

Рекомендовано Министерством образования и науки
Республики Казахстан

Г.И. Салгараева
А.С. Маханова
Л.А. Рсалина

ИНФОРМАТИКА

Учебник для учащихся 7 класса уровня основного среднего
образования по обновленному содержанию

7



УДК 373.167.1
ББК 32.973 я 72
С 16

Салгараева Г.И. и др.
С16 Информатика. Учебник для учащихся 7 класса уровня основного среднего образования по обновленному содержанию. / Г.И. Салгараева, А.С. Маханова, Л.А. Рсалина. – Нур-Султан: Издательство «Арман-ПВ», 2021. – 208 с.

ISBN 978-601-318-355-8

Учебник «Информатика» для 7 класса разработан в соответствии с Типовой учебной программой уровня основного среднего образования по обновленному содержанию с учетом возрастных особенностей учащихся. Материал изложен доступным языком, содержание включает дополнительные сведения.

УДК 373.167.1
ББК 32.973 я 72

© Салгараева Г.И.,
Маханова А.С.,
Рсалина Л.А., 2021
© Издательство «Арман-ПВ», 2021

ISBN 978-601-318-355-8

Репродуцирование (воспроизведение) любым способом данного издания без договора с издательством запрещается.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

Задания для формирования функциональной грамотности

1 — Отвечаем на вопросы

2 — Думаем и обсуждаем

3 — Анализируем и сравниваем

4 — Выполняем в тетради

5 — Выполняем на компьютере

6 — Делимся мыслями

7 — Домашнее задание

Вспомните!

*Краткое обобщение
пройденного материала*

Вы узнаете:

*Ожидаемый результат
по усвоению учебного
материала*

Ключевые термины

Научные термины
на трех языках

Определения, которые
нужно запомнить

Внимание

При необходимости вы всегда сможете найти CD с электронным приложением на сайте artan-pv.kz и загрузить его на свой компьютер для дальнейшей работы.

Предисловие

Дорогие семиклассники! Поздравляем вас с началом нового учебного года! Вы продолжите изучать курс информатики и углубите знания по данному предмету, который обеспечивает базовыми знаниями, умениями и навыками, необходимыми для эффективной работы с современными технологиями.

Учебник состоит из таких разделов, как: «Измерение информации и компьютерная память», «Сети и безопасность», «Решение задач с помощью электронных таблиц», «Программирование алгоритмов на языке Python», «Практическое программирование».

После каждого параграфа даны 6 пошаговых заданий, направленных на самостоятельное освоение новой темы и формирование функциональной грамотности. Задания блоков «Отвечаем на вопросы» и «Думаем и обсуждаем» предназначены для изучения наиболее важной информации и содержат вопросы на выявление причинно-следственных связей. Выполнение заданий блока «Анализируем и сравниваем» способствует развитию логического и творческого мышления. Задания «Выполняем в тетрадах» включают задания на сбор информации по теме, составление таблиц, постеров и др., развивая навыки самостоятельного поиска необходимой информации. Задания «Выполняем на компьютере» позволяют связать теорию полученных знаний с практикой, а также научат работать и грамотно использовать ИКТ в нужных целях. С помощью заданий блока «Делимся мыслями» можно обменяться полученными знаниями (по теории) и поделиться своими мыслями в классе о пройденном материале (по результатам практической работы).

В конце каждого раздела представлены контрольные задания для самостоятельной проверки полученных знаний.

К учебнику прилагается электронное приложение, которое помогает закрепить пройденный материал в процессе выполнения практических интерактивных заданий и тестов по каждому разделу курса.

При изучении новых знаний и опыта вы сможете размышлять и находить ответы на различные вопросы, прийти к общему выводу через совместное обсуждение, совершенствовать свою личную информационно-коммуникативную грамотность при выполнении практических или творческих учебных заданий. Мы надеемся, что эту возможность вы в дальнейшем будете успешно использовать.

Желаем удачи!

ИЗМЕРЕНИЕ ИНФОРМАЦИИ И КОМПЬЮТЕРНАЯ ПАМЯТЬ

Изучив раздел, вы научитесь:

- осуществлять перевод из одних единиц измерения информации в другие;
- описывать назначение видов памяти компьютера (оперативные запоминающие устройства, постоянные запоминающие устройства, внешние запоминающие устройства, кэш-память);
- сравнивать размеры файлов разных форматов, хранящих одинаковую информацию;
- создавать и распаковывать архивы различных форматов.

§ 1. Единицы измерения информации

Вспомните!

- Что такое информация?
- Какие единицы информации вы знаете?

Вы узнаете:

- об единицах измерения информации;
- как перевести информацию из одной единицы измерения в другую.

Бит – Бит – Bit

Двоичное число – Екілік сан – Binary digit

Байт – Байт – Byte

Единица измерения – Өлшем бірлігі – Unit of measure

Объем информации важно знать точно так же, как и площадь или размер любого предмета. Информацию тоже можно измерить, как время, массу, длину и т. п. (схема 1). Рассмотрим единицы измерения времени, массы, длины и температуры.

