). На вкладке **Заливка** можно выбрать цвет и стиль заливки ячейки таблицы.

Для оформления документа в электронных таблицах Excel используются также возможности автоматического форматирования. Для этого

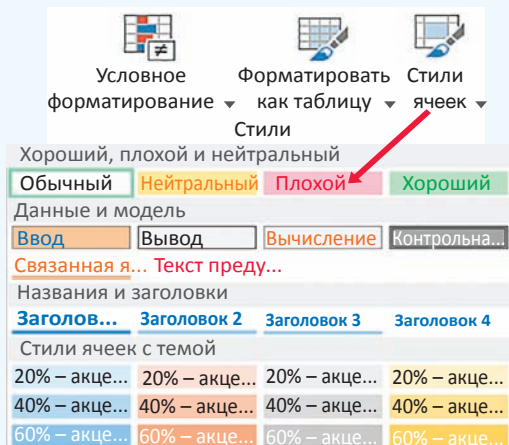
Пример 13.2. Команды форматирования окна **Формат ячеек**.



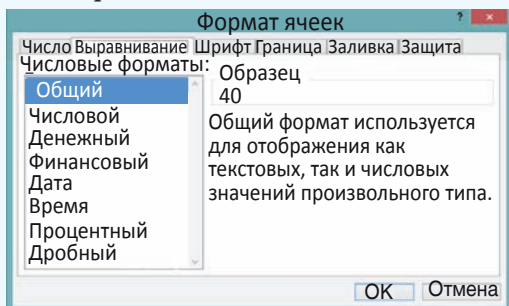
Пример 13.3. Команды установки границ и заливки.



Пример 13.4. Раздел **Стили** вкладки **Главная**.



Пример 13.5. Вкладка **Число** окна **Формат ячеек**.



Пример 13.6. Число 100,1 в разных числовых форматах.

	A	B
1	Общий	100,1
2	Числовой, 2 знака после запятой	100,10
3	Денежный	100,10Р
4	Дата	09.04.1900
5	Время	2:24:00
6	Процентный	10010%
7	Дробный	100 1/10
8	Экспоненциальный	1,00E+02
9	Текстовый	100,1

необходимо выделить диапазон ячеек таблицы, затем на ленте **Главная** в разделе **Стили** (пример 13.4) выбрать **Форматировать как таблицу** или **Стили ячеек**, из предложенного списка форматов выбрать подходящий. Команда **Форматировать как таблицу** оформляет выделенный диапазон как отдельную таблицу, чередуя оформление отдельных ячеек. Команда **Стили ячеек** применяет выбранный стиль ко всем выделенным ячейкам.

Инструмент  позволяет копировать формат оформления одной ячейки на другую.

13.3. Форматы чисел

Числовые данные в ячейке таблицы могут быть представлены в различных форматах. Уже были рассмотрены целые и действительные числа, обыкновенные и смешанные дроби. Полный список всех форматов представления числовых данных можно увидеть на вкладке **Число** окна **Формат ячеек** (пример 13.5).

Формат отображения числа позволяет показать число в ячейке таблицы так, как нужно пользователю. Однако при изменении формата отображения значение числа не изменяется, в чем можно легко убедиться, посмотрев в строку формул.

В примере 13.6 показано изменение внешнего вида числа 100,1 при изменении формата отображения числа.

Для задания форматов отображения числа можно воспользоваться

разделом **Число** на вкладке **Главная** (пример 13.7).

Формат **Общий** используется по умолчанию. Тип данных определяет- ся автоматически.

Тип данных **Числовой** позволяет установить количество цифр после запятой у действительных чисел. Если исходное число имело меньшее количество цифр, то добавляются нули, если большее — число округляется. Изменять количество цифр после запятой можно, используя кнопки:

 — для увеличения разрядности;

 — для уменьшения разрядности.

Если в числе много цифр, его можно отображать, разделяя группы разрядов с помощью кнопки .

Формат **Денежный** используется для отображения денежных величин: к числу приписывается значок денежной единицы. Установить этот формат можно с помощью кнопки , из выпадающего списка можно выбрать тип валюты (пример 13.8).

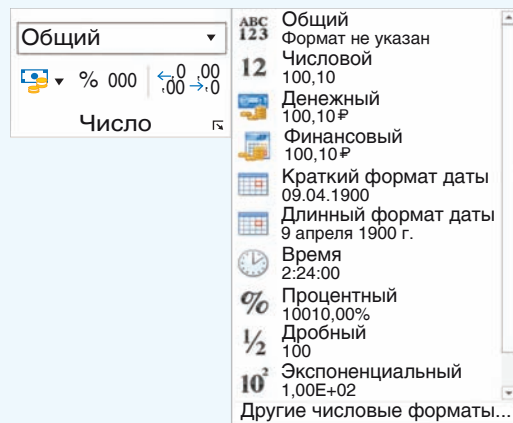
Форматы **Дата** и **Время** позволяют вводить числа в виде дат или времени.

При использовании формата **Процентный** число в ячейке умножается на 100 и к нему приписывается знак %. Его устанавливают с помощью значка .

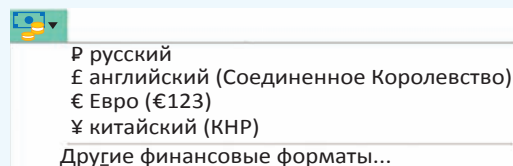
Формат **Дробный** преобразует число в дробь, а формат **Экспоненциальный** — в число в экспоненциальном представлении.

Отображение данных в разных форматах представлено в примере 13.9.

Пример 13.7. Раздел **Число** на вкладке **Главная**.



Пример 13.8. Выбор названия валюты для денежного формата.



Пример 13.9. Использование форматов отображения чисел.

	A	B	C	D	E	F	G
	№ поезда	Направление	Начало движения	Время отправления	Время прибытия	В пути	Стоимость билета
1							
2	390	Анапа	10.май	12:15	12:00	43:45	172.5 р.
3	250	С.-Петербург		18:22	9:16	14:54	91.3 р.
4	749	Полоцк	05.ноя	6:30	10:34	04:04	8.7 р.
5	122	Витебск		17:30	22:02	04:32	10.4 р.

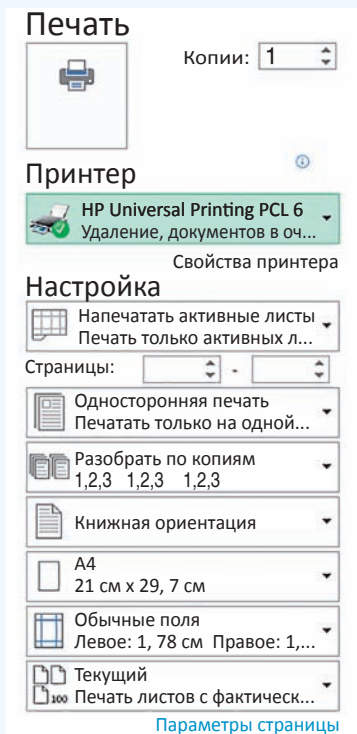
Для ячеек установлены следующие форматы:

- для данных первой строки изменена ориентация текста;
- для данных столбца A установлен **Числовой формат**, число десятичных знаков равно нулю;
- в столбце C установлен формат **Дата**;
- данные в столбцах D, E, F представлены в виде **времени**;
- для чисел столбца G установлен формат **Денежный**.

Пример 13.10. Панель быстрого доступа.



Пример 13.11. Меню Печать.



13.4. Подготовка таблицы к печати

Одним из способов вывода данных электронной таблицы на печать является использование значка **Быстрая печать** () . Если этого значка нет на панели быстрого доступа (пример 13.10), то его можно добавить с помощью кнопки  . На печать будет выведена та часть листа электронной таблицы, в которую пользователь вводил данные.

Перед тем как отправить документ на печать, рекомендуется выполнить его предварительный просмотр, используя кнопку  на панели быстрого доступа.

Меню, которое появляется в режиме предварительного просмотра документа (пример 13.11), позволяет внести необходимые изменения в формат страницы, изменить поля документа или отправить документ на печать.

Подготовка электронной таблицы к печати выполняется аналогично подготовке к печати текстового документа.

1. Какие форматы можно устанавливать для данных в ячейке электронной таблицы?
2. Как изменить форматирование текста в ячейке электронной таблицы?
3. Как установить границы ячейки?
4. Какие способы для выравнивания текста в ячейке электронной таблицы можно применять?
5. Какие форматы чисел поддерживаются в электронных таблицах?
6. Как изменить количество цифр после запятой в числе?
7. Что происходит с числом при установке формата **Процентный**?



Упражнения

- 1 Откройте таблицу, созданную в упражнении 1 после § 12. Оформите ее, применяя некоторые из шаблонов оформления для стиля ячейки.

1	Геометрические фигуры		
	1	Квадрат	
	Сторона	Периметр	Площадь
	3	12	9

2	6	2	Прямоугольник			
	7		Длина	Высота	Периметр	Площадь
	8		8	5	26	40
	9					

10	3	Куб				
11		Ребро	Площадь	Объем		
12		3	54	27		
13						

14	4	Прямоугольный треугольник				
15		Катет	Катет	Гипотенуза	Периметр	Площадь
16		3	4	5	12	6
17						

2 Откройте электронную таблицу с данными, введите формулы и примените форматирование.

Исходная таблица

	A	B	C	D
1	Счет за ремонт обуви			
2	Вид ремонта	Количество, г	Цена за п	Стоимость
3	Набойки	2	11,3	
4	Замена молнии	1	8,36	
5				
6	Итого			
7	Налог		0,02	
8	К оплате			

Готовая таблица

	A	B	C	D
1	<i>Счет за ремонт обуви</i>			
2	Вид ремонта	Количество пар обуви	Цена за пару	Стоимость
3	Набойки	2	11,30 Р	22,60 Р
4	Замена молнии	1	8,36 Р	8,36 Р
5				
6	Итого			30,96 Р
7	Налог			0,02 Р
8	К оплате			31,58 Р

В столбце **Стоимость** введите формулы для вычисления:

- в ячейке D3: =B3*C3
- в ячейке D4: =B4*C4
- в ячейке D6: =D3+D4
- в ячейке D7: =D6*C7
- в ячейке D8: =D6+D7

Для ячеек D3:D4 и D6:D8 установите формат **Денежный**. Для ячейки C7 — формат **Процентный**.

3 Создайте таблицу. Введите нужные формулы. Продумайте и примените оформление таблицы. Для числовых данных в столбцах D и F установите формат **Процентный**.

	A	B	C	D	E	F
1	Наполняемость классов					
2	Класс	Количество человек	Девочки		Мальчики	
3			Количество	Процент	Количество	Процент
4		9А	25	13		
5		9Б	31	10		
6		9В	28	16		

Добавьте в электронную таблицу строку еще об одном классе. Данные введите самостоятельно.

4 Создайте таблицу для перевода значения температуры из градусов по шкале Цельсия в градусы по шкалам Кельвина, Фаренгейта, Ранкина. Используйте для перевода следующие формулы: $T_K = T_C + 273,15$; $T_F = 1,8T_C + 32$; $T_R = T_F + 459,67$. Установите для температуры по шкалам Кельвина и Ранкина формат отображения чисел с двумя знаками после запятой, а для остальных — с одним. Оформите таблицу.

	А	В
1	Различные температурные шкалы	
2	Температура по Цельсию	20,5
3	Температура по Кельвину	
4	Температура по Фаренгейту	
5	Температура по Ранкину	

5 Создайте таблицу. Введите нужные формулы. Продумайте и примените оформление таблицы. Для числовых данных в столбцах А и В установите формат **Время**. Для данных в столбцах С и Е — формат **Денежный**.

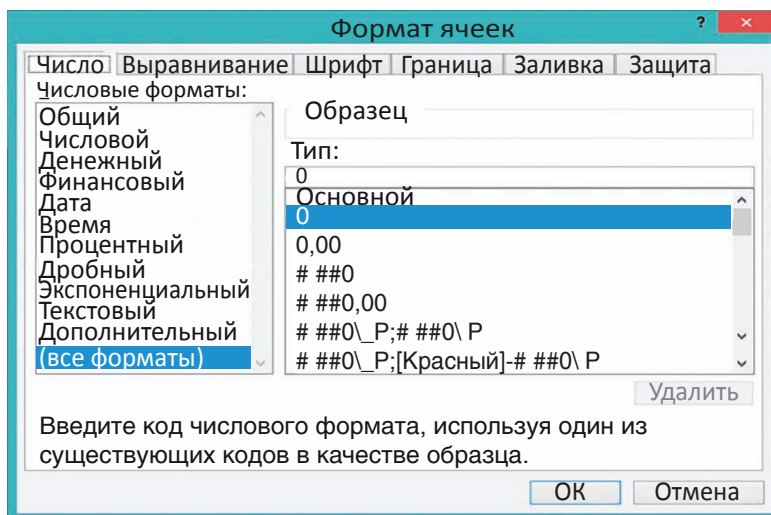
	А	В	С	Д	Е
1	Расход электроэнергии на отопление				
2	Время суток		Тариф	Расход	Стоимость
3	Начало	Конец			
4	23:00	6:00	0,10р.	65	
5	6:01	22:59	0,19р.	112	
6	Итого				

6 Создайте электронную таблицу, применив оформление и установив нужный формат отображения чисел. *Для текста в ячейке А2 установлен флажок **Переносить по словам**, горизонтальное выравнивание — по центру, вертикальное выравнивание — распределенное (можно выбрать на вкладке **Выравнивание** окна **Формат ячеек**), граница — диагональ.

	A	B	C	D	E	F
1	Погода в феврале					
	Дата Показатели	1 февраля	8 февраля	15 февраля	22 февраля	1 марта
2						
3	Температура, °С	-11	-8	-5	-4	-6
4	Осадки	-	-	снег	снег	-
5	Ветер	с	сз	з	св	с
6	Давление, мм рт. ст.	740	735	742	740	740

7* Внесите изменения в таблицу из упражнения 6.

1. Для чисел в строке **Температура** задайте формат числа со знаком °. Для этого необходимо сначала вставить в какую-либо ячейку символ ° и скопировать его в буфер обмена. Затем выполнить команду **Формат ячеек**. На вкладке **Число** выбрать **(все форматы)**. Затем выбрать формат 0. Установить курсор в поле после 0 и вставить символ ° из буфера обмена. Созданный формат добавится в список, и его можно выбирать для установки формата чисел, отображающих температуру.



2. Используя команду **Вставка символа**, замените направление ветра стрелками, а слово *снег* — соответствующим символом (Wingdings 2).

Результат выполнения работы будет следующим:

	A	B	C	D	E	F
1	Погода в феврале					
	<i>Дата</i>	1 февраля	8 февраля	15 февраля	22 февраля	1 марта
2	<i>Показатели</i>					
3	Температура, °C	-11°	-8°	-5°	-4°	-6°
4	Осадки	-	-	☂	☂	-
5	Ветер	↑	↖	←	↗	↑
6	Давление, мм рт. ст.	740	735	742	740	740

§ 14. Ссылки в формулах

Пример 14.1. Подсчет стоимости.

	A	B	C	D	E
1	Стоимость покупки				
2	№	Название продукта	Цена	Количество	Стоимость
3	1	Молоко	1,45	1	1,45
4	2	Хлеб	0,93	2	1,86
5	3	Шоколад	1,54	2	3,08
6	4	Яблоко	0,34	5	1,7
7	5	Йогурт	1,2	4	4,8

Пример 14.2. Копирование формул.

Пусть в ячейку F10 введена формула $=B5+C7$. Нужно скопировать эту формулу в ячейки E10, F8, F13, H10, I9.

	D	E	F	G	H	I
7						
8			$=B3+C5$			
9						$=E6+F8$
10		$=A5+B7$	$=B5+C7$	$=D5+E7$		
11						
12						
13			$=B8+C10$			

При копировании формулы в пределах одной строки меняется только ссылка на имя столбца (формулы в ячейках E10, H10). При копировании формулы в пределах одного столбца меняется только ссылка на номер строки (формулы в ячейках F8, F13). Для формулы в ячейке I9 изменились ссылки и на номер столбца, и на номер строки.

Принцип относительной адресации применяется и при изменении количества строк и столбцов в электронной таблице. Это означает следующее: если при удалении (вставке) строк (столбцов) изменилось местоположение исходных данных, на которые ссылается формула, то ссылки тоже будут изменены таким образом, чтобы по формуле производились прежние вычисления.

Рассмотрим пример электронной таблицы, созданной для подсчета стоимости купленных продуктов (пример 14.1).

Для вычисления стоимости каждого продукта нужно умножить содержимое ячейки в столбце **Цена** на содержимое ячейки в столбце **Количество**. Так, для подсчета стоимости молока нужно ввести формулу $=C3*D3$ в ячейку E3. Формула в ячейке E4 для подсчета стоимости хлеба будет $=C4*D4$. Формулы различаются только номерами строк, как и их местоположение в электронной таблице. По своему виду это одна формула, но записанная для различных исходных данных.

Если вычисления должны проводиться по формулам, имеющим одну структуру, но для различных исходных данных, то такие формулы можно копировать. В отличие от текстов и чисел, которые при копировании не изменяются, формулы при копировании меняются.

В электронных таблицах применяется **принцип относительной адресации**: при копировании формулы на новое место ссылки на ячейки изменяются относительно перемещения формулы.

Это означает, что при копировании формулы из одной строки в другую номера строк изменяются на столько, на сколько строк переместится формула. При копировании формулы из одного столбца в другой изменяются обозначения столбцов на столько, на сколько столбцов переместится формула (пример 14.2).

Ссылки в формулах, которые изменяются при копировании, называют **относительными**.

По умолчанию все ссылки для указания адресов ячеек относительные.

Если необходимо, чтобы при копировании ссылки не изменялись, то нужно использовать **абсолютные (фиксированные) ссылки**.

Для создания абсолютной ссылки в формуле перед именем столбца и (или) номером строки вводится знак \$.

У **смешанных ссылок** при копировании остается неизменяемой только одна часть: ссылка на номер строки (E\$4) или столбца (\$E4).

(Рассмотрите пример 14.3.)

Пусть для таблицы из примера 14.1 требуется перевести стоимость купленных продуктов из белорусских рублей в российские (пример 14.4). Для перевода стоимости молока нужно его стоимость умножить на курс перевода (из белорусских рублей в российские), т. е. в ячейку F4 ввести формулу =E4*D2. Если копировать эту формулу во все остальные ячейки в столбце **Стоимость**, то ссылки в формуле будут меняться (будем получать =E5*D3, =E6*D4 и т. д.), и результат будет ошибочным. Чтобы вычисления выполнялись верно, при копировании ссылка на ячейку D2 не должна меняться, т. е. быть абсолютной. Правильная формула имеет следующий вид: =E4*\$D\$2. При копировании формул в ячейки F5, F6... будем получать: E5*\$D\$2, E6*\$D\$2,

Пример 14.3. Примеры ссылок.

Ссылка C7 является относительной ссылкой на ячейку C7. При копировании формул, содержащих такую ссылку, ссылка будет изменяться согласно принципу относительной адресации.

Ссылка \$A\$4 будет абсолютной ссылкой на ячейку A4, и при копировании формул, содержащих такую ссылку, ссылка изменяться не будет.

Ссылка E\$4 будет изменяться при копировании формул в другие столбцы, но не будет меняться при копировании формул в другие строки.

Ссылка \$E4 будет изменяться при копировании формул в другие строки, но останется неизменной при копировании формул в другие столбцы.

Изменить тип ссылки можно клавишей F4. Сначала вводим адрес ячейки, который по умолчанию является относительной ссылкой. Затем нажимаем F4. Ссылка меняется на абсолютную. Последующие два нажатия F4 дают смешанные ссылки. Затем опять появляется относительная ссылка.

Пример 14.4. Перевод денежных единиц из одних в другие. Результат:

	A	B	C	D	E	F
1	Стоимость покупки					
2	Курс перевода			31,36		
3	№	Название продукта	Цена	Кол-во	Стоимость	Стоимость в российских рублях
4	1	Молоко	1,45	2	2,9	90,94 р.
5	2	Хлеб	0,93	2	1,86	58,33 р.
6	3	Шоколад	1,54	4	6,16	193,18 р.
7	4	Яблоко	0,34	5	1,7	53,34 р.
8	5	Йогурт	1,2	4	4,8	150,53 р.
9	Стоимость				17,42	546,29 р.

В ячейку F4 ввели формулу =E4*\$D\$2. Для копирования формулы в ячейки F5:F8 можно использовать **маркер заполнения**. Вначале делаем активной ячейку с формулой (F4), а затем при помощи маркера заполнения выделяем область для копирования формулы.



1. Что такое ссылка?
2. Сформулируйте принцип относительной адресации.
3. Какие ссылки можно использовать при создании формул?
4. В чем отличие относительных, абсолютных и смешанных ссылок?

**Упражнения**

- 1 Для электронной таблицы, изображенной в режиме показа формул, определите, в каких формулах использовали относительные, абсолютные и смешанные ссылки.

	A	B	C
1	1	2	=A1+B\$1
2	3	4	=A2+\$B2
3	=A1+\$A\$2	=B1+B2	
4			

Как изменятся формулы при копировании в указанные ячейки?

1. Из ячейки A4 в ячейку A5.
 2. Из ячейки B4 в ячейку C6.
 3. Из ячейки C1 в ячейку D2.
 4. Из ячейки C2 в ячейку C5.
- 2 Создайте электронную таблицу для вычисления стоимости канцелярских товаров, купленных к началу учебного года:

	A	B	C	D	E
1	№	Название товара	Цена	Количество	Стоимость
2	1	Тетрадь в клетку	0,49	25	
3	2	Тетрадь в линейку	0,42	50	
4	3	Тетрадь общая	2,32	5	
5	4	Ручка шариковая	1,78	4	
6	5	Линейка	1,03	2	
7	6	Простой карандаш	0,71	3	

Для подсчета стоимости тетрадей в ячейку D2 введите формулу =B2*C2. Формулу из ячейки D2 скопируйте в ячейки диапазона D3:D7. Оформите таблицу по своему усмотрению.

- 3 Для таблицы из упражнения 2 рассчитайте стоимость товаров, если вам предоставили скидку в 0,1 р. для каждого вида товара. Значение скидки внесите в ячейку B8. (Скидка вычитается из стоимости товара.)

	A	B	C	D	E	F
1	№	Название товара	Цена	Количество	Стоимость	Стоимость со скидкой
2	1	Тетрадь в клетку	0,49	25		
3	2	Тетрадь в линейку	0,42	50		
4	3	Тетрадь общая	2,32	5		
5	4	Ручка шариковая	1,78	4		
6	5	Линейка	1,03	2		
7	6	Простой карандаш	0,71	3		

4 Измените формулы для таблицы из упражнения 3 с условием того, что скидка на товар составляет 5 %, а не 0,1 р.

5 Создайте таблицу для вычисления значений функции $y = x^2 - 3x + 5$ на отрезке $[-10; 10]$, если аргумент изменяется с шагом 1.

	A	B
1	x	y
2	-10	=A2^2-3*A2+5
3	=A2+1	
4		

Скопируйте формулы из ячеек A3 и B2.

6 Создайте таблицу для вычисления значений функции $y = -x^2 + 4,1x - 3$. Начальное значение переменной x хранить в ячейке B2, шаг изменения аргумента — в ячейке C2. В ячейку B5 введите формулу, являющуюся ссылкой на начальное значение, в ячейку C5 — формулу для вычисления значения функции. В ячейке B6 должна быть формула, увеличивающая значение аргумента на значение шага (ссылка на значение шага должна быть фиксированной). Используя маркер заполнения, скопируйте формулу из ячейки B5 вниз, пока не получите 10. Формулу из ячейки C5 скопируйте вниз, пока не получите значение функции в точке $x = 10$.

	A	B	C
1		Начальное значение x	Шаг
2		-10	0,1
3			
4		x	y
5	1	=B2	
6	2		
7	3		
8	4		
9	5		

7 В таблице заданы показания счетчика учета воды за полугодие. Стоимость 1 кубического метра воды ввести в ячейку B2. Начальное показание счетчика хранится в ячейке C2. Определите стоимость воды для каждого месяца:

	A	B	C	D
1		Цена 1 куб. м воды	Начальные показания счетчика	
2		0,81	25	
3				
4	Месяц	Текущие показания счетчика	Израсходовано воды	Стоимость за месяц
5	Январь	45		
6	Февраль	68		
7	Март	82		
8	Апрель	102		
9	Май	126		
10	Июнь	145		

§ 15. Использование стандартных функций

Пример 15.1. Функции в Excel.

Запись в Excel	Значение
КОРЕНЬ(A2)	Квадратный корень из числа
SIN(A2)	Значение функции $\sin(x)$. Аргумент — радианы
ABS(A2)	Модуль числа
ПИ()	Значение константы π
СУММ(A2:A1)	Сумма чисел в диапазоне
МАКС(A2:A10)	Максимальное значение из диапазона
МИН(A2:A10)	Минимальное значение из диапазона
СРЗНАЧ(A2:A10)	Среднее значение из диапазона

15.1. Функции в Excel

В Excel имеется большое количество функций, позволяющих пользователю выполнять математические и статистические расчеты, текстовые и логические операции, а также поиск информации на рабочем листе. Результат функции зависит от ее аргументов. Аргументы функции заключаются в скобки и записываются после имени функции (пример 15.1). Аргументами функции могут быть числа, тексты, ссылки на ячейки или диапазоны ячеек. Если у функции несколько аргументов, то их перечисляют через точку с запятой.

Для вычисления значения выражения нужно записать его, исполь-

зую функции Excel (пример 15.2). При вводе нескольких символов из имени функции Excel отображает подсказку, после чего имя функции можно выбрать мышью из выпадающего списка.