Масса	Длина	Время	Температура
Миллиграмм, грамм, килограмм, центнер, тонна	Миллиметр, сантиметр, дециметр, метр, километр	Секунда, минута, час, сутки, месяц, год, век	Градус Цельсия °С, градус Фарангейта °F, Кельвин К
Для измерения массы используются весы	Для измерения длины используется линейка	Для измерения времени используется секундомер	Для измерения температуры используется термометр
			

Схема 1. Единицы измерения величин и измерительные приборы

Измерение вышеуказанных величин осуществляется с помощью специальных измерительных приборов. Чтобы измерить информацию, мы используем компьютер, который принимает ее с помощью электрических сигналов. Для обработки информации каждый символ переводится на вид сигнала. Все виды информации, хранящиеся в памяти компьютера – слова, цифры, рисунки, программы, – записываются в виде последовательности двоичных чисел. Если сигнал отсутствует, то сигнал обозначается числом 0, при наличии сигнала обозначается числом 1. Сигналы «0» или «1» называются битами. Бит принимает только одно из двух значений, таких как 0 или 1 (рис. 1).



Рис. 1. Передача информации в память компьютера

Наименьшей единицей измерения информации является **бит** (англ. bit «binary digit» – «двоичное число»).

В одном бите очень мало информации, поэтому измерять информацию с помощью бита очень неудобно. Это все равно, что сутки мерять в секундах. Предположим, что учитель задает ученикам домашнее задание и предупреждает их, что размер файла, в котором они будут работать, не должен превышать 8 388 608 бит. Ученики вряд ли поймут, файл какого размера нужно будет отправить. Поэтому в информатике и в жизни используются единицы измерения, большие, чем бит информации.

Следующей единицей измерения информации является **байт**. 1 байт равен 8 бит.

При переходе от большей единицы измерения информации к меньшей необходимо выполнять **умножение**. При переходе от меньшей единицы измерения информации к большей необходимо выполнять **деление**.

Делим ↓	1 байт = 8 бит	↑ Умножаем
	1 Килобайт (Кбайт) = 2^{10} байт = 1024 байт	
	1 Мегабайт (Мбайт) = 2^{20} байт = 1024 Кбайт	
	1 Гигабайт (Гбайт) = 2^{30} байт = 1024 Мбайт	
	1 Терабайт (Тбайт) = 2^{40} байт = 1024 Гбайт	
	1 Петабайт (Пбайт) = 2^{50} байт = 1024 Тбайт	
	1 Эксабайт (Эбайт) = 2^{60} байт = 1024 Пбайт	
	1 Зеттабайт (Збайт) = 2^{70} байт = 1024 Эбайт	
	1 Йоттабайт (Йбайт) = 2^{80} байт = 1024 Збайт	

Пример 1. В какую степень нужно возвести число 2, чтобы получить равенство 1 Гбайт = 2^x байт?

Дано:

$X = 1$ Гбайт

1 Гбайт – 2^x – ?

Решение:

1 Гбайт – это 1024 Мбайт

1 Мбайт – это 1024 Кбайт

1 Кбайт – это 1024 байт

1024 байт – это 2^{10}

1 Гбайт = $1024 * 1024 * 1024 = 2^{10} * 2^{10} * 2^{10}$

$2^{10} * 2^{10} * 2^{10} = 2^{10+10+10} = 2^{30}$

Ответ: 1 Гбайт = 2^{30} байт. Значение степени равно 30.

С помощью 1 байта кодируется один символ.

Символ – любой знак, такой как буква, цифра, знак препинания или пробел. Приведем несколько примеров для сравнения объема разной информации (*схема 2*).

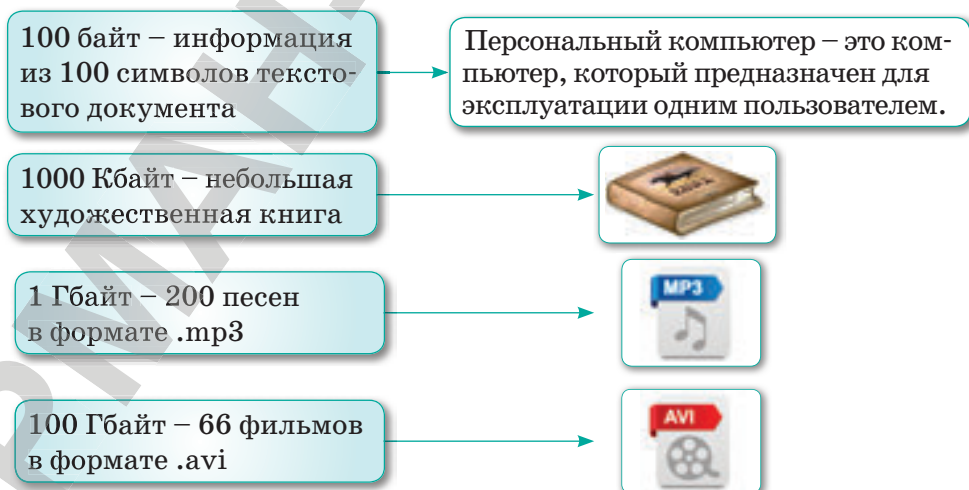


Схема 2. Сравнение объемов разной информации

Пример 2. Определим, сколько байт в 1 Мбайт.