Чаще всего употребляют функцию для суммирования значений ячеек. Эта функция вынесена на панель инструментов в виде кнопки **Автосумма**: . Чтобы воспользоваться данной кнопкой, нужно установить курсор в ячейку, в которой должен быть результат. Затем нажать на саму кнопку. Excel автоматически попытается определить диапазон значений для суммирования. Если выделенный диапазон не устраивает пользователя, то нужно изменить выделение, затем нажать клавишу Enter или кнопку

Рядом с кнопкой **Автосумма** находится значок выпадающего списка, с помощью которого можно ввести еще несколько часто встречающихся функций (пример 15.3). Используются эти функции аналогично функции вычисления суммы (пример 15.4).

15.2. Категории функций

Количество функций, поддерживаемых в Excel, очень велико. Запомнить их все невозможно. Для облегчения использования функции в Excel разделены на категории. Наиболее часто встречающиеся категории функций отражены на вкладке **Формулы**.



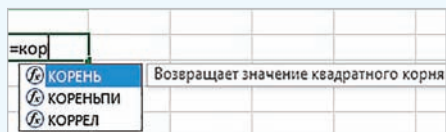
Пример 15.2. Вычисление значения выражения.

Найти значение выражения

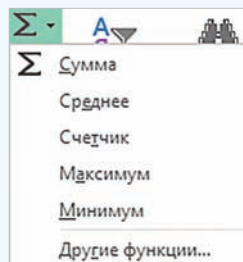
$$\sqrt{\frac{x+y}{xy}} + \sin^2(x) + \cos^2(y).$$

Пусть значение переменной x хранится в ячейке A2, а значение переменной y — в ячейке B2. Формула для вычисления выражения в Excel:

=КОРЕНЬ((A2+B2)/A2/B2) +
(SIN(A2))^2+(COS(B2))^2



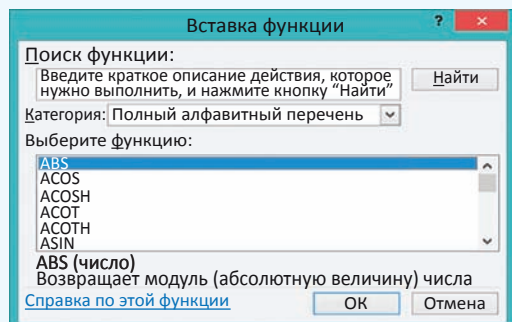
Пример 15.3. Функции из выпадающего списка **Автосумма**.



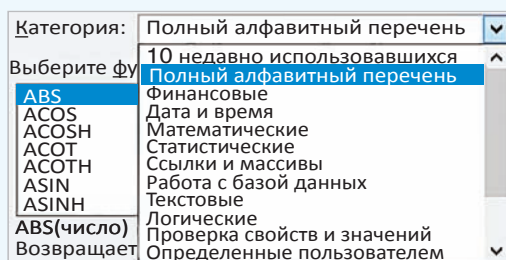
Пример 15.4. Вычисление минимального, максимального и среднего значения температуры за 7 дней января.

	А	В
1	Дата	Температура
2	1 января	-3
3	2 января	0
4	3 января	2
5	4 января	2
6	5 января	-2
7	6 января	-5
8	7 января	-7
9		
10	средняя	=СРЗНАЧ(B2:B8)
11	минимальная	=МИН(B2:B8)
12	максимальная	=МАКС(B2:B8)

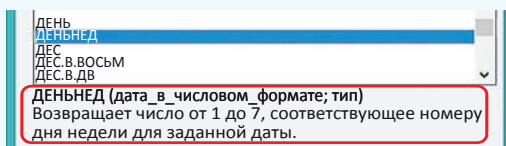
Пример 15.5. Окно Вставка функции.



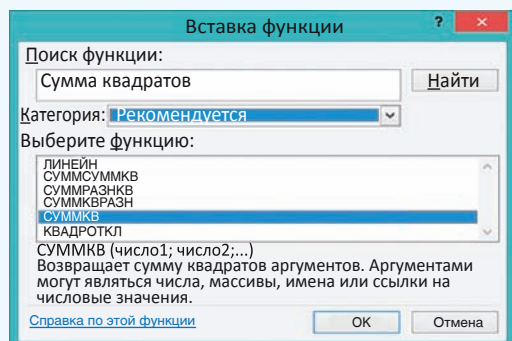
Пример 15.6. Список категорий функций.



Пример 15.7. Описание функции ДЕНЬНЕД.



Пример 15.8. Поиск функции.



Список всех категорий функций можно увидеть в окне **Вставка функции** (пример 15.5), если выполнить одно из следующих действий:

1. Нажать на кнопку  на вкладке **Формулы** или в строке формул.
2. Выбрать ссылку **Другие функции** в выпадающем списке кнопки **Автосумма**.
3. Выбрать ссылку **Вставить функцию** в выпадающем списке любой категории.

Распределение функций по категориям представлено в примере 15.6. Категория **Полный алфавитный перечень** содержит все функции, расположенные в алфавитном порядке. Список из 10 последних использованных функций изменяется при выборе пользователем новых функций.

После выбора категории в поле выбора функции появляется список функций, относящихся к данной категории. Для каждой функции внизу приводится ее название со списком возможных аргументов и краткое описание назначения функции (пример 15.7).

Если неизвестно, к какому разделу принадлежит функция, или нужно уточнить список аргументов, то можно воспользоваться поиском функций. Для этого в поле **Поиск функции** вводят краткое описание функции. После нажатия кнопки **Найти** в списке рекомендованных останутся только те функции, которые подходят под образец поиска. Например, найдем функцию, которая может вычислять сумму квадратов чисел диапазона ячеек. Для этого:

1. В поле **Поиск функции** введем описание: **сумма квадратов**.

2. Просмотрим рекомендованные функции.

3. Нужная функция: СУММКВ(число1; число2; ...) (пример 15.8).

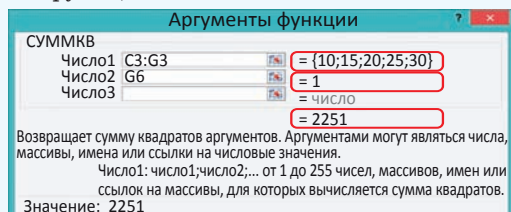
После выбора функции открывается новое окно, в котором заполняются поля для каждого из аргументов выбранной функции. Аргументы можно вводить вручную или выбирать левой клавишей мыши ячейку (диапазон ячеек), ссылка на которую(-ый) является аргументом функции (пример 15.9).

Справа от поля, в которое введены ссылки на ячейки, можно видеть значения, введенные в эти ячейки. Ниже показывается вычисленное значение функции. Если значения введены некорректно, значение функции отображаться не будет (пример 15.10).

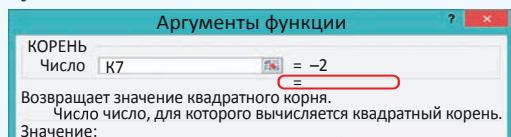
Пример 15.11*. Результаты решения задачи на олимпиаде по информатике зависят от количества пройденных тестов. Каждому участнику ставят 1, если соответствующий тест пройден, и 0 — в противном случае. Известно количество баллов за каждый тест. Вычислить результат каждого участника.

Для вычисления количества баллов Галкиной Веры, полученных за задачи, можно воспользоваться функцией СУММПРОИЗВ, вычисляющей сумму произведений соответствующих значений диапазонов. В нашем случае необходимо вычислить сумму произведений диапазона C3:G3 на C5:G5. Результат записывается в ячейку H5. Ссылки на ячейки третьей строки фиксированные, поскольку баллы за тесты одинаковы для всех участников. Затем формулу из ячейки H5 копируем для всех участников.

Пример 15.9. Возможные аргументы функции СУММКВ.



Пример 15.10. Ошибочный ввод аргументов.

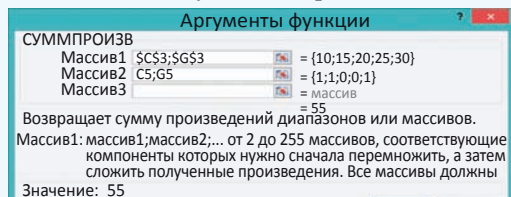


В ячейке записано отрицательное число, корень из него не вычисляется.

Пример 15.11*. Решение задачи.
Исходная таблица:

	A	B	C	D	E	F	G	H
1		Задача 1						Итог
2		Тест	1	2	3	4	5	
3		Балл	10	15	20	25	30	100
4								
5	1	Галкина Вера	1	1	0	0	1	
6	2	Иванов Алексей	1	1	1	1	1	
7	3	Кононов Федор	1	1	1	0	0	

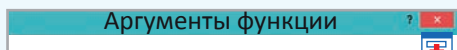
Выбираем функцию СУММПРОИЗВ. Поля в окне аргументов функции заполняются следующим образом:



Результат:

	A	B	C	D	E	F	G	H
1		Задача 1						Итог
2		Тест	1	2	3	4	5	
3		Балл	10	15	20	25	30	100
4								
5	1	Галкина Вера	1	1	0	0	1	55
6	2	Иванов Алексей	1	1	1	1	1	100
7	3	Кононов Федор	1	1	1	0	0	45

Пример 15.12. Свернутое окно **Аргументы функции**.



Чтобы вернуться в окно **Аргументы функции**, после ввода аргумента нужно нажать на кнопку .

Часто открывающееся окно **Аргументы функции** закрывает расчетную часть электронной таблицы. Возле каждого поля для ввода аргумента есть кнопка , позволяющая свернуть окно (пример 15.12).



1. Что может быть аргументом функции в Excel?
2. Какой знак служит разделителем для нескольких аргументов функции?
3. Для чего предназначена кнопка **Автосумма**?
4. Какие функции можно выбрать из выпадающего списка **Автосумма**?
5. Какие категории функций в Excel вы можете назвать?
6. В какой категории находятся все функции Excel?
7. Как открыть окно **Вставка функции**?
8. Как использовать поиск функции?
9. Как можно вводить аргументы функций?



Упражнения

- 1 По таблице **Высота некоторых известных зданий** определите высоту самого высокого здания и среднюю высоту зданий в списке.

	A	B	C	D
1	Высота некоторых известных зданий			
2		Название	Высота, м	Местонахождение
3	1	Tokyo Sky tree	634	Токио, Япония
4	2	Абрадж аль-Бейт	601	Мекка, Саудовская Аравия
5	3	Центр междунар. торговли	541	Нью-Йорк, США
6	4	Бурдж-Халифа	828	Дубай, ОАЭ
7	5	Останкинская башня	540	Москва, Россия
8	6	Падающая башня	56	Пиза, Италия
9	7	Слонимская телевышка	374	д. Новая Стража, Беларусь
10	8	Тайбэй	501	Тайбэй, Тайвань
11	9	Шанхайская башня	632	Шанхай, Китай
12	10	Эйфелева башня	300	Париж, Франция

2 Результаты соревнований по бегу и прыжкам представлены в виде таблицы. Определите лучшие результаты в беге и прыжках (лучший результат в беге — минимальное время в секундах, а в прыжках — максимальная длина в метрах).

	A	B	C	D
1	Соревнования легкоатлетов			
2	№	Фамилия	Результат бега на 60 м, с	Результат прыжков в длину, м
3	1	Власов С.	9,9	2,1
4	2	Комаров Ю.	8,2	1,83
5	3	Усов С.	8,5	1,81
6	4	Корнев Н.	10,8	1,8
7	5	Жук Р.	8,3	1,78
8	6	Ткаченко Т.	8,6	1,76
9	7	Бажков С.	8,9	1,74
10	8	Павлов А.	10,3	2,18
11	9	Сергеев А.	10,2	2,03
12	10	Лемешко К.	9,1	2,12

3 Создайте таблицу для перевода градусов в радианы. Функция для перевода — РАДИАНЫ из категории **Математические**.

	A	B
1	Градусы	Радианы
2	0	=РАДИАНЫ(A2)
3	30	
4	60	
5	90	
6	120	
7	150	
8	180	

- 4 Вычислите значения тригонометрических функций для значений из таблицы в упражнении 3. Добавьте в таблицу столбцы $\sin(x)$, $\cos(x)$, $\operatorname{tg}(x)$, $\operatorname{ctg}(x)$. Аргументы тригонометрических функций выражаются в радианах. Для вычисленных значений установите 3 цифры после запятой.

D2		:			f_x	=COS(B2)
	A	B	C	D	E	F
1	Градусы	Радианы	sin(x)	cos(x)	tg(x)	ctg(x)
2	0	0,000	0,000	1,000		
3	30	0,524	0,500			
4	60	1,047				
5	90					
6	120					

- 5 Используя функции ДЕНЬНЕД, ДНИ, ДНЕЙ360 и СЕГОДНЯ из категории **Дата и время**, определите, в какой день недели вы родились и сколько дней прошло со дня вашего рождения. (Найдите в описании отличие функций ДНИ и ДНЕЙ360.)

B3	:			f_x	=СЕГОДНЯ()
	A	B	C		
1		Дата	День недели		
2	Дата рождения	23.03.2003	7		
3	Текущая дата	18.12.2018	2		
4					
5	Прошло дней	5749			
6	Прошло дней	5665			

- 6* Соревнования по спортивной гимнастике оценивают 10 судей. Затем минимальная и максимальная оценки отбрасываются, а из остальных находят среднее значение. Вычислите оценки всех спортсменок. Определите результат победительницы.

	A	B	C	D	E	F	G	H	I	J	K
1	Соревнования гимнасток										
2	Фамилия спортсменки	Судья 1	Судья 2	Судья 3	Судья 4	Судья 5	Судья 6	Судья 7	Судья 8	Судья 9	Судья 10
3	Иванова П.С.	1	4	8	5	9	2	4	8	1	5
4	Петрова К.И.	5	3	4	3	10	8	7	2	4	4
5	Бажкова С.А.	4	3	4	6	8	1	1	4	8	1
6	Волгина Н.П.	6	1	4	9	6	2	7	4	4	9
7	Елкина В.В.	8	6	7	3	6	9	2	8	8	6
8	Комарова А.И.	8	8	10	6	4	6	4	8	1	10

*Определите самого строгого и самого лояльного судью, вычислив предварительно средние оценки судей.

7* Создайте таблицу для решения квадратного уравнения.

	A	B	C
1	Решение квадратного уравнения		
2	a	b	c
3			
4			
5	d		
6			
7	x1	x2	
8			

§ 16. Сортировка и фильтрация данных

16.1. Сортировка данных

Электронные таблицы позволяют производить **сортировку** и **фильтрацию** данных. Для этого данные организуют специальным образом:

- в одном столбце размещают данные только одного типа (текст, число, дата и т. д.);
- столбцы имеют заголовки;
- каждая строка представляет собой запись о каком-либо одном объекте.

Сортировка — упорядочение данных по возрастанию или по убыванию их значений.

Сортировка позволяет переставить строки в таблице так, чтобы в определенном столбце они располагались по порядку. Числовые данные могут располагаться в порядке возрастания или убывания, а текстовые — в алфавитном порядке или в порядке, обратном алфавитному (пример 16.1). По возрастанию или убыванию можно располагать данные формата дат и времени.

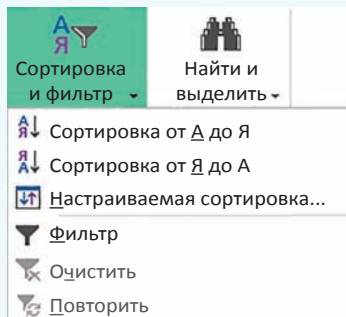
Пример 16.1. Сортировка данных. Сортировка фамилий по алфавиту:

	A	B	C	D
1	Соревнования легкоатлетов			
2	№	Фамилия	Результат бега на 60 м, с	Результат прыжков в длину, м
3	1	Бажков С.	8,9	1,74
4	2	Власов С.	9,9	2,1
5	3	Жук Р.	8,3	1,78
6	4	Комаров Ю.	8,2	1,83
7	5	Корнев Н.	10,8	1,8
8	6	Лемешко К.	9,1	2,12
9	7	Павлов А.	10,3	2,18
10	8	Сергеев А.	10,2	2,03
11	9	Ткаченко Т.	8,6	1,76
12	10	Усов С.	8,5	1,81

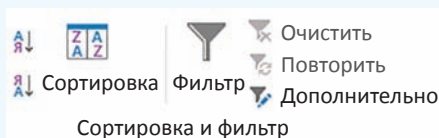
Сортировка по убыванию результатов прыжков в длину:

	A	B	C	D
1	Соревнования легкоатлетов			
2	№	Фамилия	Результат бега на 60 м, с	Результат прыжков в длину, м
3	1	Павлов А.	10,3	2,18
4	2	Лемешко К.	9,1	2,12
5	3	Власов С.	9,9	2,1
6	4	Сергеев А.	10,2	2,03
7	5	Комаров Ю.	8,2	1,83
8	6	Усов С.	8,5	1,81
9	7	Корнев Н.	10,8	1,8
10	8	Жук Р.	8,3	1,78
11	9	Ткаченко Т.	8,6	1,76
12	10	Бажков С.	8,9	1,74

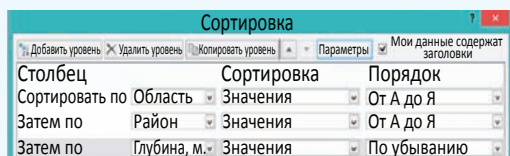
Пример 16.2. Инструменты сортировки на вкладке **Главная**:



Инструменты сортировки на вкладке **Данные**:



Пример 16.3. Окно **Сортировка**:



Первый уровень сортировки располагает данные в алфавитном порядке по названию области. Данные, у которых области одинаковые, располагаются в алфавитном порядке по названию районов. Параметр **Глубина, м** определяет порядок расположения данных, у которых одинаковые и области, и районы.

Пример 16.4. Таблица, у которой установлен **Фильтр**.

	А	В	С
1	Название	Столица	Площадь (тыс. кв.км)
2	Австрия	Вена	83,9
3	Беларусь	Минск	207,6
4	Болгария	София	110,9
5	Великобритания	Лондон	244
6	Германия	Берлин	357
7	Греция	Афины	132

Инструменты для выполнения сортировки располагаются на вкладках **Главная** и **Данные** (пример 16.2). Они дублируют друг друга.

Для сортировки нужно установить курсор в одну из ячеек столбца и выбрать (**Сортировка по возрастанию**) или (**Сортировка по убыванию**). Строки таблицы будут перепорядочены согласно выбранному критерию сортировки.

Параметры сортировки выделенного диапазона задают с помощью пиктограммы — настраиваемая сортировка. В открывшемся окне (пример 16.3) можно задавать уровни сортировки и для каждого уровня определять порядок.

Уровни сортировки необходимы, когда в каком-либо столбце имеются одинаковые данные. Тогда новый уровень позволит распределить эти данные по другому критерию.

16.2. Фильтрация данных

Фильтр — средство отображения только тех строк таблицы, которые соответствуют заданным условиям.

В отфильтрованном списке отображаются только строки, отвечающие условиям отбора данных, заданным для столбца.

Для установки **Фильтра** нужно выделить заголовки столбцов и выбрать команду **Фильтр** на вкладке **Главная** или **Данные** (команда располагается вместе с командой **Сортировка**). Спра-

ва от названий столбцов в фильтруемом списке появляются кнопки со стрелками ▼ (пример 16.4).

В отличие от сортировки фильтр не меняет порядок записей в списке. При **фильтрации** (применении фильтра) временно скрываются строки, которые не требуется отображать. Строки, отобранные при фильтрации в Excel, можно редактировать, форматировать, выводить на печать, не изменяя порядок строк и не перемещая их.

Для установки параметров фильтра нужно нажать на кнопку ▼ рядом с тем столбцом, по которому фильтруются значения. В выпадающем списке выбрать нужные значения или задать условие (пример 16.5). Для числовых и текстовых данных применяются разные условия фильтрации (пример 16.6).

Рассмотрим подробнее фильтрацию данных по различным критериям.

Фильтр по конкретному значению

Пример 16.7. В таблице указан предмет, по которому учащиеся 9-го класса посещают факультатив. Выбрать тех учащихся, которые посещают факультатив по информатике.

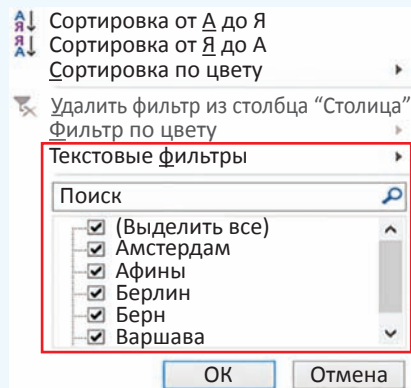
1. В столбце **Факультатив** нажать на кнопку ▼.

2. В открывшемся списке снять все выделения — снять флажок со строки **(Выделить все)**.

3. Выбрать **информатика**.

После применения фильтра цвет номера строки у отобранных данных становится синим. Для столбца, к

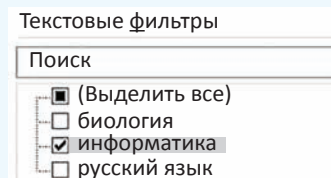
Пример 16.5. Параметры фильтра для столбца **Столица**.



Пример 16.6. Условия отбора числовых и текстовых фильтров:

Числовые фильтры	Текстовые фильтры
равно...	равно...
не равно...	не равно...
больше...	начинается с...
больше или равно...	заканчивается на...
меньше...	содержит...
меньше или равно...	не содержит...
между...	
Первые 10...	
Выше среднего...	
Ниже среднего...	
Настраиваемый фильтр...	Настраиваемый фильтр...

Пример 16.7. Параметры отбора:



Результат:

	А	В	С
1	№	Фамилия, имя	Факультатив
2	1	Бахтин Егор	информатика
7	6	Гапоненко Алексей	информатика
11	10	Костюченко Илья	информатика
12	11	Кочуновская Вера	информатика

Пример 16.8. Параметры отбора:

Результат:

	А	В	С
	Название	Столица	Площадь (тыс. кв. км)
8	Дания	Копенгаген	42,8
11	Литва	Вильнюс	65,2
12	Молдова	Кишинев	33,8
13	Нидерланды	Амстердам	41,5
17	Швейцария	Берн	41,3

Пример 16.9. Окно Пользовательский автофильтр.

Пример 16.10. Выпадающий список с условиями отбора.

Пример 16.11. Выпадающий список значений из столбца, для которого устанавливается фильтр.

которому применили фильтр, значок фильтра изменяется с ▼ на ▼.

Фильтр «Первые 10...»

Условие отбора «Первые 10...» позволяет выбрать из общего списка заданное количество строк, содержащих наибольшие или наименьшие числовые значения.

Пример 16.8. В таблице «Страны Европы» выбрать 5 стран с наименьшей площадью.

1. В столбце **Площадь (тыс. кв. км)** нажать на кнопку ▼.
2. В открывшемся списке выбрать **Числовые фильтры**.
3. Выбрать **первые 10**.
4. Установить параметры в окне.

Наложение условия по списку

Более сложные критерии отбора устанавливаются в окне **Пользовательский автофильтр** (пример 16.9). Оно открывается при выборе текстовых фильтров **начинается с...**, **заканчивается на...**, **содержит...**, **не содержит...**, а также большинства числовых фильтров, кроме **первые 10** и **выше (ниже) среднего**.

Поля под номерами 1 и 2 предназначены для задания условий отбора из выпадающего списка, показанного в примере 16.10.

Поля под номером 4 и 5 позволяют задавать значение, с которым производится сравнение. Значение выбирается из выпадающего списка, содержащего все возможные значения выбранного поля, или вводится с клавиатуры (пример 16.11).

Составные условия отбора образуются с использованием логических операций «И» и «ИЛИ». Выбор операции производится с помощью соответствующих переключателей (номер 3 в примере 16.9).

Составные условия фильтрации

Пример 16.12. В таблице «Страны Европы» выбрать страны, площадь которых больше 100 и меньше 350 тыс. кв. км.

1. В столбце **Площадь (тыс. кв. км)** нажать на кнопку .
2. В открывшемся списке выбрать **Числовые фильтры**.
3. Выбрать **между...**
4. Установить параметры в окне.

Пользовательский автофильтр

Пример 16.13. В таблице «Страны Европы» выбрать страны, названия которых заканчиваются на «ия» или на «ь».

1. В столбце **Название** нажать на кнопку .
2. В открывшемся списке выбрать **Текстовые фильтры**.
3. Выбрать **заканчивается на...**
4. Установить параметры в окне.

Восстановление данных

Для восстановления данных после фильтрации существует несколько способов. Для сброса фильтра в одном столбце достаточно открыть параметры фильтра и выбрать строку (**Выделить все**). Чтобы отменить все фильтры, нужно выполнить

Пример 16.12. Параметры отбора:

Пользовательский автофильтр ? 

Показать только те строки, значения которых:
Площадь (тыс. кв. км)

больше или равно

☒ И ☐ ИЛИ

меньше или равно

Результат:

	A	B	C
1	Название	Столица	Площадь (тыс. кв.км)
3	Беларусь	Минск	207,6
4	Болгария	София	110,9
5	Великобритания	Лондон	244
7	Греция	Афины	132
10	Италия	Рим	301,3
15	Польша	Варшава	312,7

Пример 16.13. Параметры отбора:

Пользовательский автофильтр ? 

Показать только те строки, значения которых:
Название

заканчивается на

☒ И ☐ ИЛИ

заканчивается на

Результат:

	A	B	C
1	Название	Столица	Площадь (тыс. кв.км)
2	Австрия	Вена	83,9
3	Беларусь	Минск	207,6
4	Болгария	София	110,9
5	Великобритания	Лондон	244
6	Германия	Берлин	357
7	Греция	Афины	132
8	Дания	Копенгаген	42,8
9	Испания	Мадрид	498,5
10	Италия	Рим	301,3
12	Норвегия	Осло	385,2
14	Франция	Париж	547
16	Швейцария	Берн	41,3
17	Швеция	Стокгольм	450

Пример 16.13. Продолжение.

Установленные фильтры сохраняются вместе с таблицей. Если навести указатель мыши на значок , то можно увидеть, какой фильтр установлен для столбца.

	A	B	C	D
1	Название	Столица	Площадь (тыс. кв.к)	
2	Авс	Название:		
3	Бел	Заканчивается на "ия" или Заканчивается на "ь"		

Параметры фильтра можно просмотреть, если выбрать команду **Настраиваемый фильтр**:

содержит...
не содержит...
☒ Настраиваемый фильтр...

Пример 16.14. Установка параметров отбора.

По столбцу **Название**:

Пользовательский автофильтр ?

Показать только те строки, значения которых:

Название

заканчивается на

☒ И ☐ ИЛИ

По столбцу **Площадь**:

Первые 10...
☒ Выше среднего

Результат:

	A	B	C
1	Название	Столица	Площадь (тыс. кв.к)
5	Великобритания	Лондон	244
6	Германия	Берлин	357
9	Испания	Мадрид	498,5
10	Италия	Рим	301,3
14	Норвегия	Осло	385,2
16	Франция	Париж	547
18	Швеция	Стокгольм	450

команду  **Очистить**. Ее можно выбрать в выпадающем списке команды **Сортировка и фильтр** на вкладке **Главная** или в разделе **Сортировка и фильтр** на вкладке **Данные**.

Фильтрация по нескольким столбцам

Фильтры в таблице можно устанавливать для нескольких столбцов. Применяется фильтрация последовательно, в том порядке, в котором устанавливаются фильтры.

Пример 16.14. В таблице «Страны Европы» выбрать страны, названия которых заканчиваются на «ия» и площадь которых больше средней.

1. В столбце **Название** нажать на кнопку .

2. В открывшемся списке выбрать **Текстовые фильтры**.

3. Выбрать **заканчивается на...**

4. Установить параметры в окне **Пользовательский автофильтр**.

5. В столбце **Площадь** нажать на кнопку .

6. В открывшемся списке выбрать **Числовые фильтры**.

7. Выбрать **выше среднего**.

8. Установить параметры в окне **Пользовательский автофильтр**.



1. Что такое сортировка?
2. В каком порядке Excel позволяет сортировать числовые данные? Текстовые?
3. В каких случаях задают уровни сортировки?
4. Что такое фильтр?
5. Как установить фильтр?
6. Какие текстовые фильтры можно применять к данным?
7. Какие числовые фильтры можно применять к данным?



Упражнения

1 Откройте электронную таблицу «Озера.xlsx». Скопируйте исходную таблицу 6 раз на разные листы книги и для каждой копии выполните одну сортировку.

	A	B	C	D	E	F	G
1	Озера Беларуси						
2	№	Озеро	Область	Район	Площадь, км ²	Глубина, м	Объем, млн м ³
3	1	Нарочь	Минская	Мядельский	79,6	24,8	710
4	2	Освейское	Витебская	Верхнедвинский	52,8	7,5	104
5	3	Червоное	Гомельская	Житковичский	40,3	2,9	27,35
6	4	Лумомльское	Витебская	Чашникский	37,71	11,5	249
7	5	Дривяты	Витебская	Браславский	36,1	12	222,5
8	6	Выпоновское	Брестская	Ивацевичский	26	2,3	32,1

1. Отсортируйте таблицу по областям в порядке возрастания значений.
2. Отсортируйте таблицу по названиям озер в порядке убывания значений.
3. Отсортируйте таблицу по районам в порядке возрастания значений.
4. Отсортируйте таблицу по площади в порядке убывания значений.
5. Отсортируйте таблицу по глубине озер в порядке возрастания значений.
6. Выполните многоуровневую сортировку. Отсортируйте исходную таблицу сначала по областям, затем по районам, затем по озерам в порядке возрастания значений.

2 Откройте электронную таблицу «Страны.xlsx». Скопируйте таблицу 7 раз на разные листы. Найдите с помощью фильтра записи, удовлетворяющие перечисленным условиям.

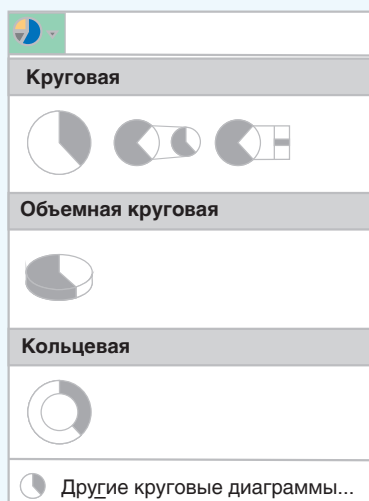
1. Страна, в которой протекает река длиной 600 км.
2. Страны, названия которых начинаются на букву «И».
3. Страны, названия столиц которых заканчиваются на букву «м».
4. Страны, самая высокая точка которых >1000 и <300 м.
5. Страны, население которых <10 или >50 млн чел.
6. Найдите первые 5 самых высоких точек Европы.
- 7*. Страны с количеством населения >7 млн чел., названия которых заканчиваются на «ия», площадь которых меньше средней.

§ 17. Построение диаграмм

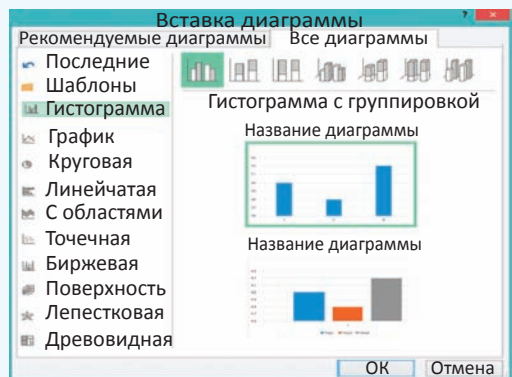
Пример 17.1. Диаграммы на вкладке **Вставка**.



Пример 17.2. Виды круговых диаграмм.



Пример 17.3. Окно **Вставка диаграммы**.



17.1. Вставка диаграммы

Для наглядного представления числовых данных электронные таблицы Excel позволяют строить **диаграммы**.

Диаграмма — условное графическое отображение числовых данных и их соотношений, выполненное геометрическими средствами.

Электронные таблицы Excel позволяют строить диаграммы различных типов. Наиболее распространенными из диаграмм являются: гистограмма (■), круговая диаграмма (●), точечная диаграмма (■). Эти и другие типы диаграмм представлены на вкладке **Вставка** в разделе **Диаграммы** (пример 17.1).

Каждый тип диаграммы, в свою очередь, содержит несколько видов диаграмм, которые можно выбрать, используя значок выпадающего списка возле пиктограммы типа диаграммы. Виды круговых диаграмм показаны в примере 17.2. Поэтому пользователь всегда может найти тот способ графического представления данных, который наиболее ярко представит его числовую информацию. Все типы диаграмм можно увидеть, если выбрать значок  на панели **Диаграммы** (пример 17.3).

Диаграмма в Excel состоит из различных объектов. Рассмотрим структуру диаграммы, построенной по таблице о количестве учащихся в

классах (пример 17.4). Номерам на диаграмме соответствуют следующие элементы:

1. Заголовок диаграммы.
2. Вертикальная ось (ось значений).
3. Область построения диаграммы.
4. Подписи осей.
5. Ряды данных.
6. Горизонтальная ось (ось категорий).
7. Легенда.
8. Линии сетки в области построения.

Чтобы построить диаграмму, в таблице необходимо выбрать данные. Затем следует определить тип диаграммы. После того как диаграмма построена, появляются дополнительные вкладки **Работа с диаграммами** → **Конструктор** и **Работа с диаграммами** → **Формат**, которые содержат набор средств для редактирования диаграмм. Вкладки можно увидеть в *Приложении 3* (с. 164).

Рассмотрим порядок построения диаграммы из примера 17.4.

1. Выделим в таблице диапазон ячеек A1:C4.

2. В разделе **Рекомендуемые диаграммы** выберем тип **Гистограмма с группировкой**.

3. Добавим название диаграммы и подписи осей.

При построении диаграммы Excel автоматически подбирает наилучший способ размещения выбранных данных на диаграмме. Однако при необходимости его можно изменить. Для изменения представления исходных данных на диаграмме нужно воспользоваться

Пример 17.4. Структура диаграммы.

Таблица:

	A	B	C
1	Класс	Количество учащихся	Девочки
2	9А	30	15
3	9Б	28	18
4	9В	32	7

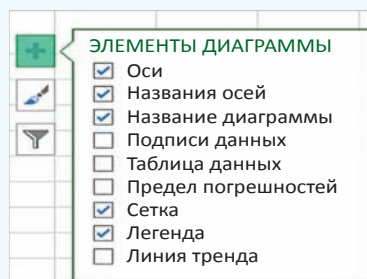
Диаграмма:



Выбор типа диаграммы:



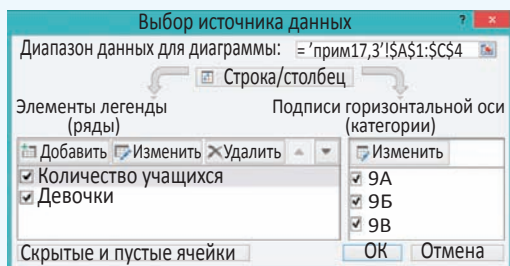
Добавление элементов диаграммы производят с помощью значка , который появляется при выделении диаграммы справа от нее:



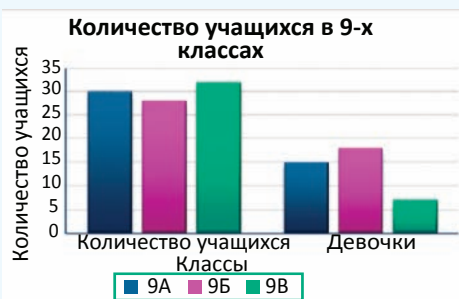
Пример 17.5. Раздел **Данные** вкладки **Работа с диаграммами** → **Конструктор**.



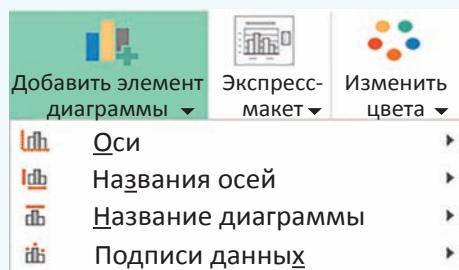
Окно **Выбор источника данных**:



Вид диаграммы после нажатия кнопки **Строка/столбец**:



Пример 17.6. Добавление элемента диаграммы с использованием соответствующей команды на вкладке **Работа с диаграммами** → **Конструктор**.



командой **Выбрать данные**, расположенной в разделе **Данные** вкладки **Работа с диаграммами** → **Конструктор** (пример 17.5). В открывшемся окне **Выбор источника данных** производят необходимые настройки. Кнопка **Строка/столбец** позволяет поменять отображение данных, расположенных по строкам/столбцам.

При изменении данных в таблице диаграмма автоматически перестраивается.

17.2. Настройка элементов диаграммы

После того как диаграмма построена, можно изменить (удалить, добавить) отдельные элементы диаграммы. Excel предоставляет несколько способов управления элементами диаграммы:

- Команды вкладки **Работа с диаграммами** (пример 17.6).

- Кнопки  и , которые появляются справа от диаграммы при ее выделении.

- Команда **Формат области построения...** из контекстного меню элемента диаграммы (пример 17.7).

Для элементов диаграммы можно настраивать не только основные, но и **дополнительные параметры**. Команда выбора дополнительных параметров открывает справа от таблицы панель, на которой можно установить необходимые параметры. Эта же панель открывается по команде **Формат области построения...** из контекстного меню элемента диаграммы. В примере 17.8 показано, как выбрать дополнительные параметры оси.

Все команды панели **Формат** объединены в группы. Выбор группы осуществляется с помощью значков в верхней части.

Значок группы **Граница и заливка**  позволяет настроить параметры заливки (цвет, узор градиент) и линии (цвет, толщина, тип штриха, наличие стрелки).

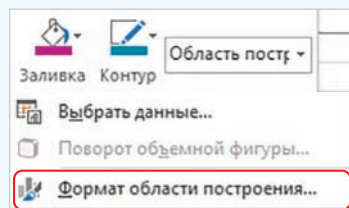
Значок группы **Эффекты**  отвечает за эффекты, которые можно применить к выделенному элементу диаграммы. Эффекты помогут настроить тень, объем, свечение и сглаживание объекта.

Значок группы **Размер и свойства**  отвечает за размер и положение элемента.

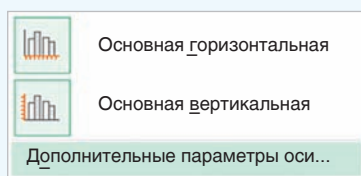
Значок группы **Параметры**  содержит список параметров выделенного объекта, значения которых можно изменить.

Диаграмма может размещаться на одном листе с таблицей или на отдельном листе. По умолчанию она размещается на том же листе, на котором находится таблица с данными. В этом случае диаграмма вставляется как объект, который можно перемещать по листу и (или) менять его размеры. Для размещения диаграммы на отдельном листе используется пиктограмма **Переместить диаграмму**, расположенная на вкладке **Работа с диаграммами** → **Конструктор** (пример 17.9). В этом случае диаграмма занимает целый лист. Имя этому листу можно задать в открывшемся окне **Перемещение диаграммы**.

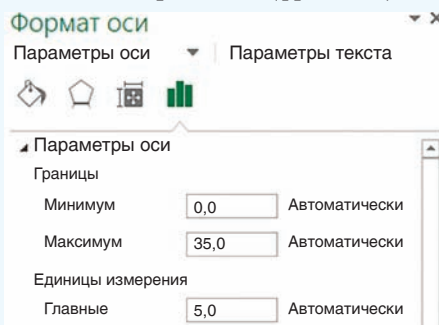
Пример 17.7. Контекстное меню области построения.



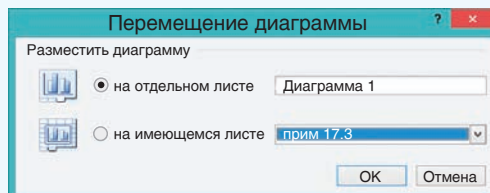
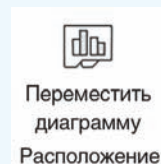
Пример 17.8. Выбор дополнительных параметров.



Панель **Формат оси** (фрагмент):



Пример 17.9. Выбор способа размещения диаграммы.



Пример 17.10. Построение круговой диаграммы.

Таблица с данными:

	А	В
1	Дипломы республиканской олимпиады по информатике 2018 г.	
2	Регион	Количество дипломов
3	Брестская обл.	7
4	Витебская обл.	9
5	г. Минск	3
6	Гомельская обл.	11
7	Гродненская обл.	4
8	Лицей БГУ	12
9	Минская обл.	2
10	Могилевская обл.	7
11	Итого	55

Диаграмма:



Пример 17.10. Построить круговую диаграмму по данным таблицы «Дипломы республиканской олимпиады по информатике».

Для построения диаграммы нужно выполнить следующие действия.

1. Выделить данные в диапазоне В3:В10 таблицы и выбрать тип диаграммы — **Круговая**.

2. Ввести название диаграммы «Дипломы республиканской олимпиады по информатике 2018 г.». Название можно скопировать из ячейки А1.

3. Легенду разместить справа.

4. Подписи данных (у края внутри).

5. Для изменения цветового оформления секторов диаграммы, фона, заголовка нужно выделять отдельно каждый объект и, выбрав из контекстного меню пункт **Формат...**, изменить форматирование. Для секторов установлена заливка градиентом или текстурой. При желании для заливки можно использовать любые растровые изображения.



1. Для чего используют диаграммы?
2. Назовите наиболее распространенные типы диаграмм.
3. Как построить диаграмму по данным таблицы?
4. Из каких шагов состоит построение диаграммы?
5. Как изменить отдельный элемент диаграммы?



Упражнения

1

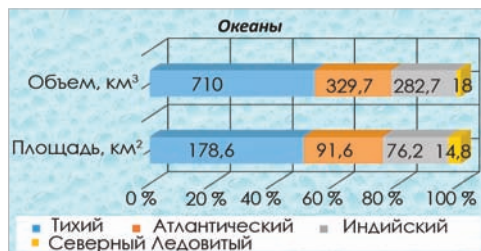
По следующим данным постройте гистограмму. Оформите отдельные элементы диаграммы.

	А	В
1	Высота некоторых вулканов	
2	Название	Высота
3	Этна	3340
4	Сангай	5232
5	Риус	5400
6	Эльбрус	5642
7	Орисаба	5747
8	Тупунгато	6880



- 2 По данным таблицы постройте линейчатую диаграмму.

	A	B	C
1	Океаны		
2		Площадь, км ²	Объем, км ³
3	Тихий	178,6	710
4	Атлантический	91,6	329,7
5	Индийский	76,2	282,7
6	Северный Ледовитый	14,8	18



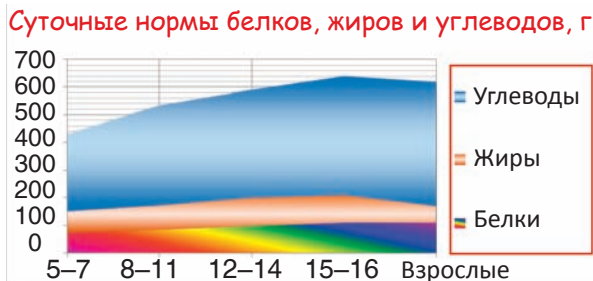
- 3 Постройте графики изменения продаж офисной мебели по данным таблицы.

	A	B	C	D	E
1	Продажа офисной мебели по кварталам				
2		1-й кв.	2-й кв.	3-й кв.	4-й кв.
3	Стол письменный	114	120	131	117
4	Стол компьютерный	142	137	152	140
5	Шкаф плательный	98	80	100	93
6	Шкаф для бумаг	123	131	132	102



- 4 Постройте по данным следующей таблицы диаграмму **С областями**. Оформите диаграмму.

	A	B	C	D
1	Суточные нормы белков, жиров и углеводов, г			
2	Возраст	Белки	Жиры	Углеводы
3	5–7	73	77	280
4	8–11	85	87	360
5	12–14	100	100	390
6	15–16	110	100	430
7	Взрослые	110	58	450



- 5 Постройте по таблице лепестковую диаграмму, отражающую данные о силе ветра.

	A	B	C
1	Ветры в марте (количество дней)		
2	Направление ветра	2018	2019
3	Северный	2	1
4	Северо-восточный	3	4
5	Восточный	5	6
6	Юго-восточный	6	5
7	Южный	7	6
8	Юго-западный	2	3
9	Западный	4	5
10	Северо-западный	2	1



§ 18. Использование электронных таблиц для выполнения практических заданий из различных предметных областей

Пример 18.1. Создание таблицы измерений роста.

1. Рост учащихся 9 "А" класса

	Сантиметров в:	дюйме	2,54
		аршине	71,12
		вершке	4,45
		футе	30,48

2. № Фамилия, имя см

1	Бахтин Егор	160
2	Белизняк Яна	157
3	Бортич Карина	156
4	Гапонов Алексей	171

3. № Фамилия, имя см дюймы аршины вершки футы

1	Бахтин Егор	160	62,99	2,25	35,96	5,25
2	Белизняк Яна	157	61,81	2,21	35,28	5,15
3	Бортич Карина	156	61,42	2,19	35,06	5,12
4	Гапонов Алексей	171	67,32	2,40	38,43	5,61

Рост учащихся 9 "А" класса

Сантиметров в:	дюйме	2,54
	аршине	71,12
	вершке	4,45
	футе	30,48

№	Фамилия, имя	см	дюймы	аршины	вершки	футы
1	Бахтин Егор	160	62,99	2,25	35,96	5,25
2	Белизняк Яна	157	61,81	2,21	35,28	5,15

Средний рост	169	66,54	2,38	37,98	5,54
Макс. рост	179	70,47	2,52	40,22	5,87
Мин. рост	156,00	61,42	2,19	35,06	5,12

Пример 18.2. Создание таблицы «Растворы».

1.	А	В	С
1	Доля вещества		Растворы
2			0,26
3	Количество вещества (моль)		2

18.1. Решение расчетных задач

Табличные вычисления востребованы в различных сферах повседневной жизни. Инструментарий электронных таблиц позволяет решать многие задачи в области технических и естественных наук, обрабатывать большие массивы данных. Рассмотрим некоторые из таких задач.

Пример 18.1. Создайте таблицу для перевода измерений роста учащихся из сантиметров в дюймы, аршины, вершки, футы. Определите для всех видов измерений: средний рост учащихся в классе, максимальный и минимальный рост.

1. Создайте таблицу. В 7 ячейках первой строки разместите заголовок. В строках 3—6 введите данные о мерах длины.

2. Введите данные о росте учащихся вашего класса в сантиметрах.

3. Добавьте в таблицу столбцы с формулами для перевода сантиметров в дюймы, аршины, вершки и футы. Используя соответствующие функции, подсчитайте средний рост учащихся в классе, их максимальный и минимальный рост.

4. Примените форматирование к ячейкам таблицы и данным в них. Используйте цветовые стили.

Пример 18.2. Создайте таблицу для расчета необходимого количества составляющих для приготовления раз-

личных растворов определенной концентрации.

1. Создайте таблицу. В строки 1—3 введите заголовок и данные о концентрации растворов и количестве веществ (в молях).

2. Введите данные о веществах (название, химическая формула, молярная масса).

3. Добавьте в таблицу столбцы для вычислений (масса вещества, масса раствора, масса воды). Введите соответствующие формулы:

$$\text{масса вещества} =$$

= молярная масса · количество вещества;

$$\text{масса раствора} = \frac{\text{масса вещества}}{\text{доля вещества}}.$$

4. Используя соответствующие функции, найдите массу самого тяжелого и самого легкого вещества.

5. Примените форматирование к ячейкам таблицы и данным в них. Используйте цветовые стили. Для химических формул используйте шрифт BatangChe (RGB:192/0/0; полужирный).

18.2. Ввод прогрессий

При вводе последовательностей можно растягивать диапазон, удерживая нажатой правую кнопку мыши. В этом случае из контекстного меню можно выбрать способ формирования последовательности (пример 18.3). Контекстное меню появляется сразу, как только закончили растягивать диапазон.

Если выбрать **линейное приближение**, то последовательность формируется на основе арифметической прогрессии,

Пример 18.2. Продолжение.

2.		Химическая формула	Молярная масса (а. е. м.)
4	Вещ-во		
5	Хлорид калия	KCl	74
6	Хлорид натрия	NaCl	58
7	Сульфат натрия	MgSO ₄	120
8	Нитрат аммония	NH ₄ NO ₃	80

3.		Химич. формула	Молярная масса (а. е. м.)	Масса вещ-ва (г)	Масса р-ра (г)	Масса воды (г)
4	Вещ-во					
5	Хлорид калия	KCl	74			
6	Хлорид натрия	NaCl	58			
7	Сульфат натрия	MgSO ₄	120			

4.	A	B	C	D	E	F
1	Растворы					
2	Доля вещества		26%			
3	Кол-во вещества (моль)		2			
4	Вещ-во	Химич. формула	Молярная масса (а. е. м.)	Масса вещ-ва (г)	Масса р-ра (г)	Масса воды (г)
5	Вещ-во калия	KCl	74			
6	Хлорид натрия	NaCl	58			
7	Сульфат магния	MgSO ₄	120			
8	Нитрат аммония	NH ₄ NO ₃	80			
9	Бромид бария	BaBr ₂	297			
10						
11	Самое тяжелое вещество					
12	Самое легкое вещество					

Пример 18.3. Выбор вида последовательности:

Линейное приближение
Экспоненциальное приближение
Мгновенное заполнение
Прогрессия...

Выбор типа прогрессии:

Прогрессия ? x

Расположение	Тип	Единицы
☐ по строкам	☐ арифметическая	☒ день
☒ по столбцам	☒ геометрическая	☐ рабочий день
	☐ даты	☐ месяц
	☐ автозаполнение	☐ год

☒ Автоматическое определение шага

Шаг: 1 Предельное значение:

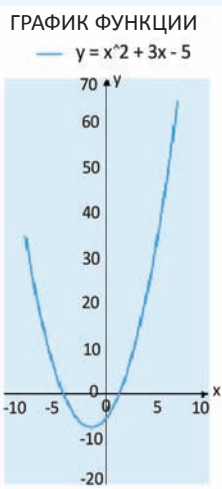
Пример 18.4. Ввести последовательность из 10 чисел в первый столбец электронной таблицы. Первое число 1, а каждое следующее в 3 раза больше предыдущего:

1. Ввести число 1 в ячейку A1.
2. Ввести число 3 в ячейку A2.
3. Выделить диапазон A1:A2.
4. Используя маркер заполнения, растягивать диапазон, удерживая нажатой правую кнопку мыши.
5. Закончить растяжку диапазона в 10-й строке. Из контекстного меню выбрать экспоненциальное приближение.

	A
1	1
2	3
3	9
4	27
5	81
6	243
7	729
8	2187
9	6561
10	19683

Пример 18.5. Построение графика функции $y = x^2 + 3x - 5$. Таблица значений и график функции:

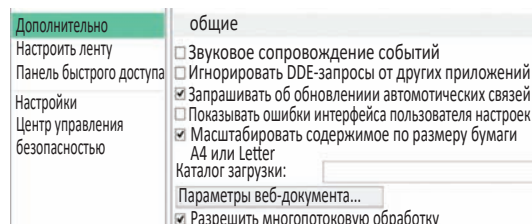
	A	B
1	x0	-9
2	шаг	1
3		
4	x	y
5	-9	49
6	-8	35
7	-7	23
8	-6	13
9	-5	5
10	-4	-1
11	-3	-5
12	-2	-7
13	-1	-7
14	0	-5
15	1	-1
16	2	5
17	3	13
18	4	23
19	5	35
20	6	49



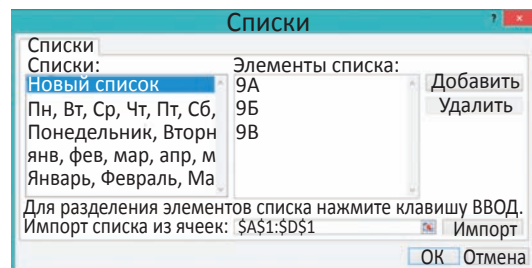
а если экспоненциальное — то на основе геометрической. Для формирования других типов последовательностей нужно выбрать **Прогрессия**.

В примере 18.4 показано, как ввести последовательность чисел, используя выбор типа последовательности.

Можно создавать свои списки, перечисляя их элементы. Для этого нужно выполнить команду **Файл** → **Параметры** → **Дополнительно**, раздел **Общие**, кнопка **Изменить списки**.



Далее создать или отредактировать список:



18.3. Аналитическая графика

Пример 18.5. Построить график функции $y = x^2 + 3x - 5$ на промежутке $[-9; 9]$.

1. Создайте таблицу значений функции. Шаг изменения аргумента равен 1.

2. Для построения диаграммы выделите диапазон: A4:B21. Тип диа-

граммы — **Точечная**, вид можно выбрать из двух:  или .

3. Введите подписи осей X и Y.

Примечание. Подписи осей вписаны в надписи, которые являются объектами. Их можно перемещать в области диаграмм.

4. Поместите легенду вверху диаграммы.

5. Уберите линии сетки и закрасьте фон области построения диаграммы.

6. Изменяя начальные значения x_0 и шага, постройте график вблизи точки пересечения графика функции и оси OX. При наведении указателя мыши на точку пересечения с осью появляется значение.

Примечание. Если требуется большая точность, то можно еще уменьшить шаг.

Пример 18.6. Используя данные таблицы «Медальный зачет», выполните:

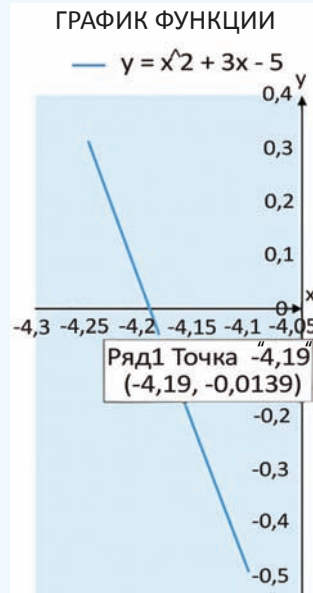
1. Расчет общего количества медалей для каждой страны и общего количества разыгранных медалей. Примените форматирование к ячейкам с результатом вычислений.

2. Построение линейчатой диаграммы распределения золотых медалей в десятке лидеров. Настройки диаграммы: подписи данных — снаружи; направление текста в надписи у вертикальной оси — повернуть на 270° ; в параметрах горизонтальной оси значение максимума равно 14.

3. Расчет количества очков, набранных командами каждой страны.

Пример 18.5. Продолжение.

Определение корня при $x_0 = -4,25$ и шаге 0,01 (корень $\approx -4,19$).



Пример 18.6. Вычисления в таблице «Медальный зачет зимних Олимпийских игр — 2018».

1. **Медальный зачет зимних Олимпийских игр–2018**

Страна	Золото	Серебро	Бронза	Всего
Норвегия	14	14	11	39
Германия	14	10	7	31
Канада	11	8	10	29
США	9	8	6	23
Нидерланды	8	6	6	20
Швеция	7	6	1	14
Южная Корея	5	8	4	17
Швейцария	5	6	4	15
Франция	5	4	6	15
Австрия	5	3	6	14
Новая Зеландия			2	2
Казахстан			1	1
Латвия			1	1
Лихтенштейн			1	1
Хорватия				0
Всего	103	102	102	307

Пример 18.6. Продолжение.

2.



3.

	A	B	C	D	E	F
1	Медальный зачет зимних Олимпийских игр-2018					
2	Количество очков					
3	золото	3				
4	серебро	2				
5	бронза	1				
6	Страна	Золото	Серебро	Бронза	Всего	Очки
7	Норвегия	14	14	11	39	81
8	Германия	14	10	7	31	69
9	Канада	11	8	10	29	59

4.



Пример 18.7. Сортировка и применение фильтра в таблице «Глубочайшие пещеры мира».

1.

2	№	Название	Глубина, м	Длина, м	Страна
3	1	Ану Иффлис			Алжир
4	2	Эгма			Турция
5	3	Кузгун			Турция

Для этого вставьте после заголовка таблицы строки и введите в них данные об очках, начисляемых за медали. В столбце F введите формулы, позволяющие рассчитать количество очков для каждой страны. Примените оформление в соответствии с образцом.

4. Построение круговой (кольцевой) диаграммы по данным об очках, набранных каждой страной. Настройки диаграммы:

- стиль диаграммы — 7;
- легенда справа (откорректируйте ее положение и размер до полного отображения данных);
- в параметрах ряда данных значение диаметра отверстия равно 50 %.

18.4. Распределение данных и организация поиска

Пример 18.7. Используя данные таблицы «Глубочайшие пещеры мира», выполните:

1. Сортировку:

- по глубине в порядке убывания;
- по длине в порядке возрастания.

2. Сортировку по столбцу «Страна» в алфавитном порядке.

3. С помощью фильтра найдите:

- пещеры, глубина которых меньше 1200 м;
- пещеры, которые расположены в Мексике;
- пещеры, глубина которых не менее 1340 м и не более 1350 м.

Примечание. Перед выполнением каждой сортировки или применением нового фильтра создайте копию

листа, на котором расположена исходная таблица.

Пример 18.8. С помощью фильтра из таблицы «Прайс-лист» выберите товар, соответствующий запросу.

1. Ноутбук, розничная цена <1000 р., не под заказ.

Пользовательский автофильтр

Показать только те строки, значения которых:

Наименование товара

равно Ноутбук

☒ И ☐ ИЛИ

Знак вопроса "?" обозначает один любой знак
Знак "*" обозначает последовательность любых знаков

OK Отмена

Пользовательский автофильтр

Показать только те строки, значения которых:

Розничная цена

меньше 1000

☒ И ☐ ИЛИ

Знак вопроса "?" обозначает один любой знак
Знак "*" обозначает последовательность любых знаков

OK Отмена

Пользовательский автофильтр

Показать только те строки, значения которых:

Наличие

не равно Под заказ

☒ И ☐ ИЛИ

Знак вопроса "?" обозначает один любой знак
Знак "*" обозначает последовательность любых знаков

OK Отмена

2. Жесткий диск или твердотельный накопитель, розничная цена от 150 до 200 р.

3. Мышь и клавиатура, цвет белый, не беспроводная.

Пример 18.7. Продолжение.

2.

№	Название	Глубина, м	Длина, м	Страна
1	Лампрехтсофен	1 735	51 000	Австрия
2	Хирлатцхёле	1 560	112 929	Австрия
3	Ану Иффлис	1 170	2 000	Алжир
34	Зибенхэнгсте	1 340	157 000	Швейцария
35	Пьер-Сен-Мартин	1 408	83 649	Франция/Испания

3.

N	Название	Глубина	Длина	Страна
2	Ану Иффлис	1 170	2 000	Алжир
8	Казумура	1 102	65 000	США
2	Абиссо Паоло	1 560	75 602	Мексика
32	Уатла	1 278	6 710	Мексика
34	Чарко	1 488	45 949	Мексика
35	Чеве	1 350	21 000	Италия
7	Зибенхэнгсте	1 340	157 000	Швейцария
Сима де лас	Пуэртас	1 344	19 000	Испания
24	Илламинако	1 349	45 246	Испания
26	Система Аранонера			

Пример 18.8. Применение фильтра в таблице «Прайс-лист».

1.

	A	C	D
4	Наименование товара	Розничная цена	Наличие
11	Ноутбук	826,73 р.	Мало
12	Ноутбук	826,73 р.	Мало
13	Ноутбук	800,55 р.	Мало
14	Ноутбук	573,49 р.	Достаточно
15	Ноутбук	538,38 р.	Мало

2.

	A	C
4	Наименование товара	Розничная цена
150	Жесткий диск	175,56 р.
157	Жесткий диск	163,86 р.
177	Твердотельный накопитель	168,54 р.
186	Твердотельный накопитель	156,54 р.
1014	Жесткий диск	166,19 р.

3.

	A	B
4	Наименование товара	Марка и характеристики
634	Мышь	LOGITECH B100 (910-003360) белый
636	Мышь	LOGITECH B100 (910-001605) белый



1. В каких сферах деятельности человека могут быть использованы вычислительные возможности электронных таблиц?
2. Как с помощью диаграммы типа **Точечная** можно найти корень уравнения?
3. Почему применение сортировки и фильтрации к данным электронных таблиц увеличивает эффективность поиска информации?



Упражнения

- 1 Выполните сортировку данных в таблице из примера 18.1 по убыванию роста.
- 2 Для примера 18.2 выполните перечисленные задания.
 1. Добавьте в таблицу данные о других веществах. Выполните для этих веществ расчет масс вещества, раствора и воды.
 2. Измените концентрацию веществ. Как это влияет на результаты расчетов?
- 3 Используя возможности ввода геометрической прогрессии, получите таблицу умножения.
- 4 Найдите второй корень уравнения из примера 18.5.
- 5 Постройте по данным таблицы из примера 18.6 гистограмму по количеству золотых медалей, завоеванных представителями всех стран.
- 6 Произведите поиск в таблице из примера 18.8 флеш-накопителя марки KINGSTON на 64 GB.
- 7 Постройте графики функций.
 1. $y = 0,3x^2 + 3x - 5$ на промежутке $[-3; 3]$.
 2. $y = |3x - 4| - 5$ на промежутке $[-6; 6]$.
 3. $y = x \sin x$ на промежутке $[-10; 10]$.
- 8* Графически найдите корни уравнений.
 1. $x^2 + x + 6 = 0$.
 2. $x^3 + x^2 - 6x - 3 = 0$.
- 9 Постройте в одной системе координат графики двух функций. Найдите точки пересечения графиков.
 1. $y = 0,3x^2 + 3x - 5$ и $y = -5x^2 - 2x + 10$ на промежутке $[-7; 7]$.
 2. $y = |0,3x - 2|$ и $y = \cos x$ на промежутке $[0; 10]$.
 3. $y = 3 + \sqrt{5x + 2}$ и $y = 0,5x^2 + 2x - 8$ на промежутке $[-10; 7]$.
- 10* Графически решите уравнение $y = \sqrt{|x - 3|}$.

Глава 4

КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ МОДЕЛИ

§ 19. Модели и виды моделей

19.1. Понятие модели

Модель (латинское *modulus* — образец) как понятие возникло в античные времена и связано с практической деятельностью человека.

Модель — объект или процесс, который для каких-либо целей рассматривается вместо другого объекта или процесса.

В обыденной жизни модель — это повторение в уменьшенном масштабе реального объекта, который называют *моделируемым объектом* или *объектом-оригиналом* (пример 19.1).

Модель всегда является лишь приближенным подобием объекта-оригинала и отображает не все его свойства, а только те, которые существенны для решаемой задачи.

Модель, объемно представляющая только внешний вид объекта-оригинала, называют *макетом* (пример 19.2).

Объектом-оригиналом для модели может быть и вымышленный объект. К примеру, макеты будущих зданий и сооружений используются в архитектуре и строительстве. Именно такие модели с древнейших времен обеспечивают технологию обработки металлов литьем (пример 19.3).

В современном понимании модель не всегда внешне похожа на моделируемый объект и не всегда создается

Пример 19.1. Глобус — модель земного шара, манекен — модель фигуры человека, многие детские игрушки — модели реальных объектов.



Пример 19.2. Макет Национальной библиотеки Республики Беларусь в Музее государственности Республики Беларусь.



Пример 19.3. В III тыс. до н. э. уже использовались предметы из бронзы, которые были изготовлены путем отливки в глиняные или песчаные формы. Формы создавались по моделям-образцам (отсюда и пошло *modulus*). С тех пор эта технология совсем не изменилась.



Бронзовый топор,
конец II — начало I тыс. до н. э.

Пример 19.4. Деловые игры являются моделями реальных процессов коллективного управления, процесс испытаний автомобиля является моделью процесса его эксплуатации, детские игры часто моделируют отношения взрослых.

Пример 19.5. Глобус — материальная модель земного шара, то же самое можно сказать о многих игрушках, об объемных макетах зданий и сооружений.

Пример 19.6. Фотография велосипеда дает представление об этом материальном объекте, представляет информацию о нем. Тогда данная фотография — это информационная модель велосипеда.



А учебное пособие по истории, к примеру, содержит информацию о различных исторических событиях. Тогда с точки зрения информатики оно содержит информационные модели различных исторических событий.

Пример 19.7. Каждый имеет в сознании мысленную модель квартиры, где он живет. С ее помощью даже в полной темноте можно легко ориентироваться в квартире.

Хороший автомеханик всегда знает, как устранить неисправность в автомобиле, потому что в его сознании существует мысленная модель исправного автомобиля.

Специалист в любой области отличается именно тем, что в его сознании есть достаточное количество мысленных моделей, которые он использует в работе.

искусственно. Например, летчики, показывая друг другу особенности фигур высшего пилотажа, часто используют ладонь руки. Ладонь в этом случае становится моделью самолета.

Модели реальных процессов на практике используются также очень широко (пример 19.4).

19.2. Основные виды моделей

По определению модель — это некоторый объект или процесс. В основу деления моделей на виды положена форма их представления.

Модель может быть представлена в форме материального объекта, мысленного образа, бумажного документа, файла в памяти компьютера. Поэтому различают два основных вида моделей: *материальные* и *информационные*.

Материальная модель — модель, которая является материальным объектом.

Материальные модели наиболее известны. Иногда их также называют *предметными*. До сих пор мы рассматривали именно материальные модели (пример 19.5).

Информационная модель — модель, которая является информацией на некотором носителе.

Информационные модели представляют информацию об объекте-оригинале (пример 19.6).

19.3. Виды информационных моделей

По виду используемого носителя информации информационные модели делятся на *мысленные*, *документальные* и *компьютерные*.

Мысленная модель — это информационная модель в форме мысленного образа.

Человек способен представить в своем воображении самые разные предметы, ситуации, явления. Такие представления человека и есть мысленные модели (пример 19.7). Мысленные модели воображаемых объектов дают простор творчеству инженеров и конструкторов, художников и дизайнеров.

Документальная модель — информационная модель в форме бумажного документа.

Документальные модели уже были рассмотрены в примере 19.6. Еще несколько таких моделей описаны в примере 19.8.

Компьютерная (электронная) модель — информационная модель в форме файла на компьютерном носителе информации и ее изображение на экране компьютера.

Образец компьютерной модели показан в примере 19.9.

Многие документальные модели были переведены в электронную форму и стали компьютерными моделями.

Со многими объектами окружающего мира мы знакомы только по их компьютерным моделям.

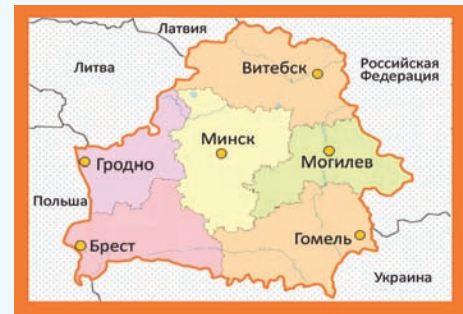
19.4. Компьютерные информационные модели

В последнее время при моделировании преобладает использование компьютерных информационных моделей. Это связано с тем, что возможности

Пример 19.8. Документальная модель может содержать информацию в текстовой, табличной и графической формах. Некоторые документальные модели можно представить только в табличной форме.

Дата	Осадки	Температура днем	Температура ночью
10.02.2019	Без осадков	+2°	-2°
11.02.2019	Мокрый снег	+1°	0°
12.02.2019	Без осадков	+1°	+2°
13.02.2019	Снег	0°	-1°
14.02.2019	Без осадков	+1°	+1°

Документальными графическими моделями являются бумажные чертежи машин и механизмов, карты.



Пример 19.9. Компьютерная модель Мирского замка.



Пример 19.10. Специалистами по численному моделированию с помощью компьютерных моделей на суперкомпьютерах выполнялся расчет распространения крупномасштабных поверхностных волн в морях и океанах, проводилось моделирование климата и его изменений, механизмов образования галактик, решались проблемы глобальных экологических катастроф.

На компьютерных моделях проведена обработка сейсмических данных, в результате которой были обнаружены ранее неизвестные месторождения природных ресурсов на Сахалине и в Казахстане.

Пример 19.11. К программным средствам моделирования относятся системы компьютерной математики MathCAD, Mathematics, MATLAB, Maple. Они используются в случаях, когда математическая модель уже построена и ее требуется только исследовать.

Пакет прикладных математических программ SCILAB используется для проведения инженерных, технических и научных расчетов.

Программное средство LabVIEW позволяет не только моделировать объекты, но и проводить их испытания, измерения и управлять ими.

компьютеров заметно возросли и позволяют создавать компьютерные модели любой сложности.

Компьютерные модели обладают универсальными возможностями. С одной стороны, они могут заменить документальные модели (с помощью текстовых и графических редакторов), с другой — материальные (пример 19.10).

Таким образом, компьютерные модели позволяют в процессе моделирования обойтись без документальных и материальных моделей.

19.5. Средства реализации компьютерных моделей

Компьютерные модели могут быть созданы с помощью разных программных средств. Среди них:

- текстовые редакторы (построение текстовых и табличных моделей);
- графические редакторы (построение графических моделей);
- электронные таблицы (построение табличных и графических моделей);
- системы программирования (построение моделирующих программ).

Существуют и другие программные средства, которые позволяют создавать компьютерные модели (пример 19.11).



1. Что такое модель?
2. Что такое макет?
3. Всегда ли модель внешне похожа на объект-оригинал?
4. Какие особенности моделей считаются основными?
5. Что такое материальная модель?
6. Что такое информационная модель?
7. Что такое мысленная модель?
8. Что такое документальная модель?
9. Что такое компьютерная модель?
10. В чем состоят преимущества компьютерных моделей перед документальными и материальными моделями?



Упражнения

- 1 Приведите примеры материальных моделей.
- 2 Приведите примеры мысленных моделей.
- 3 Приведите примеры документальных моделей.
- 4 Приведите примеры компьютерных моделей.
- 5 Повторите таблицу в тетради и дополните ее столбцы примерами моделей для данных объектов.

Примеры моделей	
Объект «человек»	Объект «планета Земля»
Манекен	Глобус

§ 20. Моделирование и его этапы

20.1. Моделирование

Моделирование — процесс создания и использования моделей для решения научных и практических задач.

Жизнь постоянно ставит перед человечеством самые разные задачи. Моделирование позволяет решать эти задачи быстрее и с меньшими затратами средств.

Варианты решения возникающих задач все чаще проверяются на моделях, потому что испытания сложных и дорогостоящих систем требуют больших затрат и не всегда возможны в принципе (пример 20.1).

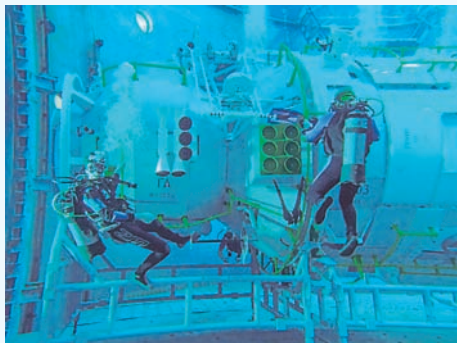
Хорошо известен пример использования моделирования при строительстве купола Флорентийского собора (пример 20.2).

Пример 20.1. Для изучения сейсмической устойчивости зданий их не разрушают, а строят и испытывают на вибрационном стенде модели.

Пример 20.2. В эпоху итальянского Возрождения при сооружении купола Флорентийского собора архитектор Филиппо Брунеллески использовал его модель как образец.



Пример 20.3. Гидролаборатория в российском Центре подготовки космонавтов имени Ю. А. Гагарина позволяет провести тренировку по оказанию помощи космонавтам в критических ситуациях.



Пример 20.4. Первую подробную геоцентрическую модель движения Солнца, Луны и планет построил Клавдий Птолемей во II в. н. э. Планеты двигались по замысловатым траекториям, но частичная адекватность была достигнута.



Пример 20.5. Направляясь из школы домой, задачу выбора правильного пути каждый решает в уме (с помощью мысленной модели). Модель адекватно отражает реальную дорогу.

В наше время полеты космических кораблей и станций всегда сопровождаются постройкой макетов космических объектов в натуральную величину в бассейнах с водой (пример 20.3).

Существуют задачи, для решения которых модель исследовать не нужно. В таких задачах модели используются в других целях. Например, в музеях и при проектировании макеты зданий используются для демонстрации.

Модель создается для того, чтобы отразить информацию о некоторых свойствах моделируемого объекта или процесса.

Адекватность модели — степень соответствия свойств модели тем свойствам объекта-оригинала, которые она отражает.

Оценка адекватности модели может быть самой разной: от неадекватности и частичной адекватности до полной адекватности.

Частичная адекватность может возникать, когда численные значения свойств модели и оригинала отличаются на небольшую величину (пример 20.4).

20.2. Научные методы в моделировании

В процессе моделирования любая модель сначала создается в сознании исследователя и является мысленной. Этого достаточно только при решении простых задач (пример 20.5). Для решения более сложных задач нужно строить и документальную модель (пример 20.6).

При создании очень сложных моделей обычно используются научные методы, формулы, законы. Каждая наука при описании фактов и явлений пользуется своим языком (пример 20.7).

Используемую при создании модели науку обычно указывают (*математика, биология* и др.), или модель получает название (*математическая, биологическая* и др.). Если модель с конкретной отраслью знаний не связана, то ее называют *простейшей*.

20.3. Этапы компьютерного моделирования

В настоящее время компьютерные информационные модели являются основным видом моделей, используемых для моделирования.

Процесс компьютерного моделирования всегда занимает какое-то время и проходит в несколько этапов. Основными из них являются:

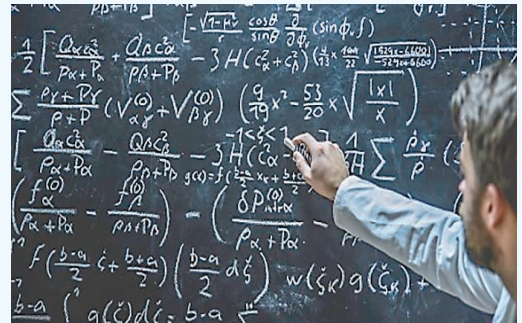
1. Постановка задачи.
2. Выбор плана создания модели.
3. Создание модели.
4. Исследование модели.
5. Получение решения задачи.

Постановка задачи — необходимый этап, который придает определенность всему процессу моделирования.

Для создания реальной компьютерной модели, конечно, необходимо мысленное представление о ней. Это будет *мысленная модель*. Чтобы по мысленной модели получить компьютерную, могут понадобиться и другие виды моделей (пример 20.8). Создание

Пример 20.6. Большинство задач по математике трудно решать в уме, и нужно вести записи. Это и есть документальные модели.

Пример 20.7. Языком науки обычно называют формулы.



Для математики это язык цифр и математических формул, для физики — язык физических формул и закономерностей. Свои языки имеют химия, биология, социология и т. д.

Пример 20.8. В процессе создания сложной модели одна форма представления может сменять другую. В разных задачах могут использоваться одна, две или три формы представления.

Для реализации мысленной модели могут понадобиться записи (текстовая документальная модель), формулы (математическая документальная модель) или чертежи (графическая документальная модель).

С другой стороны, компьютерная модель должна создаваться с помощью какого-либо программного средства (текстового редактора, электронной таблицы, системы программирования).

Пример 20.9. Пусть требуется выяснить, какова аэродинамика нового автомобиля на высокой скорости (этап 1). Моделировать надо внешнюю форму самого автомобиля и силы, которые на него воздействуют в скоростном потоке воздуха. Чтобы построить компьютерную модель, на втором этапе моделирования решается, что нужно сначала создать документальную математическую модель автомобиля и потоков воздуха, а затем компьютерную.

В этом и состоит план создания компьютерной модели. На четвертом этапе для известных режимов проверяется ее адекватность. На пятом этапе в результате использования компьютерной модели получают оценки аэродинамических свойств нового автомобиля. Задача получает решение.

компьютерной модели может проходить через создание ряда разных моделей. План очередности этих моделей выбирается на втором этапе моделирования. Дополнительные этапы мы будем нумеровать 3а, 3б и т. д.

Этап исследования компьютерной модели является проверкой ее адекватности. Часто на этом этапе тестированием с известными данными проверяется, насколько точно модель отражает реальный объект или процесс.

Последний, пятый этап связан с получением результатов моделирования и решением поставленной задачи (пример 20.9).



1. Что такое моделирование?

2. Почему моделирование позволяет решать сложные задачи быстрее и с меньшими затратами средств?

3. Как используемые при моделировании законы некоторой науки отражаются в названии модели?

4. В чем состоит план создания модели в процессе моделирования?



Упражнения

1 Приведите примеры использования моделирования.

2 Перечислите этапы моделирования.

3 Приведите примеры задач, при решении которых вслед за мысленной моделью сразу можно построить компьютерную модель.

4 Приведите примеры задач, при решении которых компьютерную модель невозможно построить сразу после мысленной модели.

§ 21. Моделирование в задаче роста и убывания

Экологические проблемы в современном мире становятся все значительнее и постепенно выходят на первый план. Этому способствуют в первую очередь техногенные факторы, связанные с массовым производством.

21.1. Постановка задачи (этап 1)

Задача. Объем деловой древесины на лесном участке оценивается в 150 тыс. м³. Каждый год этот объем увеличивается на 7 % за счет естественного прироста. Начиная с 3-го года

на хозяйственные нужды вырубается 20 тыс. м³ древесины ежегодно.

Вопросы:

1. Наступит ли уменьшение объема деловой древесины на участке до 100 тыс. м³ и через сколько лет?

2. Каков ответ на первый вопрос, если, начиная с 4-го года, процент прироста уменьшится до 5 %, а вырубка с 7-го года увеличится до 25 тыс. м³ в год?

21.2. Выбор плана создания модели (этап 2)

Объектом исследования является объем деловой древесины на лесном участке.

Будем строить таблицу, рассчитывая объемы древесины по годам (пример 21.1). Такую таблицу можно получить расчетами вручную или с помощью электронных таблиц (компьютерная модель).

Выбираем компьютерную модель. Причем для ее построения необходимо сначала построить документальную математическую модель с математическими формулами для расчетов. Следовательно, создание модели будет проходить в два этапа:

- 3а — создание документальной математической модели;
- 3б — создание компьютерной модели с помощью электронных таблиц.

21.3. Создание документальной математической модели (этап 3а)

Перенумеруем годы, начиная с нулевого (начального). Пусть объем деловой древесины в i -м году будет $D(i)$, процент прироста древесины — $P(i)$, а объем вырубки — $V(i)$.

Бездумное воздействие общества на природу перешагнуло грань, которая позволяла неограниченно использовать природные ресурсы. Результатом стало загрязнение значительных территорий суши и вод Мирового океана бытовыми отходами и отходами промышленного производства, сокращение запасов полезных ископаемых, лесных и рыбных ресурсов. Теперь принятие любого производственного решения требует просчета всех его последствий.

Пример 21.1. В таблицах для расчетов по годам объемов ресурсов (сумм денежных вкладов, регулярных денежных расходов и т. п.) строки связаны с номерами лет.

Столбцы таблицы содержат объемы ресурсов (денежные суммы), проценты прироста или убывания, объемы прироста или убывания по каждому году.

Год	Сумма	Процент
0	200	5
1	210	4
2	218,40	3
3	224,95	2
4	229,45	4
5	238,63	5
6	250,56	6
7	265,59	7
8	284,19	3
9	292,71	5
10	307,35	6
11	325,79	5

Любая таблица в задачах роста и убывания построена на основе формулы прироста (убывания), которая по одной строке таблицы вычисляет значение объема ресурса для следующей строки (начало следующего года). Нумерацию лет можно начинать с 0 или брать фактические номера лет.

Пример 21.2. Схема размещения данных и заголовков модели в электронных таблицах.

	A	B	C	D	E
1		Модель роста и убывания			
2	Исходные данные				
3	150000	: начальный объем древесины (куб.м)			
4	7	: начальный процент прироста			
5	0	: начальный объем вырубки			
6					
7	Расчетная таблица				
8	Год	Объем	Процент	Вырубка	
9					

Длинные тексты (название модели, названия разделов модели, описания данных) вводятся в ячейки, которые на схеме содержат их первые буквы. Описания данных начинаются знаком «:» (двоеточие), что позволяет отделить описания от численных значений.

Строка заголовков столбцов рабочей таблицы (ячейки диапазона A8:D8) залита серым цветом в результате выбора этого цвета в меню кнопки **Цвет заливки** (на панели **Главная** в группе **Шрифт**).

Пример 21.3. Начало рабочей таблицы модели до ввода данных по вырубке, которая начинается с 3-го года.

Расчетная таблица			
Год	Объем	Процент	Вырубка
0	150000	7	0
1	160500	7	0
2	171735	7	0
3	183756	7	0

Пример 21.4. Для проверки адекватности модели достаточно проверить правильность вычислений во второй строке расчетной таблицы.

Вычислениями по формуле на калькуляторе можно получить, что объем древесины на начало первого года должен составить 160 500 м³, что соответствует данным расчетной таблицы.

Тогда объем древесины в $(i + 1)$ -м году вычисляется по формуле

$$D(i + 1) = D(i) \cdot (1 + P(i)/100) - V(i).$$

21.4. Создание компьютерной модели (этап 3б)

Компьютерную модель будем строить, используя схему размещения данных и заголовков, приведенную в примере 21.2.

В первой строке расчетной таблицы (в строке 9) будут отображаться начальные данные для нулевого года, поэтому вводим:

$$A9: 0 \quad B9: =A3 \quad C9: =A4 \quad D9: =A5$$

Далее основная для расчетов строка 10:

$$A10: 1$$

$$B10: =B9*(1 + C9/100) - D9$$

$$C10: =C9$$

$$D10: =D9$$

Теперь заполним еще 20 строк расчетной таблицы (включая строку 30). Столбец A заполняем вниз содержимым диапазона A9:A10. Затем выделяем диапазон B10:D10 и заполняем вниз содержимым этого диапазона еще 20 строк (пример 21.3).

Так как с 3-го года начались вырубки, в столбец «Вырубка» в строке с номером года 3 вводим число 20 000.

21.5. Исследование модели (этап 4)

Полная адекватность модели сомнений не вызывает, т. к. данные и формулы модели полностью соответствуют данным задачи. Адекватность также может быть проверена тестированием, т. е. расчетами вручную (пример 21.4).

21.6. Получение решения задачи (этап 5)

Для ответа на первый вопрос задачи анализируем расчетную таблицу (пример 21.5).

Для ответа на второй вопрос вносим изменения в графы «Процент» (для 4-го года) и «Вырубка» (для 8-го года) и анализируем таблицу (пример 21.6).

Пример 21.5. Ответ на первый вопрос задачи: объем деловой древесины на участке уменьшится до 100 тыс. м³ через 12 лет.

Пример 21.6. Ответ на второй вопрос задачи: объем деловой древесины на участке может уменьшиться до 100 тыс. м³ уже через 9 лет.



Упражнения

- 1 Перечислите этапы моделирования в задаче роста и убывания.
- 2 Повторите на компьютере решение задачи роста и убывания, рассмотренное в параграфе.
- 3 Изменив построенную модель роста и убывания, решите следующую задачу: «На пищевом комбинате в установку по производству дрожжей заложена 1 т дрожжевой массы. При поддержании оптимальной температуры за сутки масса дрожжей возрастает на 150 %. 1,5 т массы ежедневно пускают в производство. В результате неполадки температура в установке повысилась, и прирост составил 160 % в сутки. Через сколько суток масса дрожжей в установке может достигнуть 3,5 т?»

4 Решите следующую задачу: «Закон изменения массы m колонии вируса гриппа имеет вид $m = m_0 2^{\frac{t}{\Delta t}}$, где m_0 — первоначальная масса колонии, t — время в часах, Δt — шаг времени в часах. Установите, через сколько часов масса колонии превысит 1,9 г, если $m_0 = 0,03$ г, $\Delta t = 1$ ».

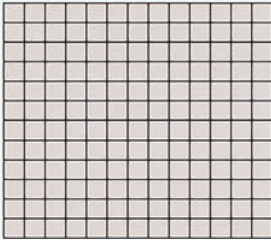
§ 22. Моделирование в задаче определения температурных режимов

22.1. Постановка задачи (этап 1)

Задача. Имеется квадратная однородная металлическая пластина, которая является деталью некоторого устройства. Во время работы устройства вокруг пластины и на ее краях поддерживается температура 20 °С. В центральной зоне пластина нагревается сварочной дугой до 6000 °С. Каким будет распределение температур во внутренних точках пластины?

При сварке металлического изделия сосредоточенным источником тепла производится интенсивный кратковременный местный нагрев металла до высоких температур. Тепло, выделяемое источником нагрева, расплавляет небольшой объем металла в месте сваривания и вследствие теплопроводности распространяется в прилегающие слои основного металла.

Пример 22.1. Разбиение пластины на квадраты.



Можно считать, что внутри каждого квадрата температура постоянна. Это предположение справедливо лишь приближенно, но чем меньше сторона квадрата, тем меньше ошибка.

Пример 22.2. Для учета состояния внешней среды по краям модели пластины создадим замкнутую цепочку квадратов, которые в модели будут представлять окружающую внешнюю среду. По условию квадраты внешней среды должны иметь постоянную температуру 20 °С.

Будем считать также, что сварочная дуга нагревает зону из четырех квадратов в центре пластины.

Пример 22.3. Чтобы перенастроить электронную таблицу, выделим ее всю, щелкнув по ячейке на пересечении строки и столбца заголовков.

На вкладке **Главная** в группе **Ячейки** щелкаем по кнопке **Формат**. Выпадает меню, в котором выбираем пункт **Высота строки...**. Появляется диалоговое окно, в котором для высоты ячейки вводим значение 26. Не снимаем выделение.

На вкладке **Главная** в группе **Ячейки** опять щелкаем по кнопке **Формат**. Выпадает меню, в котором выбираем пункт **Ширина столбца...**. Появляется диалоговое окно, в котором для ширины ячейки вводим значение 5.

Снять выделение с ячеек таблицы можно, щелкнув в любом месте электронной таблицы.

22.2. Выбор плана создания модели (этап 2)

При решении задачи используем метод дискретизации, при котором плоская пластина условно разбивается на небольшие квадраты (пример 22.1).

Для описания теплообмена между условными квадратами и теплоизлучения построим математическую модель, а для расчетов — электронные таблицы. Создание модели будет проходить в два этапа:

- **3а** — создание документальной математической модели;
- **3б** — создание компьютерной модели с помощью электронных таблиц.

22.3. Создание документальной математической модели (этап 3а)

Разобьем квадратную пластину на условные квадраты общим числом 20×20 . Из физических законов следует, что температуру $t_{\text{кв}}$ любого квадрата можно считать по формуле:

$$t_{\text{кв}} = 20 + \left(\frac{t_{\text{лкв}} + t_{\text{вкв}} + t_{\text{пкв}} + t_{\text{нкв}}}{4} - 20 \right) 0,8.$$

Здесь индекс «лкв» указывает на квадрат слева, «пкв» — на квадрат справа и т. д. Нужно учесть влияние и других факторов (пример 22.2).

22.4. Создание компьютерной модели (этап 3б)

Название модели в электронную таблицу вводить не будем, чтобы иметь возможность строить модель, начиная с ячейки A1.

При построении табличной модели каждому условному квадрату поставим в соответствие ячейку электронной таблицы. Удобно сделать ячейки модели квадратными (пример 22.3).

Температуру будем показывать в целых числах (пример 22.4).

Чтобы вся модель была видна в окне редактора, устанавливаем соответствующий масштаб отображения (пример 22.5). Снимаем выделение и переходим к заполнению табличной модели. Сначала вводим значение 20 в ячейки внешней среды. Для этого вводим 20 в ячейку A1. Заполняем вниз содержимым ячейки A1 ячейки строк до 22 включительно. Далее устанавливаем табличный курсор на ячейку A1 и заполняем вправо все ячейки до столбца V включительно (это 22 столбца). Аналогично заполняем диапазоны V1:V22 и A22:V22.

Можно переходить к вводу формул в квадраты-ячейки модели. Легко предвидеть, что во внутренних соседних ячейках будут стоять формулы, которые ссылаются друг на друга. Таблицы Excel без дополнительной настройки воспринимают такие перекрестные ссылки как ошибки. Таблицы нужно перенастроить (пример 22.6).

Теперь во внутреннюю ячейку пластины B2 вводим формулу

$$= 20 + ((A2 + B1 + C2 + B3) / 4 - 20) * 0,8$$

Ставим табличный курсор на ячейку B2 и заполняем вправо содержимым ячейки B2 всю строку до края пластины. Выделение не убираем и заполняем вниз диапазоном B2:B21 всю внутренность модели.

Чтобы визуально наблюдать нагрев пластины, используем условное форматирование, которое будет задавать цвет квадрата в зависимости от его температуры (пример 22.7).

Пример 22.4. Выделяем всю таблицу и на вкладке **Главная** в группе **Число** щелкаем по кнопке со стрелкой в правом нижнем углу группы. Появляется диалоговое окно. На закладке **Число** выбираем формат **Числовой** и правее в поле **Число десятичных знаков** ставим 0. Щелкаем по кнопке **ОК**.

Пример 22.5. Панель **Выбор масштаба** в правом нижнем углу окна программы.



Пример 22.6. Для перенастройки электронных таблиц на вкладке **Файл** щелкаем по пункту **Параметры**. Открывается диалоговое окно. На вкладке **Формулы** ставим галочку около надписи **Включить итеративные вычисления**, а в поле **Предельное число итераций** вводим число 500. Щелкаем по кнопке **ОК**.

Перенастройка закончена.

Пример 22.7. Для включения в ячейках пластины условного форматирования цветом в зависимости от значения числа в ячейке выделяем диапазон A2:U21 всей пластины. На вкладке **Главная** в группе **Стили** щелкаем по кнопке **Условное форматирование**. В выпадающем списке переводим указатель мыши на пункт **Цветовые шкалы**, справа появляется дополнительное меню, в котором щелкаем по пункту **Другие правила ...**.

Открывается диалоговое окно. В зоне **Минимальное значение** в строке **Цвет** выбираем светло-серый. В зоне **Максимальное значение** в строке **Цвет** выбираем ярко-красный, в строке **Тип** открываем список и выбираем пункт **Число**, а ниже в пункте **Значение** вводим число 1000.

Пример 22.8. Введем число 20 в ячейку K11. Если температура во внутренних квадратах не изменится, то модель адекватно отражает распределение температур.

Восстановим таблицу копированием в ячейку K11 формулы из любой соседней ячейки.

Обратите внимание на то, как в результате визуально проявляются автоматические вычисления. Через какое-то время модель выдает искомое распределение температур.

Одновременно нагрев пластины отображается визуально.

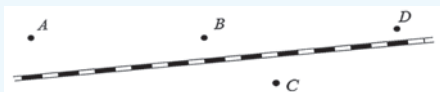


Упражнения

- 1 Повторите на компьютере решение задачи по определению температурных режимов, рассмотренное в параграфе.
 - 2 Каким будет распределение температуры во внутренних точках пластины, если в дополнение к условиям исходной задачи температура внешней среды по верхнему краю пластины равна 20 °С только в левом квадрате и возрастает на 50 °С с переходом к каждому следующему вправо квадрату?
 - 3 Каким будет распределение температуры во внутренних точках пластины, если в условиях исходной задачи сварочная дуга нагревает пластину до 6000 °С не в центральной зоне, а по вертикальной линии шириной один квадрат от верхнего края до нижнего?
- Рекомендация. Если в модели температурных режимов содержимым диапазона K10:L10 заполнить вниз диапазон со значениями 6000 и числом 20 верхнюю и нижнюю цепочку квадратов внешней среды, модель вернется в исходное состояние до нагрева.

§ 23. Моделирование в задаче выбора положения железнодорожной станции

Пример 23.1. Схема расположения железной дороги и населенных пунктов может выглядеть так, как показано на рисунке.



Расположение пунктов выше или ниже железной дороги роли не играет. Но два пункта не могут находиться на одной автомобильной дороге.

22.5. Исследование модели (этап 4)

Проверим адекватность модели. Для этого достаточно ввести число 20 (температуру воздуха) в центральную ячейку пластины вместо формулы (пример 22.8).

22.6. Получение решения задачи (этап 5)

Для ответа на вопрос задачи вводим в ячейки диапазона K11:L12 число 6000. Анализируя расчетную таблицу, приходим к выводу о симметричности распределения температуры во внутренних квадратах и о снижении температуры от центра к краям пластины.

23.1. Постановка задачи (этап 1)

Задача. В районе расположения четырех населенных пунктов A, B, C и D проходит прямолинейный участок железной дороги (пример 23.1).

По просьбе жителей в этом районе решено построить железнодорожную станцию и от нее прямолинейные автомобильные дороги к населенным пунктам.

Найти положение железнодорожной станции, при котором общая длина новых автомобильных дорог будет минимальной.

23.2. Выбор плана создания модели (этап 2)

Поскольку в задаче речь идет о положении пункта на местности (пример 23.2), сначала необходимо сформулировать математическую задачу (построить документальную математическую модель), а потом найти метод решения этой задачи.

Решение задачи будем искать с помощью электронных таблиц. Таким образом, модель создадим по следующему плану:

- 3а — создание документальной математической модели;
- 3б — выбор метода решения математической задачи;
- 3в — создание компьютерной модели с помощью электронных таблиц.

23.3. Создание документальной математической модели (этап 3а)

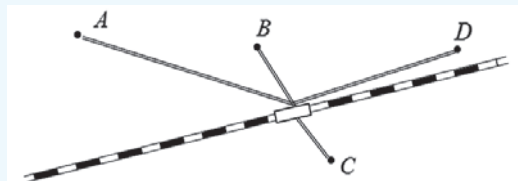
Чтобы составить математическую задачу, проще всего ввести прямоугольную систему координат и все объекты расположить на ней (пример 23.3).

По построению точка S имеет координаты $(x; 0)$, и нужно найти значение x , решающее данную задачу.

Для искомого x расстояние $|AS|$ между точками $A(x_1; y_1)$ и $S(x; 0)$ находится по формуле $|AS| = \sqrt{(x_1 - x)^2 + y_1^2}$.

Аналогично вычислив расстояния от S до остальных пунктов, можно найти сумму расстояний $f = |AS| + |BS| + |CS| + |DS|$.

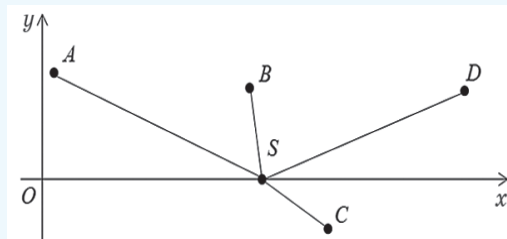
Пример 23.2. Объектом исследования является положение железнодорожной станции на участке железной дороги. На рисунке приведена одна из возможных схем расположения железнодорожной станции и автомобильных дорог.



Для определенности условий задачи должно быть точно задано положение населенных пунктов и железной дороги. Пользуясь подробной картой, эти данные можно получить.

Из пространственных соображений ясно, что нужное положение станции должно существовать.

Пример 23.3. По железной дороге направим ось OX , а ось OY построим левее населенных пунктов. Обозначим положение станции точкой S .



В этой системе координат каждый населенный пункт и станция получают свои координаты.

Будем считать, что работа с картой проведена и координаты (в км) вычислены: $A(1; 4)$, $B(4; 3)$, $C(5; -2)$, $D(8; 2)$.

Пример 23.4. Из анализа чертежа на координатной плоскости следует, что нет смысла брать точку S левее перпендикуляра к оси OX через крайнюю левую точку A .

Аналогично получается и правая граница значений x . Таким образом, для наших исходных данных получаем, что имеет смысл рассматривать значения x только на отрезке $[1; 8]$.

Выбираем начальное значение $x = 1$. Шаг переменной x возьмем равным $0,25$, т. е. расчеты будем вести в 29 точках:

$$(8 - 1) / 0,25 + 1 = 29.$$

Пример 23.5. Схема размещения данных и заголовков модели. Длинные заголовки вводятся в ячейку, которая на схеме содержит их первые буквы.

	A	B	C	D	E	F	G
1	Модель выбора положения						
2	Исходные данные						
3	Пункты	A	B	C	D		
4	Коорд. x	1	4	5	8		
5	Коорд. y	4	3	-2	2		
6							
7	1 : начальное значение x						
8	0,25 : шаг переменной x						
9							
10	Расчетная таблица						
11	x	AS	BS	CS	DS	f(x)	
12							

Координаты населенных пунктов сведены в таблицу. В таблицах ячейки с заголовками столбцов залиты серым цветом, и данные в них отцентрированы. Описания начальных данных ниже таблицы начинаются знаком «!» (двоеточие).

Пример 23.6. В расчетной таблице используется заполнение вниз содержимым ячейки $A13$ диапазоном $A13:A30$ и заполнение диапазоном $B12:F12$ диапазоном $B12:F30$.

23.4. Выбор метода решения математической задачи (этап 3б)

Сумма расстояний f меняется при изменении x (при изменении положения станции). Таким образом, в задаче построена функция $f(x)$ и нужно найти, при каком значении x она принимает минимальное значение.

Это типичная задача исследования функций в электронных таблицах, для решения которой нужно построить таблицу значений функции.

А для построения таблицы нужно найти начальное значение и шаг переменной x (пример 23.4).

23.5. Создание компьютерной модели (этап 3в)

Исходные данные и начало расчетной таблицы разместим по схеме, приведенной в примере 23.5.

Первая ячейка первой строки расчетной таблицы (12-й строки рабочей таблицы) повторяет начальное значение x : $A12: =A7$.

В следующую ячейку строки вводится первая формула

$$B12: =КОРЕНЬ((B\$4-\$A12)^2+B\$5^2)$$

Абсолютные ссылки нужны для обеспечения копирования формул.

Диапазон $C12:E12$ заполняется вправо содержимым ячейки $B12$. В последней ячейке первой строки проводится суммирование

$$F12: =СУММ(B12:E12)$$

В следующей строке $A13: =A12+A\$8$.

Далее в расчетной таблице используется заполнение (пример 23.6).

23.6. Исследование модели (этап 4)

Проверку адекватности расчетной модели проведем тестированием с помощью известных результатов (пример 23.7). Результат расчетов согласуется с данными расчетной таблицы — 19,99. Модель адекватна.

23.7. Получение решения задачи (этап 5)

Для ответа на вопрос задачи анализируется графа «Сумма» расчетной таблицы и находится значение x , при котором функция «Сумма» принимает минимальное значение.

Значение x следует уточнить (пример 23.8).

Пример 23.7. Используя калькулятор, посчитаем с помощью формул математической модели сумму расстояний от точки S до остальных точек при $x = 1$.

$$\begin{aligned} S &= \sqrt{16} + \sqrt{9+9} + \sqrt{16+4} + \sqrt{49+4} = \\ &= 4 + 4,2426 + 4,4721 + 7,2801 = \\ &= 19,9948 \end{aligned}$$

Пример 23.8. Для уточнения значения нужно уменьшить шаг переменной x и задать в модели новое начальное значение, расположенное в таблице выше найденного значения x . Таким способом положение станции можно найти с любой точностью.

**Упражнения**

- 1 Повторите на компьютере решение задачи по выбору положения железнодорожной станции, рассмотренное в параграфе.
- 2 Найдите решение исходной задачи для населенных пунктов с координатами: $A(2; 6)$, $B(3; -3)$, $C(7; 5)$, $D(10; -5)$.
- 3 Найдите решение исходной задачи для пяти населенных пунктов с координатами: $A(1; 3)$, $B(1; -4)$, $C(5; 5)$, $D(6; -2)$, $E(7; 4)$.
- 4 В исходной задаче населенные пункты имеют координаты: $A(2; 4)$, $B(3; -5)$, $C(4; 7)$, $D(8; -4)$. Найдите положение железнодорожной станции, при котором она по возможности более равномерно удалена от всех четырех пунктов (разность между расстоянием до дальнего пункта и расстоянием до ближнего будет минимальной).

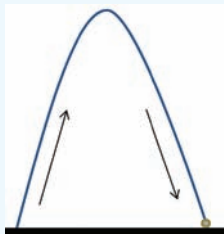
Указание. Добавить в расчетную таблицу графу «Разность» с формулой $\text{МАКС}() - \text{МИН}()$.

- 5 В исходной задаче населенные пункты имеют координаты: $A(1; -4)$, $B(2; 5)$, $C(3; -5)$, $D(10; -4)$. Найдите решение исходной задачи в случае, когда участок железной дороги не является прямолинейным, а задается графиком функции

$$y = \frac{2x(10 - x)}{25} - 1.$$

§ 24. Моделирование в задаче полета тела, брошенного под углом к горизонту

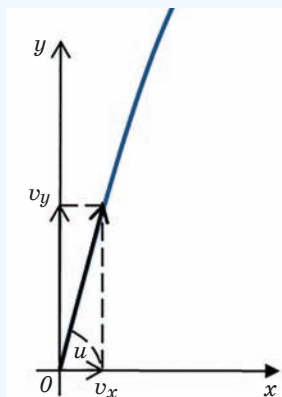
Пример 24.1. Нетрудно представить, что после броска камень полетит по гладкой траектории вида



Пример 24.2. Решение математической задачи возможно двумя путями. Его можно получить в виде математической формулы. Это аналитическое решение. Второй путь связан с построением компьютерной модели пошаговых вычислений. Это численное решение.

Будем строить компьютерную модель, рассчитывая для разных моментов времени координаты камня в пространстве. Для этого используем метод построения таблиц значений функции в электронных таблицах.

Пример 24.3. Прямоугольная система координат строится в вертикальной плоскости полета камня, и начало координат этой системы размещено в точке вылета камня.



24.1. Постановка задачи (этап 1)

Задача. Брошен камень с начальной скоростью 30 м/с под углом 60° к горизонту. Сопротивление воздуха не учитывать (пример 24.1). Вопросы:

1. Как далеко от места бросания камень упадет?
2. Сколько секунд камень будет находиться в полете?
3. Какова наибольшая высота взлета камня?
4. Как скоро от начала полета будет достигнута наивысшая точка полета?

24.2. Выбор плана создания модели (этап 2)

Для создания модели нужно составить математическую задачу (документальную математическую модель) и решить ее (пример 24.2). Таким образом, получаем план создания модели:

- 3а — создание документальной математической модели;
- 3б — создание компьютерной модели с помощью электронных таблиц.

24.3. Создание документальной математической модели (этап 3а)

В вертикальной плоскости полета камня зададим прямоугольную систему координат (пример 24.3).

Начальная скорость v (м/с) раскладывается на составляющие v_x и v_y по углу бросания u в градусах:

$$v_x = v \cdot \cos\left(\frac{u \cdot 3,14}{180}\right);$$

$$v_y = v \cdot \sin\left(\frac{u \cdot 3,14}{180}\right).$$

Положение тела в полете определяется парой координат $x(t)$, $y(t)$. Зависимость координат от времени t (с) описывается формулами

$$x(t) = v \cdot \cos\left(\frac{u \cdot 3,14}{180}\right) \cdot t;$$

$$y(t) = v \cdot \sin\left(\frac{u \cdot 3,14}{180}\right) \cdot t - \frac{9,81 \cdot t^2}{2},$$

где $g = 9,81$ — ускорение свободного падения. Положение камня в полете будем рассматривать в отдельные моменты времени (пример 24.4).

24.4. Создание компьютерной модели (этап 3б)

Исходные данные и начало расчетной таблицы разместим по схеме из примера 24.5. Ячейки первой строки расчетной таблицы заполняем нулями. Вторая строка содержит формулы: A10: =A9+\$A\$5

B10: =A10*\$A\$3*cos(\$A\$4*3,14/180)

C10: =A10*\$A\$3*sin(\$A\$4*3,14/180) – 9,81*A10^2/2

Следующие 39 строк расчетной таблицы, включая строку 49, заполняются вниз содержимым диапазона A10:C10.

Для наглядности построим траекторию полета камня как диаграмму графика функции (пример 24.6). На странице появится диаграмма с траекторией. Ее границы нужно расширить так, чтобы масштабы по осям стали примерно одинаковыми. Когда диаграмма выделена, к основным вкладкам с инструментами добавляются три новые для работы с диаграммой.

Оформим диаграмму с помощью инструментов вкладки **Макет** (пример 24.7).

Пример 24.4. Пусть начальный момент равен 0, а последующие моменты отстоят друг от друга на одну и ту же величину 0,2 с, называемую *шагом времени*.

Пример 24.5. Схема размещения данных и заголовков модели.

	A	B	C	D	E
1	Модель полета тела				
2	Исходные данные				
3	30 : начальная скорость (м/с)				
4	60 : угол бросания (градусы)				
5	0,2 : шаг времени (с)				
6					
7	Расчетная таблица				
8	Время	x(t)	y(t)		
9					

Пример 24.6. Для построения диаграммы выделяем диапазон B9:C49 (второй и третий столбцы расчетной таблицы) и на вкладке **Вставка** в группе **Диаграммы** выбираем диаграмму **Точечная**. В последних версиях электронных таблиц удобно выбрать диаграмму в списке всех диаграмм. Чтобы открыть список, надо щелкнуть по стрелке в правом нижнем углу группы и выбрать вкладку **Все диаграммы**.

Появляется панель с изображениями разновидностей диаграммы, на которой, пользуясь подсказками, выбираем диаграмму **Точечная с гладкими кривыми и маркерами**.

Пример 24.7. В группе **Подписи** с помощью кнопки **Легенда** удалим легенду с диаграммы. С помощью кнопки **Название диаграммы** добавим над диаграммой название «Траектория полета». С помощью кнопки **Названия осей** добавим для горизонтальной оси название «Дальность», а для вертикальной оси повернутое название «Высота».

Пример 24.8. Когда в столбце $y(t)$ найдены указанные соседние строки, моментом падения можно считать среднее арифметическое значений времени столбца «Время» расчетной таблицы в этих строках. Дальностью падения считается среднее арифметическое значений столбца $x(t)$ в этих же строках. Наибольшая высота взлета ищется как максимальное значение в столбце $y(t)$.

Можно заметить, что когда координата $y(t)$ становится отрицательной, модель является неадекватной (камень оказывается ниже уровня земли).



Упражнения

- 1 Повторите на компьютере рассмотренное в параграфе решение задачи полета тела, брошенного под углом к горизонту.
- 2 С помощью модели полета тела подбором найдите угол бросания, при котором камень с начальной скоростью 40 м/с упадет в 100 м от места бросания. Найдите время полета.
- 3 Подбором найдите начальную скорость, при которой камень, брошенный под углом 60° , упадет в 100 м от места бросания.
- 4 Подбором найдите начальную скорость, при которой камень, брошенный под углом 60° , сойдет неподвижную цель на удалении 100 м и на высоте 20 м.

Указание. Для обозначения цели на диаграмму нужно добавить маркер цели.

Диаграмму нужно выделить и на вкладке **Конструктор** в группе **Данные** щелкнуть по кнопке **Выбрать данные**. Появляется диалоговое окно **Выбор источника данных**, в котором в зоне **Элементы легенды (ряды)** следует щелкнуть по кнопке **Добавить**. Открывается диалоговое окно **Изменение ряда**, в котором вводится имя ряда «Цель 1», значение X, равное 100, и значение Y, равное 20. Кнопкой **ОК** закрывается одно окно, затем — второе. На диаграмме появляется маркер цели.

- 5 Подбором найдите угол бросания, при котором камень, имеющий начальную скорость 40 м/с, сойдет неподвижную цель на удалении 60 м и на высоте 30 м.
- 6 Подбором найдите начальную скорость с углом бросания 70° , при которой камень сойдет неподвижную цель на удалении 50 м и на высоте 60 м.
- 7* Подбором найдите начальную скорость и угол бросания, при которых камень сойдет две неподвижные цели: первую — на удалении 50 м и на высоте 30 м, вторую — на удалении 100 м и на высоте 5 м.

24.5. Исследование модели (этап 4)

Модель адекватна реальному процессу только с допущением об отсутствии сопротивления воздуха и для положительных значений координат.

24.6. Получение решения задачи (этап 5)

Чтобы ответить на вопросы задачи, анализируется расчетная таблица.

Чтобы ответить на первый вопрос, по числам в столбце $y(t)$ находятся две соседние строки, в которых стоят числа разных знаков. Ответы на остальные вопросы находятся в других столбцах этих строк (пример 24.8).

§ 25. Моделирование средствами текстового процессора

25.1. Моделирование бумажных документов

Мы привыкли, что текстовые редакторы используются для подготовки к печати бумажных документов, и как-то не задумываемся о том, что в текстовом редакторе создаются компьютерные модели будущих документов.

Электронный документ, создаваемый в текстовом редакторе, используется вместо создаваемого бумажного. Это образец документа, что является главным признаком модели.

В процессе компьютерного создания бумажного документа можно выделить все этапы моделирования (пример 25.1). Использование компьютерного моделирования при создании бумажных документов существенно уменьшило их трудоемкость. Исчезли пишущие машинки, на которых печатались документы, исчезла и профессия машинистки (пример 25.2).

В настоящее время многие электронные документы остаются моделями бумажных документов только формально, так как распечатки на бумаге не требуют.

25.2. Компьютерные носители вместо бумаги

Повсеместное распространение компьютеров привело к тому, что многие информационные модели также часто не достигают формы записи на бумаге, а остаются в форме файлов на компьютерных носителях.

Текстовые редакторы позволяют записывать на компьютерные носители в

Пример 25.1. Задача создания бумажного документа включает несколько этапов.

Этап 1

Вначале осуществляется постановка задачи — документ требует создания.

Этап 2

План создания бумажного документа традиционный: создать компьютерную модель документа, затем использовать ее для распечатки на бумаге.

Этап 3

Создается компьютерная модель текста.

Этап 4

Адекватность модели текстового документа — это прежде всего ее безошибочность, что обеспечивается встроенными средствами проверки.

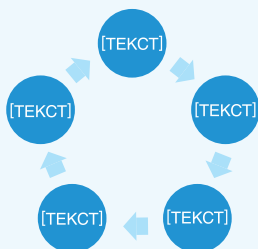
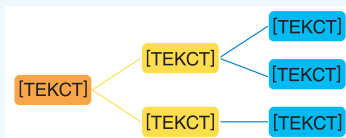
Этап 5

Решение задачи с помощью модели — распечатка компьютерной модели документа.

Пример 25.2. Печатанием документов занимались целые отделы, которые назывались машинописными бюро.



Пример 25.3. В текстовом редакторе удобно создавать разнообразные структурные модели с помощью средства SmartArt, которое размещено на вкладке **Вставка** в группе **Иллюстрации**. Вот некоторые шаблоны для создания структур.



Пример 25.4. На столе в ряд стоят бутылка минеральной воды, кружка, чашка, стакан и кувшин. В них находятся различные напитки: кофе, чай, молоко, квас и минеральная вода. Если стакан поставить между посудой с чаем и молоком, то по соседству с молоком будет квас, а кофе окажется точно посередине.

Для решения логических задач такого рода очень часто используются таблицы, в которых строки закреплены за одними объектами задачи, а столбцы — за другими объектами задачи, связанными с первыми.

В клетки вносят плюсы и минусы, отмечая связи между объектами.

бумагоподобной форме многие информационные модели. Это могут быть описания исторических событий (исторические модели), математические формулы (математические модели), таблицы (табличные модели), схемы и чертежи (графические модели) и др. (пример 25.3). Такие информационные модели также очень удобно хранить именно на компьютерных носителях.

25.3. Решение логических задач

Существует класс логических задач, для решения которых нужно создавать табличные модели (пример 25.4). Создание таблиц возможно как средствами электронных таблиц, так и средствами текстовых редакторов.

Рассмотрим метод решения таких логических задач с использованием средств текстового редактора.

25.4. Постановка задачи (этап 1)

Задача. В симфонический оркестр приняли на работу трех музыкантов: Брауна, Смита и Вессона, умеющих играть на скрипке, флейте, альте, кларнете, гобое и трубе. Известно, что:

- 1) Смит — самый высокий;
- 2) играющий на скрипке меньше ростом играющего на флейте;
- 3) играющие на скрипке, флейте и Браун любят пиццу;
- 4) когда между альтистом и трубачом возникает ссора, Смит мирит их;
- 5) Браун не умеет играть ни на трубе, ни на гобое.

На каких инструментах играет каждый из музыкантов, если каждый владеет двумя музыкальными инструментами?

25.5. Выбор плана создания модели (этап 2)

На этапе 3 моделирования будем строить компьютерную модель таблицы средствами текстового редактора.

25.6. Создание компьютерной модели (этап 3)

Для решения задачи в текстовом редакторе нужно построить таблицу с тремя строками (по числу музыкантов) и шестью столбцами (по числу музыкальных инструментов, которыми они владеют). В таблицу следует добавить одну строку и один столбец для описания объектов (пример 25.5). Таблицу необходимо также оформить.

25.7. Исследование модели (этап 4)

Проверка адекватности больших усилий не требует, т. к. таблица полностью соответствует условиям задачи.

25.8. Получение решения задачи (этап 5)

Будем заполнять клетки таблицы плюсами и минусами в соответствии с условиями задачи. Из условия 4) следует, что Смит не играет на альте и трубе (ставим два минуса в его строку). Из условия 3) следует, что Браун не играет на скрипке и флейте (ставим два минуса в его строку). Из условия 5) следует, что Браун не умеет играть на трубе и гобое (ставим еще два минуса в его строку). Получаем таблицу, представленную в примере 25.6.

Анализ таблицы показывает, что Браун умеет играть на альте и кларнете (ставим два плюса в его строку), а Вессон играет на трубе (ставим плюс в его строку).

Существуют также задачи, при решении которых строки таблицы закреплены за объектами задачи, а столбцы — за свойствами этих объектов.

В клетки такой таблицы на основании условий задачи вносят плюсы или минусы, которые показывают связи между объектами и их свойствами.

Для решения подобных задач достаточно построить и заполнить таблицу соответствующего размера.

В данной задаче рассматриваются только объекты: три музыканта и шесть музыкальных инструментов.

Текстовые редакторы позволяют строить таблицы практически любой размерности.

Пример 25.5. Табличная модель для решения задачи, построенная средствами текстового редактора.

	Скрипка	Флейта	Альт	Кларнет	Гобой	Труба
Браун						
Смит						
Вессон						

Адекватность подобных табличных моделей зависит также от правильности заполнения таблиц при решении задачи.

Пример 25.6.

	Скрипка	Флейта	Альт	Кларнет	Гобой	Труба
Браун	■	■			■	■
Смит			■			■
Вессон						

Пример 25.7.

	Скрипка	Флейта	Альт	Кларнет	Гобой	Труба
Браун	■	■	+	+	■	■
Смит		+	■			■
Вессон	+					+

Из условий 1) и 2) следует, что Вессон играет на скрипке, а Смит на флейте (ставим плюсы в эти строки). Получаем таблицу, представленную в примере 25.7. Анализ таблицы показывает, что Смит может играть еще только на гобое (ставим плюс). Задача решена.

Ответ: Браун играет на альте и кларнете; Смит играет на флейте и гобое; Вессон играет на скрипке и трубе.



Упражнения

- 1 Решите задачу. Беседуют трое друзей: Белокуров, Рыжов и Чернов. Черноволосый сказал Белокурову: «Любопытно, что один из нас белокурый, другой — черноволосый, третий — рыжий, но ни у кого цвет волос не соответствует фамилии». Определите цвет волос мальчиков.
- 2 Решите задачу. Три одноклассника — Влад, Тимур и Юрий — встретились спустя 10 лет после окончания школы. Выяснилось, что один из них стал врачом, другой — физиком, а третий — юристом. Один полюбил туризм, другой — бег, а страсть третьего — регби. Юра сказал, что на туризм у него не хватает времени, хотя его сестра — единственный врач в семье — заядлый турист. Врач сказал, что он разделяет увлечение коллеги. Забавно, но у двоих из друзей в названиях их профессий и увлечений не встречается ни одна буква их имен. Определите, кто чем любит заниматься в свободное время и у кого какая профессия.
- 3 Решите задачу. Однажды в «Артеке» за круглым столом оказалось пятеро ребят из Москвы, Санкт-Петербурга, Новгорода, Перми и Томска: Юра, Толя, Леша, Миша и Влад. Москвич сидел между томичем и Владом, петербуржец — между Юрой и Толей, а напротив сидели пермяк и Толя. Миша никогда не был в Санкт-Петербурге, а Юра не был в Москве и Томске. Томич с Толей регулярно переписываются. Определите, в каком городе живет каждый из ребят.
- 4 Решите задачу. Маша, Лена, Оля и Валя — замечательные девочки. Каждая из них играет на каком-нибудь музыкальном инструменте и говорит на одном из иностранных языков. Инструменты и языки у них разные. Маша играет на рояле. Девочка, которая говорит по-французски, играет на скрипке. Оля играет на виолончели. Маша не знает итальянского языка, а Оля не владеет английским. Лена не играет на арфе, а виолончелистка не говорит по-итальянски. Определите, на каком музыкальном инструменте играет каждая из девочек и каким иностранным языком она владеет.

§ 26. Знакомство с 3D-редактором

26.1. Понятие о 3D-графике

Графические информационные модели реальных и вымышленных объектов (рисунки, чертежи, фотографии) традиционно и долгое время создавались на бумаге. С появлением компьютеров такие модели стали строить с помощью графических редакторов. Со временем появились средства построения компьютерных моделей в форме объемных виртуальных объектов, которые на экране компьютера можно поворачивать вокруг любой оси и рассматривать со всех сторон (пример 26.1).

2D-графика (двумерная графика) — плоские компьютерные изображения и технологии их создания.

3D-графика (трехмерная графика) — виртуальные объемные объекты и технологии их создания.

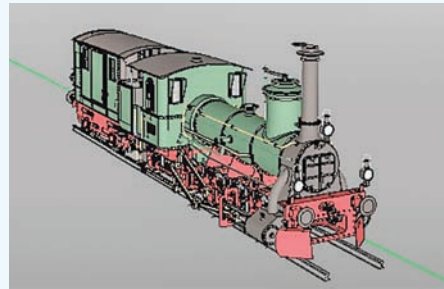
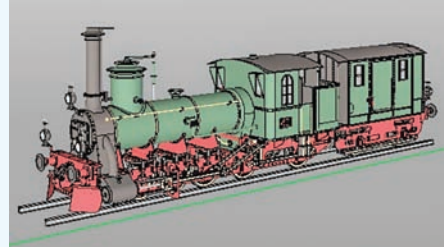
Буква *D* в названиях происходит от английского слова *dimensions* («измерения»).

Объекты 3D-графики называются *трехмерными моделями* или *3D-моделями*. По своей структуре они делятся на *поверхностные* и *твердотельные*.

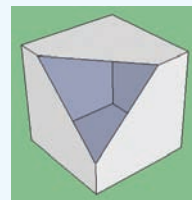
Поверхностная 3D-модель представляет внешнюю поверхность объекта оригинала в виде виртуальной пленки (пример 26.2).

Твердотельная 3D-модель представляет собой объект, выпиленный из однородного виртуального материала (пример 26.3).

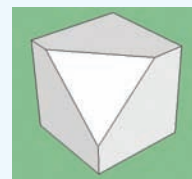
Пример 26.1. Изображения объемной компьютерной модели паровоза с разных сторон.



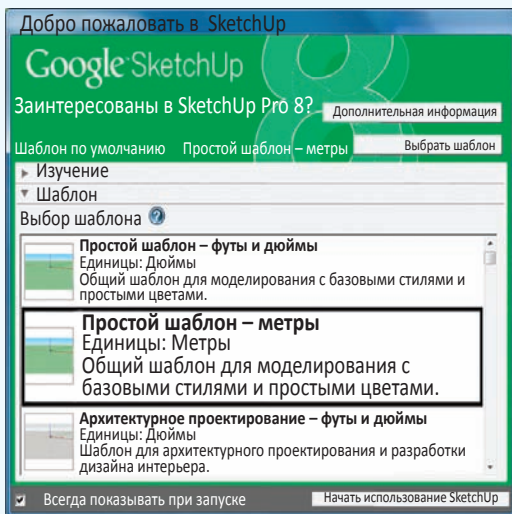
Пример 26.2. Поверхностная 3D-модель куба с косым срезом. В срезе видна поверхность куба изнутри.



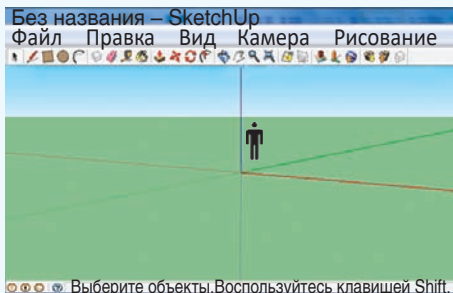
Пример 26.3. Твердотельная 3D-модель куба с косым срезом. На срезе виден виртуальный материал модели.



Пример 26.4. Диалоговое окно Добро пожаловать в SketchUp.

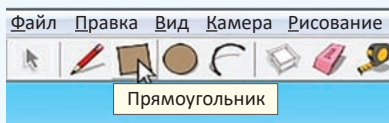


Пример 26.5. Начальное изображение области рисования в окне редактора.



Красная и зеленая оси определяют горизонтальную прозрачную плоскость. Синяя ось определяет вертикаль.

Пример 26.6. Всплывающая подсказка около указателя мыши.



3D-модели с успехом заменяют макеты и создаются с помощью специальных графических редакторов — 3D-редакторов. Мы будем использовать 3D-редактор Google SketchUp 8¹.

26.2. Запуск и выбор шаблона

После запуска редактора появится диалоговое окно **Добро пожаловать в SketchUp** (пример 26.4), в котором предлагается выбрать шаблон и единицу измерения размеров модели.

Мы будем моделировать здания, поэтому следует выбрать **Простой шаблон — метры** и щелкнуть по кнопке **Начать использование SketchUp**.

С интерфейсом основного окна редактора Google SketchUp 8 можно познакомиться в *Приложении 4* (с. 165).

В области рисования редактора на фоне удаленного горизонта изображена трехмерная система координат с цветными осями: красной, зеленой и синей. Около начала координат для ориентации и восприятия масштаба размещена фигура человека (пример 26.5).

Для управления редактором используются меню, кнопки на **Панели инструментов**, мышь и клавиатура.

В редакторе действует система всплывающих подсказок, которая позволяет познакомиться с названиями кнопок и данными создаваемой модели (пример 26.6).

26.3. Перемещение камеры

Считается, что в области рисования редактора существует виртуальная камера, которая передает изображение в окно редактора. Камеру переме-

¹ Доступен для скачивания на сайте <http://o-soft.ru/foto-grafika/3d/>

щают и рассматривают 3D-модели со всех сторон.

Режимы перемещения камеры можно включать кнопками пятого раздела **Панели инструментов** (пример 26.7).

Кнопка  **Орбита** включает режим **Орбита** вращения камеры вокруг центра области рисования.

Кнопка  **Панорама** включает режим **Панорама** перемещения камеры вертикально или горизонтально (пример 26.8).

Кнопка  **Масштаб** включает режим **Масштаб** приближения камеры к модели или удаления от нее. В этом режиме неподвижной остается точка в центре экрана (пример 26.9).

Кнопка  **В размер окна** щелчком мыши мгновенно устанавливает камеру так, что создаваемая модель полностью заполняет экран.

В любом режиме изменить положение камеры можно только мышью и клавиатурой.

Нажатие колесика мыши включает режим **Орбита**, а дополнительное нажатие клавиши **Shift** клавиатуры включает режим **Панорама**.

Приближает или удаляет камеру вращение колесика мыши. При этом неподвижной остается точка экрана, в которой находится указатель мыши (пример 26.10).

26.4. Простейшие объекты

Набор простейших объектов редактора (примитивов) невелик, и все они являются двумерными. Инструмент для рисования (вывода примитива на экран) выбирается на **Панели инструментов** (пример 26.11).

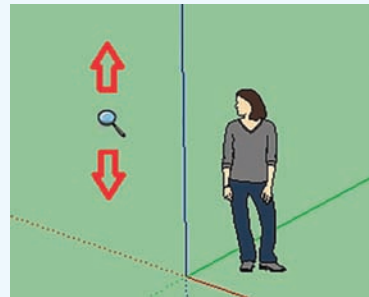
Пример 26.7. Кнопки включения режимов перемещения на **Панели инструментов**.



Пример 26.8. Чтобы переместить камеру в режимах **Орбита** или **Панорама**, надо нажать кнопку мыши с указателем в любой точке области рисования и перетащить указатель.

В режиме **Орбита** удержание клавиши **Shift** клавиатуры включает режим **Панорама**.

Пример 26.9. Чтобы в режиме **Масштаб** приблизить или удалить камеру, нужно нажать кнопку мыши с указателем в любой точке области рисования и перетащить указатель мыши вверх или вниз.



Пример 26.10. Чтобы в области рисования приблизиться к объекту или удалиться от него, нужно перенести указатель мыши на объект и покрутить колесико мыши.

Используя режимы перемещения камеры, можно переместить камеру в любую точку области рисования и рассмотреть создаваемую модель с любой стороны и в любом масштабе.

Пример 26.11. Кнопки инструментов для рисования.

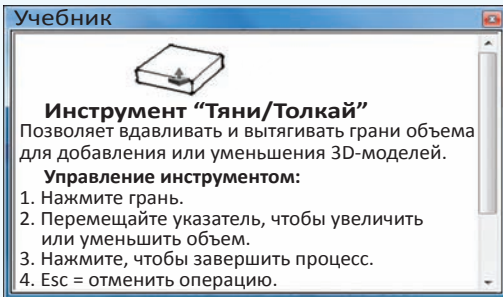


Пример 26.12. Чтобы вывести прямоугольник, нужно щелчком мыши отметить один угол прямоугольника, отвести указатель мыши в противоположный угол и щелкнуть второй раз.

Чтобы вывести окружность, нужно отметить щелчками ее центр и точку на окружности (построить ее радиус).

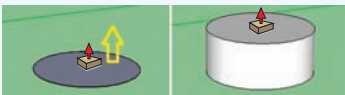
Инструмент **Линия** щелчками и перемещениями мыши рисует непрерывную ломаную линию. Чтобы замкнуть ломаную, нужно щелкнуть по начальной точке. Если нужна незамкнутая ломаная, то построение завершается клавишей **Esc** клавиатуры.

Пример 26.13. Окно **Учебник** вызывают командой меню **Окно** → **Учебник**.



Пример 26.14. Кнопки преобразования объектов:

Пример 26.15. После выбора инструмента **Тяни/Толкай** указатель мыши помещают на плоскую грань 3D-объекта или примитива. Грань выделяется, по ней делают щелчок и перемещают указатель. Грань перемещается вслед за указателем мыши.



Объем объекта увеличивается или уменьшается. Преобразование завершает второй щелчок. Нажатие клавиши **Ctrl** клавиатуры включает/выключает режим наращивания, при котором на месте сдвигаемой грани остается ее копия.

Результат рисования инструментами **Линия**, **Прямоугольник**, **Окружность** и **Дуга** легко предсказать по опыту работы с графическим редактором **Paint**. Вот только в редакторе **Google SketchUp** привычная протяжка мыши при выводе *не используется*.

Вывод примитива начинается щелчком, указатель мыши переносится, а завершение вывода осуществляется вторым щелчком (пример 26.12).

После вывода прямоугольника, окружности и замкнутой ломаной становится серой часть плоскости, которую они ограничивают.

Плоские области, которые ограничены линиями, в редакторе называются **гранями**.

К каждому используемому инструменту редактор **Google SketchUp** может выводить развернутые подсказки, которые открываются в окне **Учебник** (пример 26.13).

При выводе не бойтесь экспериментировать. Любое действие в редакторе можно отменить клавиатурной комбинацией **Ctrl + z**.

26.5. Преобразования объектов

За преобразование объектов в редакторе отвечают кнопки четвертого раздела **Панели инструментов** (пример 26.14).

Инструмент **Тяни/Толкай** позволяет вытягивать и вдавливать грани объектов для добавления или уменьшения объема 3D-моделей (пример 26.15).

Инструмент  **Переместить** позволяет перемещать, растягивать и копировать объекты в области рисования. Чтобы растянуть объект, на нем щелчком выделяют грань, линию или точку. Затем указатель мыши в виде  перемещают — объект деформируется. Преобразование завершается вторым щелчком мыши.

Инструмент  **Смещение** позволяет на любой грани создавать уменьшенную или увеличенную копию ее границы. В этом режиме, кроме указателя мыши в виде , на границе грани появляется красная точка (пример 26.16).

26.6. Текстуры и материалы

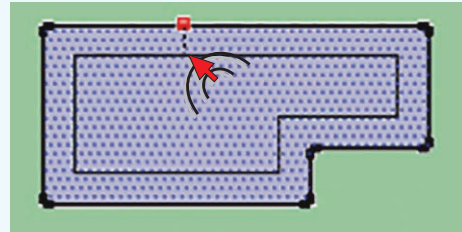
Любой графический редактор позволяет заливать (раскрашивать) замкнутые области цветом или узором. В 3D-редакторах, кроме цвета, можно использовать материалы или текстуры.

В 3D-моделировании **текстура (материал)** — изображение реального материала, которое можно наносить на поверхность 3D-модели как узор.

Некоторые текстуры меняют не только внешний вид поверхности модели, но и ее свойства (пример 26.17).

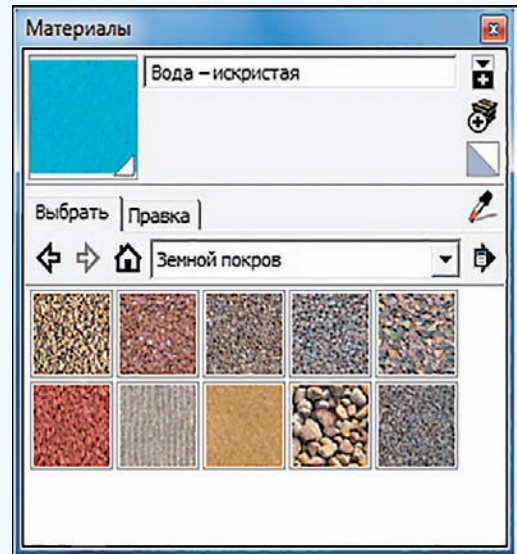
В Google SketchUp используется термин *материалы*. Для работы с цветом и материалами предназначен узнаваемый инструмент  **Заливка**, который размещен в третьем разделе **Панели инструментов**. Приемы работы с этим инструментом также хорошо узнаваемы (пример 26.18).

Пример 26.16. Копия границы грани.



Пример 26.17. Светопроницаемая текстура придает любой поверхности модели свойства стекла, включая прозрачность.

Пример 26.18. Инструмент  **Заливка** выбирается на **Панели инструментов**. Появляется диалоговое окно **Материалы**.

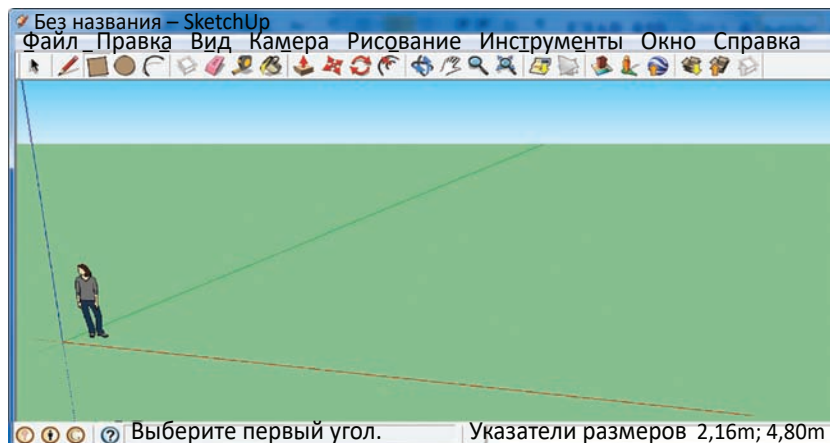


В списке щелчком выбирается тип материала, конкретный материал, а затем делается щелчок по поверхности модели.



Упражнения

- 1 Запустите 3D-редактор Google SketchUp. Если появляется диалоговое окно **Добро пожаловать в SketchUp**, выберите **Простой шаблон — метры** и перейдите в основное окно программы.
- 2 В 3D-редакторе установите указатель мыши на фигуру человека в области рисования. Отработайте приближение и удаление камеры колесиком мыши.
- 3 В 3D-редакторе выполните указанные действия:
 1. Щелкните по кнопке  **В размер окна** и отведите камеру от фигуры человека колесиком мыши.
 2. Отработайте включение/выключение режима **Орбита** колесиком мыши.
 3. Включите режим **Орбита**, удерживая колесико мыши, и перетащите указатель по окну, чтобы переместить камеру.
 4. Отработайте операцию перемещения камеры вокруг центра окна.
 5. В режиме **Орбита** сдвиньте указатель мыши вверх настолько, чтобы камера опустилась ниже горизонтальной плоскости, затем поднимите камеру выше горизонтальной плоскости.
- 4 В 3D-редакторе выполните указанные действия.
 1. Щелкните по кнопке  **В размер окна** и отведите камеру от фигуры человека колесиком мыши.
 2. Отработайте включение режима **Панорама** одновременным удерживанием колесика мыши и клавиши **Shift** клавиатуры.
 3. В режиме **Панорама** перетащите указатель мыши по окну, чтобы переместить камеру вверх, вниз, вправо или влево.
- 5 Переведите окно 3D-редактора в следующее состояние, которое будем использовать как исходное для построений в следующих упражнениях.



- 6 В 3D-редакторе выполните указанные действия.
1. Выведите на горизонтальную плоскость прямоугольник, круг, замкнутую ломаную линию, незамкнутую ломаную линию.
 2. Медленно переместите указатель мыши над гранями и границами выведенных примитивов и проследите за подсказками около указателя мыши.
 3. Очистите окно 3D-редактора от построений упражнения 6 многократным использованием клавиатурной комбинации **Ctrl + z**.
- 7 В 3D-редакторе, используя инструмент  **Тяни/Толкай**, постройте прямоугольный параллелепипед, для чего выполните указанные действия.
1. Выведите на горизонтальную плоскость прямоугольник.
 2. Установите указатель мыши на грани прямоугольника и сделайте щелчок, затем переместите указатель мыши вверх и сделайте второй щелчок (указатель мыши вытягивает грань, и образуется объемная фигура).
- 8 По аналогии с упражнением 7 постройте цилиндр, вытягивая вверх грань круга.
- 9 Познакомьтесь с режимом наращивания инструмента  **Тяни/Толкай**, выполняя указанные действия.
1. Постройте прямоугольный параллелепипед.
 2. Нажмите клавишу **Ctrl** клавиатуры несколько раз (около указателя появляется и пропадает знак «плюс», который указывает на включение режима наращивания).
 3. В режиме наращивания щелкните по верхней грани параллелепипеда, сдвиньте указатель мыши вверх и щелкните второй раз (указатель мыши вытянет грань, грань на исходном месте тоже останется, а режим наращивания выключится автоматически сдвигом указателя мыши).
 4. В режиме наращивания вытяните верхнюю грань объемной фигуры еще раз.
 5. Медленно (без нажатий) переместите указатель мыши над гранями фигуры и проследите за изменением их цвета, которое показывает привязку инструмента к грани.
 6. На верхней части объемной фигуры вытяните 4 боковые грани, включая 4 раза режим наращивания, и вдавите верхнюю грань.
- 10 Используйте результаты упражнения 9 и познакомьтесь с инструментом  **Переместить**, выполняя указанные действия.
1. Установите указатель мыши на любой грани объемной фигуры, сделайте щелчок, переместите указатель и сделайте второй щелчок (фигура деформируется).
 2. Переместите любую линейную часть границы любой грани объемной фигуры.
 3. Переместите любую угловую точку объемной фигуры.

11 Используйте результаты упражнения 10 и познакомьтесь с инструментом



Смещение, выполняя указанные действия.

1. Установите указатель мыши на любой грани объемной фигуры, сделайте щелчок.
2. Перемещайте указатель мыши, пока на грани не появится уменьшенная копия границы этой грани.
3. Завершите операцию вторым щелчком мыши.

12 Используйте результаты упражнения 11 и познакомьтесь с инструментом

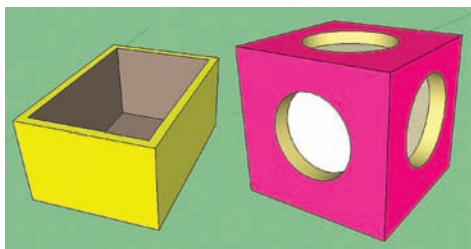


Заливка, выполняя указанные действия.

1. В диалоговом окне **Материалы** самостоятельно выберите тип материала, конкретный материал и залейте любую грань или поверхность на объемной фигуре.
2. Залейте другие грани и поверхности на объемных фигурах материалами и цветами по своему выбору. Обязательно используйте материалы типа «Светопроницаемые».

13 Результаты упражнения 12 сохраните в файле командой меню **Файл** → **Сохранить**. 3D-редактор SketchUp сохраняет модели в файлах с расширением **skp**.

14 В 3D-редакторе SketchUp создайте и сохраните в файлах следующие 3D-модели. Новые проекты создаются командой меню **Файл** → **Создать**.



§ 27. Моделирование в задачах проектирования строений

Создание объемных 3D-моделей строящихся и проектируемых строений и сооружений — одно из основных предназначений изучаемого нами 3D-редактора Google SketchUp. Его возможности позволяют создавать модели очень сложных строений и сооружений.

27.1. Постановка задачи (этап 1)

Задача. Для представления заказчику проекта дачного домика требуется создать компьютерный объемный макет будущего строения (пример 27.1).

Строение имеет внешние размеры 6×6 м, высота стен составляет 3 м. Вокруг строения находится отмостка шириной 1 м и толщиной 0,1 м.

Двускатная крыша имеет высоту конька 5,7 м от земли, свес кровли — 0,5 м, подшивка свеса крыши имеет толщину 0,1 м.

Особенности устройства крыльца, окон, входной двери и материалы будут подобраны в процессе создания модели.

27.2. Выбор плана создания модели (этап 2)

Для создания компьютерного объемного макета будущего строения будем использовать 3D-редактор Google SketchUp.

План создания модели включает только этап 3 — создание компьютерной модели с помощью 3D-редактора.

27.3. Создание компьютерной модели (этап 3б)

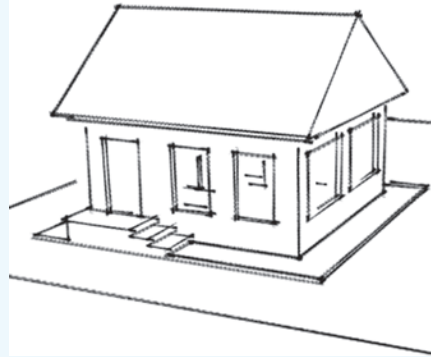
Запускаем 3D-редактор Google SketchUp и изобразим сначала земельный участок в форме квадрата 20×20 м (пример 27.2).

Строим основание дома (6×6 м) и отмостку (шириной 1 м) одновременно. Следовательно, по такой же схеме выводим примерно на середину участка еще один квадрат размерами 8×8 м.

Основание строения поднимаем на толщину отмостки 0,1 м. Для этого выбираем инструмент  **Тяни/Толкай**, щелкаем по квадрату основания и перемещаем указатель вверх. Затем вводим с клавиатуры толщину отмостки «0,1» и нажимаем клавишу **Enter** клавиатуры (пример 27.3).

Далее поднимаем стены. Для этого щелкаем по квадрату основания

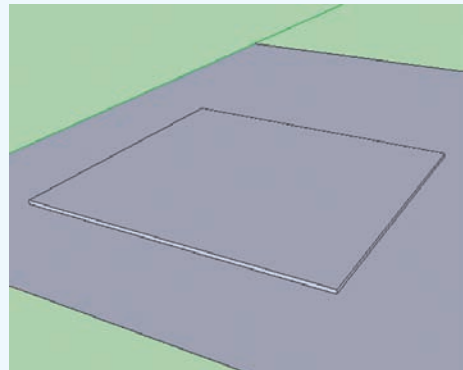
Пример 27.1. Карандашный эскиз строения, выполненный художником-дизайнером.

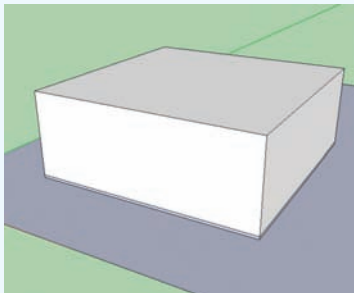
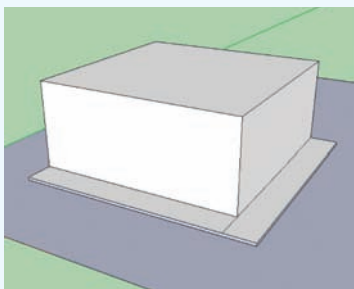
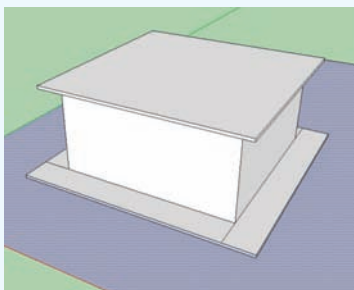


Домик будет размещаться на участке размерами 20×20 м.

Пример 27.2. Выбираем инструмент **Прямоугольник**, щелчком устанавливаем угол прямоугольника в начало координат области рисования и сдвигаем указатель мыши вправо. Затем вводим с клавиатуры размеры «20;20» и нажимаем клавишу **Enter** клавиатуры.

Пример 27.3. Основание поднято на высоту отмостки.



Пример 27.4. Построены стены.**Пример 27.5.** Две стены сдвинуты внутрь дома.**Пример 27.6.** Слой подшивки выдвинут наружу на 0,5 м.

3-D-редактор Google SketchUp позволяет строить красочные и естественные на вид объекты, которые можно рассмотреть со всех сторон и во всех подробностях. Любители создают библиотеки таких 3D-моделей.

и нажимаем клавишу **Ctrl** клавиатуры, чтобы отделить слой отмостки. Около указателя мыши появляется знак «плюс» режима наращивания.

Перемещаем указатель вверх. Затем вводим с клавиатуры высоту стен «3» и нажимаем клавишу **Enter** клавиатуры. Опять нажимаем клавишу **Ctrl** клавиатуры, чтобы выключить режим наращивания (пример 27.4).

Сдвигаем стены внутрь дома на 1 м и таким способом отделяем отмостку от стен. Для этого щелкаем по одной стене, сдвигаем указатель мыши внутрь дома. Затем вводим с клавиатуры ширину отмостки «1» и нажимаем клавишу **Enter** клавиатуры. Последовательно сдвигаем внутрь дома все стены (пример 27.5).

Надстраиваем на доме технологический слой подшивки свеса крыши толщиной 0,1 м.

Устанавливаем указатель мыши на верхний квадрат дома. Щелкаем мышью и нажимаем клавишу **Ctrl** клавиатуры, чтобы включить режим наращивания. Перемещаем указатель вверх. Затем вводим с клавиатуры толщину подшивки «0,1», нажимаем клавишу **Enter**. Опять нажимаем клавишу **Ctrl** клавиатуры для выключения режима наращивания. Уже известным приемом выдвигаем слой подшивки по горизонтали наружу на величину свеса 0,5 м (поочередно, во все стороны, пример 27.6).

Строим крышу. Для этого выбираем инструмент  **Линия** и проводим

линию посередине верхнего квадрата параллельно красной оси.

Выбираем инструмент **Переместить**, щелчком устанавливаем его на новую линию и перемещаем указатель вверх. Затем вводим с клавиатуры рассчитанную высоту крыши «2,5» и нажимаем клавишу **Enter** (пример 27.7).

На передней стене у левого угла установим крыльцо основанием 1×3 м и высотой 0,5 м (пример 27.8). Для этого от угла здания выкладываем на отмостку прямоугольник размером 1×3 м, вытягиваем его вверх на 0,5 м. Инструментом **Линия** на верхней грани линиями отделяем две ступени, которые потом аккуратно вдавливаем вниз.

Прямоугольниками отмечаем на передней стене дверь на крыльце и правее два окна по своему вкусу. Прямоугольники двери и окон вдавливаем в здание на 0,15 м (пример 27.9). На остальных стенах устанавливаем окна по своему вкусу.

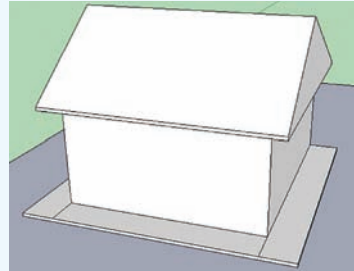
Инструментом **Заливка** придаем всем граням вид материалов (пример 27.10). Не забываем про светопроницаемые материалы для окон.

27.4. Исследование модели и получение решения (этапы 4 и 5)

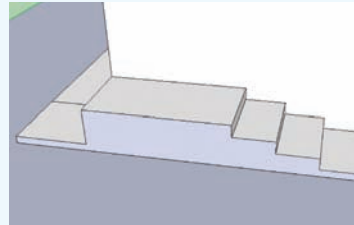
Адекватность модели проектируемому строению определяется соблюдением размеров и применяемых материалов.

Решением задачи является демонстрация модели заказчику.

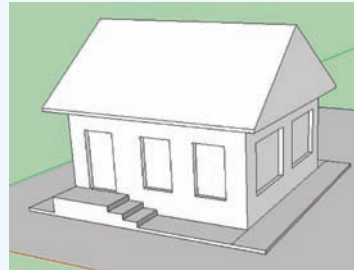
Пример 27.7. Крыша, построенная инструментом  **Переместить**.



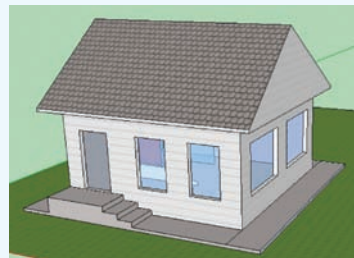
Пример 27.8. Крыльцо с тремя ступенями.



Пример 27.9. Дверь и окна.



Пример 27.10. Материалы.



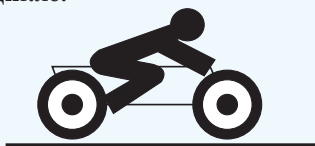


Упражнения

- 1 В модель домика добавить дверной косяк и оконные рамы.
- 2 На участке около домика расположить дорожки, площадки и клумбы.
- 3 Вокруг участка построить забор.

§ 28. Моделирование движения мотоцикла

Пример 28.1. В графическое окно системы PascalABC.NET выводится движущаяся фигура мотоциклиста на мотоцикле:



Пример 28.2. Кинограмма прыжка по заданию 1.



Пример 28.3. Кинограмма маневра по заданию 2.



Пример 28.4. Кинограмма маневра по заданию 3.



Компьютерная модель реально-го процесса всегда каким-то образом имитирует течение времени процесса. Значение времени в модели не всегда совпадает с реальным временем исполнения программы и во многом зависит от быстродействия компьютера.

Это расхождение и требует настрой-ки в первую очередь.

28.1. Постановка задачи (этап 1)

Задача. Исследовать движение мотоцикла, управляемого мотоциклистом, на готовой компьютерной модели в среде PascalABC.NET (пример 28.1). Управляющие параметры модели: величина тяги двигателя, момент отключения тяги и номер испытательного трека, который соответствует номеру задания к задаче. Тяга двигателя мотоцикла в модели включается автоматически в начальный момент движения.

1. На треке 1 с трамплином, имеющим уклон 15° , с точностью до сотых найти минимальную величину тяги, при которой мотоцикл после прыжка с трамплина приземляется на оба колеса (пример 28.2).

2. На треке 2 с трамплином, имеющим уклон 20° , с точностью до сотых найти минимальную величину тяги, при которой мотоцикл проедет трамплин без аварии (пример 28.3).

3. На ровном треке 3 найти величину тяги и момент ее отключения, при которых мотоцикл безаварийно «встает на дыбы», а затем продолжает движение в обычном положении (пример 28.4).

28.2. Выбор плана создания модели (этап 2)

Поскольку компьютерная модель уже создана, то ее следует изучить, а

затем запустить и настроить на конкретном компьютере. Получаем следующие этапы плана:

- 3а — изучение готовой компьютерной модели;
- 3б — настройка готовой компьютерной модели.

28.3. Изучение готовой компьютерной модели (этап 3а)

Файл Program28.pas после скачивания с Национального образовательного портала загружается в систему программирования PascalABC.NET.

Моделирующая программа использует математическую модель, которая представляет мотоциклиста и мотоцикл как единую систему нескольких материальных точек, соединенных упругими и вязкими связями.

Для конкретных расчетов по формулам математической модели построена вычислительная модель, которая записана на языке программирования PascalABC.NET (пример 28.5).

В первых строках текста программы задаются значения нескольких констант, которые описаны в строках-комментариях.

28.4. Настройка готовой компьютерной модели (этап 3б)

В начале настройки программу запускают со следующими значениями управляющих параметров:

- номер испытательного трека — 3;
- величина тяги двигателя — 3;
- момент отключения тяги — 50.

Программа завершает работу, и в графическом окне появляются значения модельного времени и фактического времени проезда мотоцикла по графическому экрану.

Моделирующая программа основана на сложной математической модели и весьма реально отображает динамичный процесс гонок на мотоцикле.

Модель имеет пять управляемых параметров-констант. Три первые константы являются управляющими параметрами модели (номер испытательного трека, величина тяги двигателя и момент отключения тяги двигателя). Остальные параметры используются для настройки модели (величина шага модельного времени и кратность вывода графики).

Для номера испытательного трека возможные значения только 1, 2 или 3. Величина тяги двигателя может принимать любые численные значения от 0,5 до 7,0. Момент отключения тяги двигателя может быть любым положительным числом. Кратность вывода графики — любое целое число от 1 до 20. Величина шага модельного времени не превосходит 0,01.

Пример 28.5. В вычислительной модели после запуска устанавливается нулевое значение для переменного времени моделирования, вычисляются начальные координаты материальных точек системы и в графическое окно выводится начальное изображение трека и мотоцикла.

Затем значение переменной времени моделирования увеличивается на шаг времени, вычисляются новые координаты точек системы, в графическом окне стирается предыдущее изображение и по новым координатам строится новое изображение трека и мотоцикла.

Этот процесс повторяется, пока мотоцикл не достигнет края трека или пока не произойдет остановка или авария.

Следует знать, что на графический экран выводятся изображения не всех положений мотоцикла, а только одного из заданного числа положений (кратность вывода графики).

Пример 28.6. Если модельное время меньше фактического, то для их выравнивания следует слегка увеличить величину шага модельного времени. Это первый параметр настройки. Следует знать, что в любом случае величина шага времени не должна превышать 0.01.

Если в результате увеличения шаг модельного времени достиг предела в 0.01, а значения времени еще не сравнялись, нужно постепенно увеличивать другой параметр настройки — кратность вывода графики.

Если модельное время больше фактического, то для выравнивания следует уменьшить величину шага модельного времени.

Пример 28.7. Программу запускают на исполнение. После наблюдения за поездкой мотоцикла значение одного параметра меняют, и программу запускают снова. Так, в задании 1 следует постепенно увеличивать величину тяги мотоцикла, чтобы получить требуемый вид прыжка. Аналогично поступают и в остальных случаях.

Программу нужно настроить так, чтобы модельное время не более чем в два раза отличалось от фактического. Для этого модель запускают с разными значениями параметров настройки и анализируют полученные показания времени (пример 28.6).

28.5. Исследование модели (этап 4)

Адекватность готовой компьютерной модели проверена разработчиком в различных режимах и со значениями параметров в рекомендованных пределах.

28.6. Получение решения задачи (этап 5)

Для получения ответа на задачу в текст программы вводят значения управляющих параметров. В заданиях 1—2 это номер трека (число 1, 2 или 3), величина тяги двигателя (число от 0.1 до 7.1), а в задании 3 еще и время отключения тяги (положительное число).

Искомые значения требуемых параметров находят подбором по результатам компьютерных испытаний (пример 28.7).



Упражнения

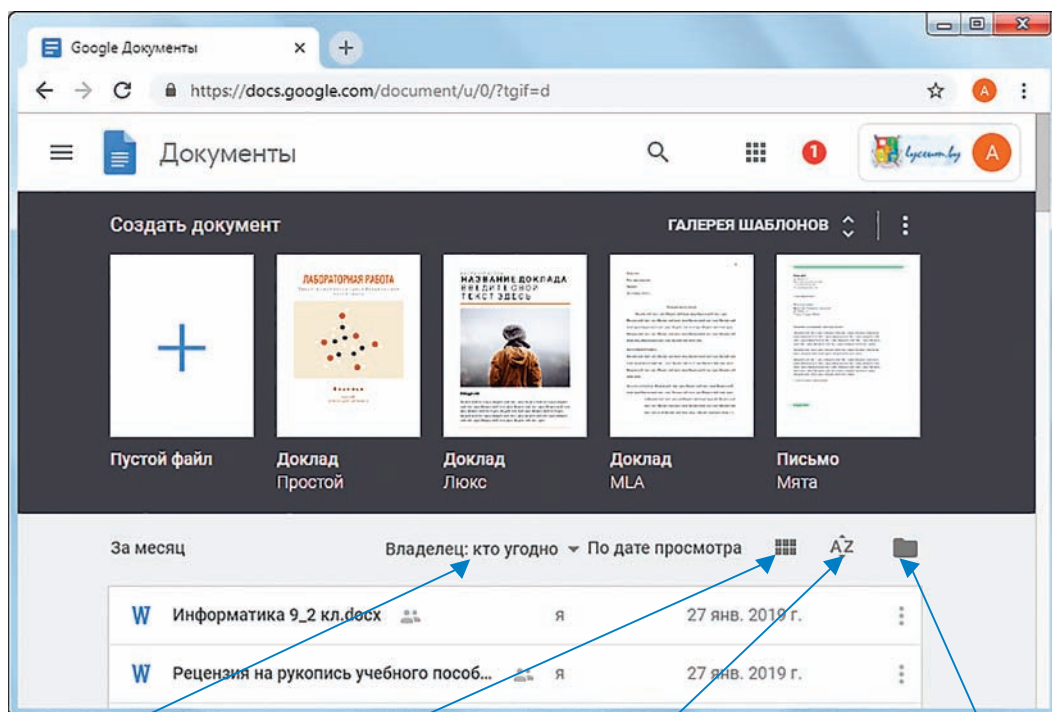
- 1 Опишите математическую модель, которая лежит в основе компьютерной модели движения мотоцикла.
- 2 Опишите принцип работы вычислительной модели движения мотоцикла.
- 3 Назовите управляющие параметры компьютерной модели движения мотоцикла.
- 4 Настройте готовую компьютерную модель в соответствии с пунктом 28.4 параграфа.
- 5 Для выполнения каждого из трех заданий проведите этап 5 моделирования, задавая номер трека и подбирая значения других параметров по результатам многократных запусков модели.

ПРИЛОЖЕНИЯ

Приложение 1

Интерфейс приложения Google Docs Документы

Стартовая страница:



Выбор владельца
документа

✓ Владелец: кто угодно
Владелец: я
Владелец: не я

Вид списка
документов



В виде сетки



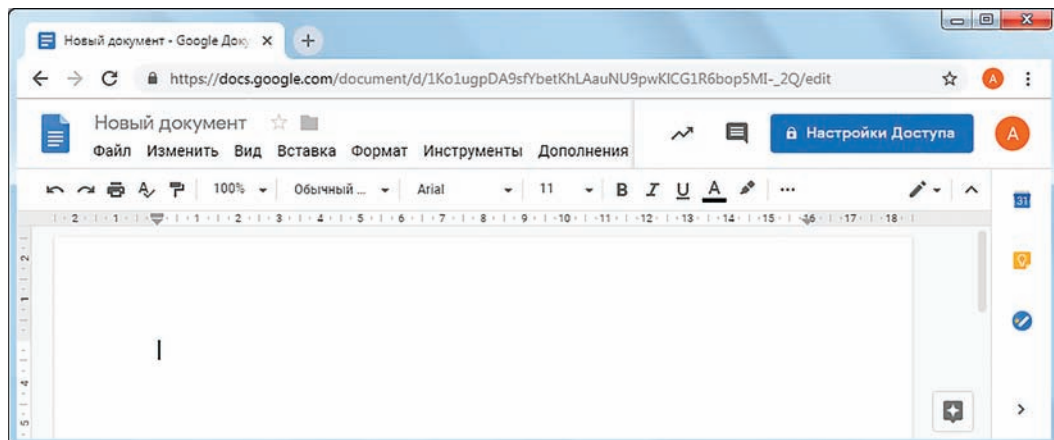
В виде списка

Сортировка
документов

✓ По дате просмотра
По дате моих изменений
По дате изменений
Название

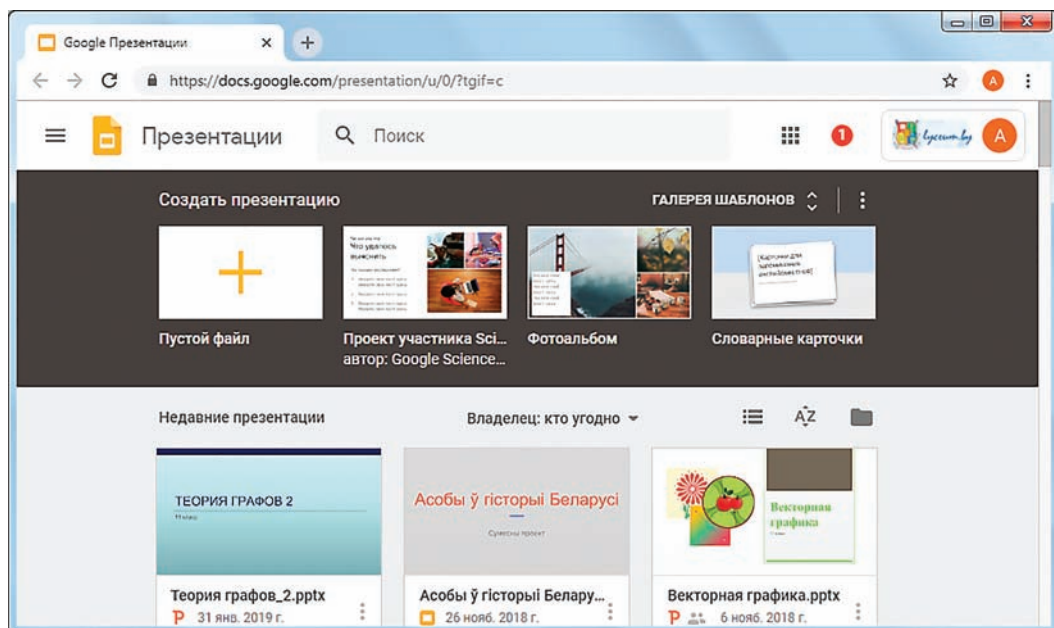
Открыть окно
выбора файла

Документы можно создавать, используя предложенные шаблоны. После выбора (например, **Пустой файл**) открывается окно текстового редактора:

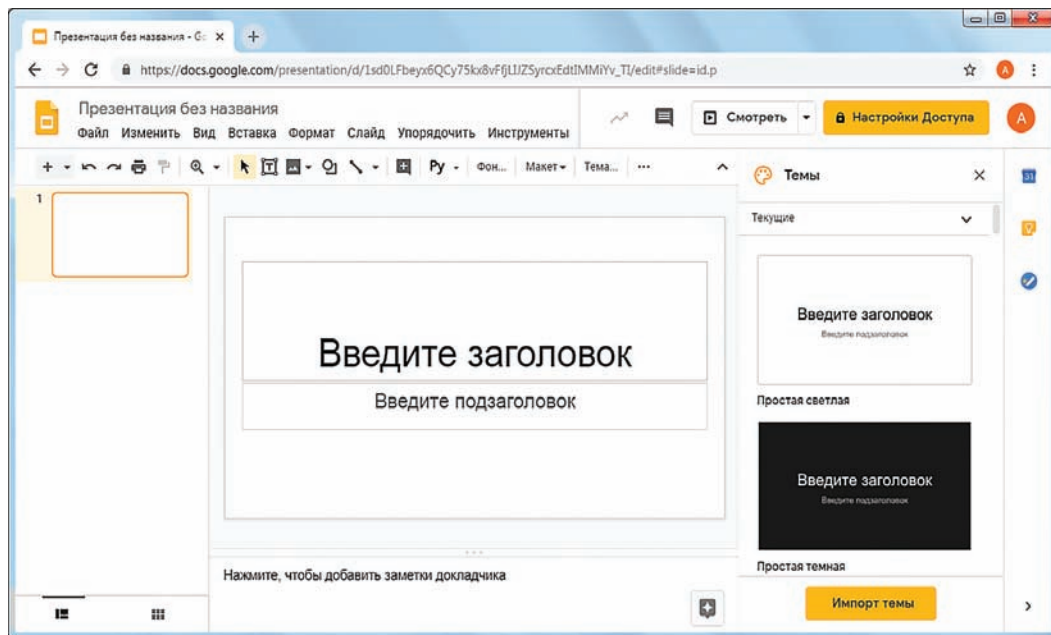


Интерфейс приложения Google Docs Презентации

Стартовая страница:



Окно редактора презентаций:



Совместная работа над документом

1. Выбрать **Настройки доступа**.
2. Сохранить документ в Google Drive.

Требуется название

×

Перед тем как предоставить доступ к документу, дайте ему название:

Пример|

Пропустить

Сохранить

3. Ввести имена или адреса электронной почты людей, которым открывается совместный доступ.

Совместный доступ Копировать ссылку общего доступа

Люди

Введите имена или адреса эл. почты...

Готово

Расширенные

4. Для каждого выбрать вид разрешенной деятельности.

Совместный доступ Копировать ссылку общего доступа

Люди

Иван Петрович Иванов x Добавьте пользователей...

Добавьте описание

✓ Редактирование
Комментирование
Просмотр

Оповестить пользователей ☒

Отправить Отмена

Расширенные

Приложение 2

Некоторые графические примитивы

Запись на Pascal	Описание
<code>SetPixel(x, y, c);</code>	Закрашивает пиксел с координатами (x, y) цветом c .
<code>MoveTo(x, y);</code>	Устанавливает текущую позицию рисования в точку (x, y) .
<code>LineTo(x, y);</code>	Рисует отрезок от текущей позиции до точки (x, y) . Текущая позиция переносится в точку (x, y) .
<code>Line(x1, y1, x2, y2);</code>	Рисует отрезок от точки $(x1, y1)$ до точки $(x2, y2)$.
<code>Circle(x, y, r);</code>	Рисует заполненный круг с центром (x, y) и радиусом r .
<code>Ellipse(x1, y1, x2, y2);</code>	Рисует заполненный эллипс, ограниченный прямоугольником, заданным координатами противоположных вершин $(x1, y1)$ и $(x2, y2)$.
<code>Rectangle(x1, y1, x2, y2);</code>	Рисует заполненный прямоугольник, заданный координатами противоположных вершин $(x1, y1)$ и $(x2, y2)$.
<code>RoundRect(x1, y1, x2, y2, w, h);</code>	Рисует заполненный прямоугольник со скругленными краями; $(x1, y1)$ и $(x2, y2)$ задают пару противоположных вершин, а w и h — ширину и высоту эллипса, используемого для скругления краев.
<code>Arc(x, y, r, a1, a2);</code>	Рисует дугу окружности с центром в точке (x, y) и радиусом r , заключенную между двумя лучами, образующими углы $a1$ и $a2$ с осью OX ($a1$ и $a2$ — вещественные, задаются в градусах и отсчитываются против часовой стрелки).

Продолжение таблицы

Запись на Pascal	Описание
<code>Pie(x, y, r, a1, a2);</code>	Рисует заполненный сектор круга с центром в точке (x, y) и радиусом r , заключенный между двумя лучами, образующими углы $a1$ и $a2$ с осью OX ($a1$ и $a2$ — вещественные, задаются в градусах и отсчитываются против часовой стрелки).
<code>FloodFill(x, y, c);</code>	Заливает область одного цвета цветом c , начиная с точки (x, y) .

Команды модуля GraphABC для работы с текстом

Название процедуры	Действие
<code>SetFontSize(s);</code>	Устанавливает размер текущего шрифта в пунктах.
<code>SetFontName(name);</code>	Устанавливает имя текущего шрифта, например: <code>SetFontName('Times New Roman');</code>
<code>SetFontColor(c);</code>	Устанавливает цвет текущего шрифта. Значения цвета: cl + английское название, например $clRed$, $clPink$.
<code>SetFontStyle(fs);</code>	Устанавливает стиль текущего шрифта. Значения: $fsNormal$ — обычный; $fsBold$ — жирный; $fsItalic$ — наклонный; $fsBoldItalic$ — жирный наклонный; $fsUnderline$ — подчеркнутый; $fsBoldUnderline$ — жирный подчеркнутый; $fsItalicUnderline$ — наклонный подчеркнутый; $fsBoldItalicUnderline$ — жирный наклонный, подчеркнутый.
<code>TextOut(x, y, z)</code>	Выводит строку или число z в прямоугольник с координатами левого верхнего угла (x, y) .

Функции для работы со строками

Название подпрограммы	Описание
<code>Copy(s, index, count)</code>	Возвращает подстроку строки <code>s</code> длины <code>count</code> с позиции <code>index</code> .
<code>FloatToStr(a)</code>	Преобразует вещественное число <code>a</code> к строковому представлению.
<code>IntToStr(a)</code>	Преобразует целое число <code>a</code> к строковому представлению.
<code>LastPos(subs, s)</code>	Возвращает позицию последнего вхождения подстроки <code>subs</code> в строке <code>s</code> . Если не найдена, возвращает 0.
<code>LastPos(subs, s, from)</code>	Возвращает позицию последнего вхождения подстроки <code>subs</code> в строке <code>s</code> , начиная с позиции <code>from</code> . Если не найдена, возвращает 0.
<code>LeftStr(s, count)</code>	Возвращает первые <code>count</code> символов строки <code>s</code> .
<code>Length(s: string)</code>	Возвращает длину строки.
<code>LowerCase(s)</code>	Возвращает строку в нижнем регистре.
<code>Pos(subs, s)</code>	Возвращает позицию подстроки <code>subs</code> в строке <code>s</code> . Если не найдена, возвращает 0.
<code>PosEx(subs, s, from)</code>	Возвращает позицию подстроки <code>subs</code> в строке <code>s</code> , начиная с позиции <code>from</code> . Если не найдена, возвращает 0.
<code>ReverseString(s)</code>	Возвращает инвертированную строку.
<code>RightStr(s, count)</code>	Возвращает последние <code>count</code> символов строки <code>s</code> .
<code>StringOfChar(ch, count)</code>	Возвращает строку, состоящую из <code>count</code> символов <code>ch</code> .
<code>StrToFloat(s)</code>	Преобразует строковое представление вещественного числа <code>s</code> к числовому значению.

Продолжение таблицы

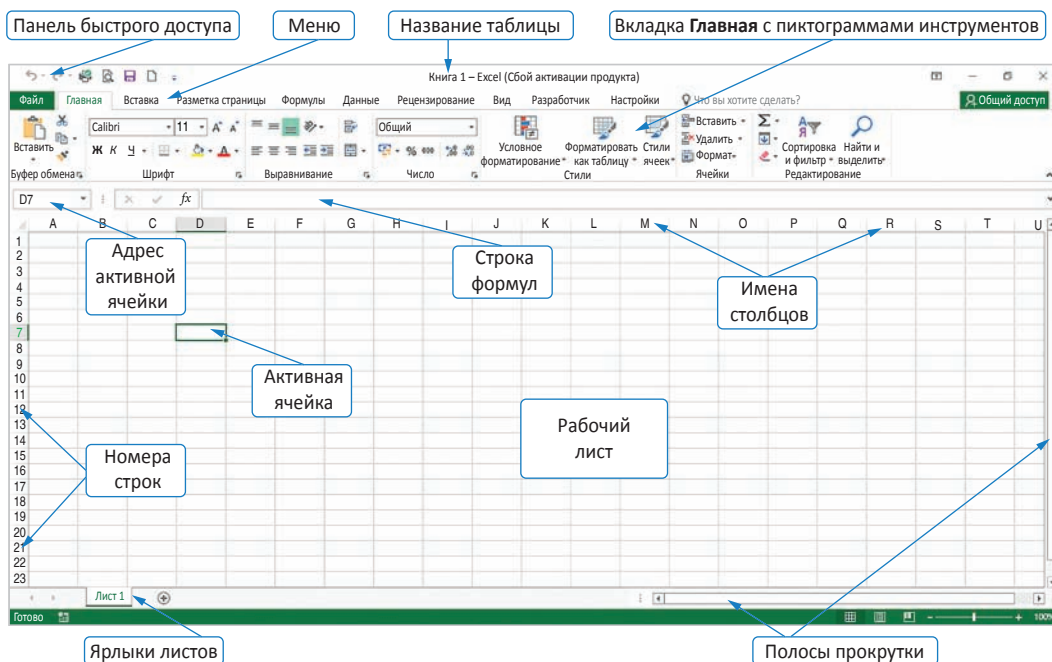
Название подпрограммы	Описание
<code>StrToInt(s)</code>	Преобразует строковое представление целого числа <i>S</i> к числовому значению.
<code>Trim(s)</code>	Возвращает строку с удаленными начальными и конечными пробелами.
<code>UpperCase(s)</code>	Возвращает строку в верхнем регистре.

Процедуры для работы со строками

Название подпрограммы	Описание
<code>Delete(s, index, count);</code>	Удаляет из строки <i>s</i> <i>count</i> символов, начиная с позиции <i>index</i> .
<code>Insert(source, s, index);</code>	Вставляет подстроку <i>source</i> в строку <i>s</i> с позиции <i>index</i> .
<code>Str(i, s);</code>	Преобразует число <i>i</i> к строковому представлению и записывает результат в <i>s</i> . Значение <i>i</i> может быть целым или вещественным.
<code>Val(s, value, err);</code>	Преобразует строковое представление <i>s</i> к числовому значению и записывает его в переменную <i>value</i> . Если преобразование успешно, то <i>err</i> = 0, иначе <i>err</i> > 0.

Приложение 3

Главное окно программы Excel



Часто встречающиеся ошибки

— данная ошибка может возникать, если столбец недостаточно широк для отображения результата или результат вычислений не может быть представлен в указанном для ячейки формате. В первом случае необходимо изменить ширину столбца. Если это не помогает, то необходимо устранить ошибку в ячейке, проверить формат ячейки. Ошибка может возникнуть, если результатом вычислений является отрицательное число, а для ячейки установлен формат **Дата** или **Время**. В этом случае проверьте правильность введенной формулы или измените формат ячейки.

#ЗНАЧ! — возникает при использовании недопустимого типа значений. Например, в формуле для сложения двух значений ячеек используется ссылка на ячейку, содержащую текст. Для исправления такой ошибки необходимо тщательно проверить все ссылки в формуле и формат тех ячеек, на которые ссылается формула.

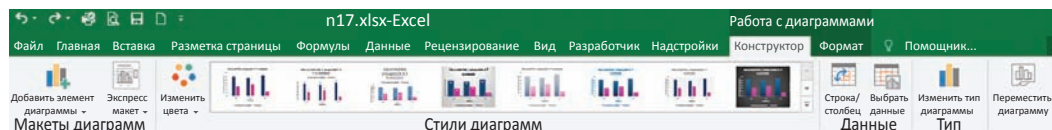
#ДЕЛ/0 — возникает при делении на нуль. Для исправления требуется изменить значение в ячейке, содержащей нуль, или ссылку на ячейку.

#ИМЯ — возникает в том случае, когда Excel не может распознать введенное в формулу имя или ссылку на ячейку. Например, имя некоторой функции введено неверно (СИН вместо SIN). Эта же ошибка выдается, если при написании формулы ссылку на ячейку написать по-русски. Для исправления ошибки нужно исправить неверно введенное имя.

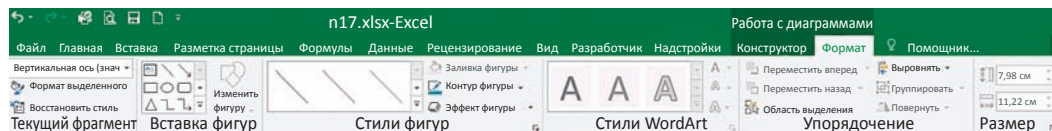
#ЧИСЛО — возникает при использовании недопустимых аргументов функции. Например, при попытке извлечь корень квадратный из отрицательного значения. Для исправления следует проверить ссылки на ячейки и значения в этих ячейках.

Вкладки для работы с диаграммами

Вкладка Работа с диаграммами → Конструктор

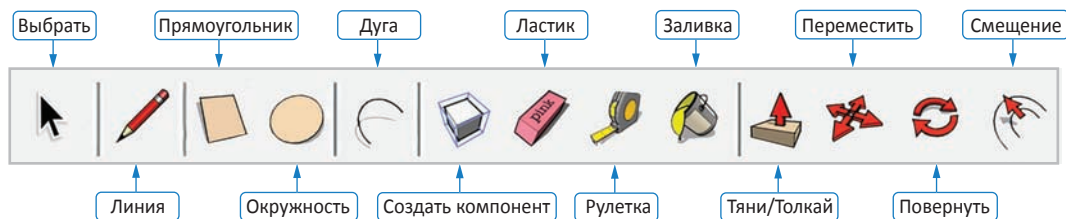
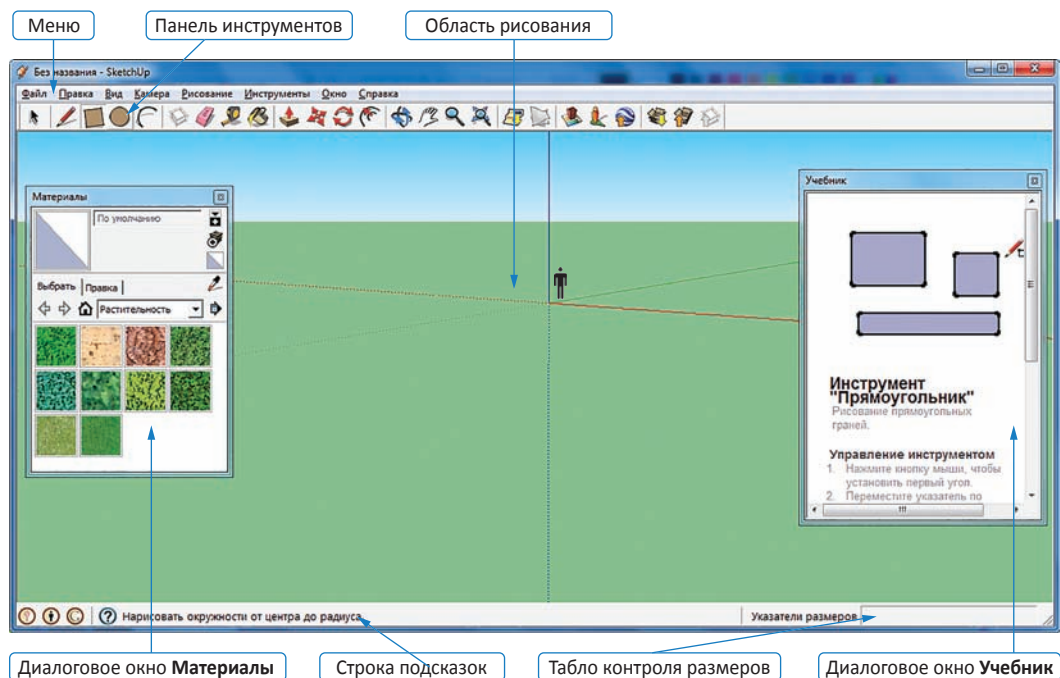


Вкладка Работа с диаграммами → Формат



Приложение 4

Элементы интерфейса 3D-редактора Google SketchUp 8 rus



Диалоговое окно Учебник

Диалоговое окно **Учебник** 3D-редактора Google SketchUp 8 rus открывается командой меню **Окно** → **Учебник**. Окно содержит краткое описание приемов управления инструментами, размещенными на **Панели инструментов** и динамичную анимацию для изображения этих приемов.

В окне описаны приемы управления инструментом, который выбран щелчком мыши на **Панели инструментов**. Щелчок мышью по другому инструменту сразу же меняет содержание окна.

Табло контроля размеров

Табло контроля размеров выполняет сразу несколько функций.

При построении простейших объектов редактора (примитивов) на табло отображаются их линейные размеры.

При выводе ломаной линии инструментом **Линия** отображается длина последнего создаваемого прямолинейного отрезка ломаной. Если в этот момент ввести с клавиатуры численное значение, оно отобразится на табло. Нажатие клавиши **Enter** клавиатуры приведет к построению отрезка указанной длины.

При выводе прямоугольника на табло контроля размеров через точку с запятой выводятся его размеры. При построении после первого щелчка мыши можно с клавиатуры ввести через точку с запятой требуемые размеры, и прямоугольник с такими размерами будет построен нажатием клавиши **Enter** клавиатуры.

При выводе окружности измеряется и контролируется ее радиус.

При выводе дуги сначала выводится и контролируется длина первоначального отрезка, а затем выпуклость — расстояние самой дальней точки дуги от первоначального отрезка.

При использовании инструмента **Тяни/Толкай** на табло выводится расстояние сдвигаемой грани от ее первоначального положения. Ввод с клавиатуры численного значения и нажатие клавиши **Enter** клавиатуры приводят к сдвигу грани на введенное значение в сторону сдвига указателя мыши.

Особенности режима наращивания с инструментом Тяни/Толкай

Режим наращивания включается нажатием клавиши **Ctrl**, а около указателя мыши появляется знак «плюс».

Если инструмент **Тяни/Толкай** используется в режиме наращивания, то величина сдвига грани отображается на табло контроля размеров. После завершения сдвига любой сдвиг указателя мыши приводит к отключению режима наращивания.

Если инструмент **Тяни/Толкай** используется в режиме наращивания и с клавиатуры с нажатием клавиши **Enter** вводится значение величины сдвига грани, то режим наращивания выключается только принудительно нажатием клавиши **Ctrl** клавиатуры.

Учебное издание

Котов Владимир Михайлович

Лапо Анжелика Ивановна

Быкадоров Юрий Александрович

Войтехович Елена Николаевна

ИНФОРМАТИКА

Учебное пособие для 9 класса
учреждений общего среднего образования
с русским языком обучения

Зав. редакцией *Г. А. Бабаева*. Редактор *Е. И. Черникова*. Художественный редактор *Е. А. Проволович*. Обложка *А. Н. Богушевича*. Техническое редактирование и компьютерная верстка *И. И. Дубровской*. Корректоры *В. С. Бабеня, О. С. Козицкая, Е. П. Тхир, А. В. Алешко*.

Подписано в печать 22.07.2019. Формат 70 × 90 ¹/₁₆. Бумага офсетная. Гарнитура школьная. Печать офсетная. Усл. печ. л. 12,29. Уч.-изд. л. 10,0. Тираж 115 500 экз.
Заказ .

Издательское республиканское унитарное предприятие «Народная асвета» Министерства информации Республики Беларусь. Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий 1/2 от 08.07.2013. Пр. Победителей, 11, 220004, Минск, Республика Беларусь.

ОАО «Полиграфкомбинат им. Я. Коласа». Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 2/3 от 10.09.2018.
Ул. Корженевского, 20, 220024, Минск, Республика Беларусь.

Правообладатель Народная асвета

(Название и номер учреждения образования)

Учебный год	Имя и фамилия учащегося	Состояние учебного пособия при получении	Оценка учащемуся за пользование учебным пособием
20 /			
20 /			
20 /			
20 /			
20 /			
20 /			