Дано:

$X = 1$ Мбайт

$X_{\text{байт}} - ?$

Решение:

$X_{\text{байт}} = 1 * 1024 * 1024 = 1048576$ байт

Ответ: 1048576 байт.

Пример 3. Определим информационный объем слова «Информатика».

Дано:

слово «Информатика»

$X_{\text{бит}} - ?$

Решение:

1 символ = 1 байт

Информатика = 11 символов = 11 байт.

1 байт = 8 бит.

$X_{\text{бит}} = 11 \text{ байт} * 8 \text{ бит} = 88 \text{ бит}.$

Ответ: информационный объем слова

«Информатика» – 88 бит.

Пример 4. В книге 50 страниц. Каждая страница содержит 40 строк, а в каждой строке – 60 символов. Определим, сколько Мбайт составляет информационный объем этой книги.

Дано:

$X_{\text{книга}} =$

50 страниц,

40 строк,

60 символов

$X_{\text{Мбайт}} - ?$

Решение:

$X_{\text{книга}} = 50 * 40 * 60 = 120000$ байт

$X_{\text{Кбайт}} = 120000 \text{ байт} / 1024 = 117,1875$ Кбайт

$X_{\text{Мбайт}} = 117,1875 \text{ Кбайт} / 1024 = 0,1144$ Мбайт

Ответ: информационный объем книги 0,1144 Мбайт.

Пример 5. Во сколько раз информация в 3 Мбайт больше информации в 32 Кбайт?

Дано:

$X_1 = 3$ Мбайт

$X_2 = 32$ Кбайт

X_1 / X_2

Решение:

$X_1 = 3 \text{ Мбайт} = 3 * 1024 = 3072$ Кбайт

$X_1 / X_2 = 3072 \text{ Кбайт} / 32 \text{ Кбайт} = 96$

Ответ: в 96 раз

Пример 6. Какую часть занимает книга из 350 страниц в носителе информации, объемом в 600 Мбайт? На одной странице книги имеется 40 строк из 60 символов.

Дано:

$X_{\text{книга}} = 350 \text{ стр, } 40 \text{ строк,}$
 60 символов

$X_{\text{носитель информации}} = 600 \text{ Мбайт}$

$X_{\text{книга}} / X_{\text{носитель информации}} - ?$

Решение:

$X_{\text{книга}} = 350 * 40 * 60 = 840000 \text{ байт}$

$X_{\text{носитель информации}} = 600 \text{ Мбайт} =$

$600 * 1024 * 1024 = 629145600 \text{ байт}$

$X_{\text{книга}} / X_{\text{носитель информации}} = 840000 \text{ байт} /$

$629145600 \text{ байт} = 0,0013$

Ответ: 0,0013 (часть)

В настоящее время объем компьютерных носителей информации измеряется мегабайтами и гигабайтами. Больше всего объема информации занимают графические файлы и еще больше объема – видеофайлы. Также используются носители информации, имеющие 1–2 терабайтных объема (например, жесткие диски) (рис. 2).



Рис. 2. Носитель информации объемом в 1–2 терабайта

Пример 7. Необходимо рассчитать объем изображения размером 240 x 400 пикселей, каждый из которых кодируется 32 битами или 4 байтами. Определим, какой объем информации получит приведенное изображение в килобайтах, мегабайтах.

Дано:

Рисунок = 240x400 пикселей

Найти:

$X_{\text{Мбайт}} - ?$

$X_{\text{Кбайт}} - ?$

Решение:

$240 * 400 * 4 = 384000 \text{ байт}$

$X_{\text{Кбайт}} = 384000 \text{ байт} / 1024 = 375 \text{ Кбайт.}$

$X_{\text{Мбайт}} = 375 \text{ Кбайт} / 1024 = 0,37 \text{ Мбайт}$

Ответ: 375 Кбайт, 0,37 Мбайт

1

Отвечаем на вопросы

1. Что такое бит?
2. Сколько бит в 1 байте?
3. Чему равен 1 Эбайт?
4. Как определить объем информации?
5. Как перевести информацию из одной единицы измерения в другую?
6. Что такое байт, Кбайт, Мбайт, Гбайт? Как они связаны между собой?

2

Думаем и обсуждаем

1. Для чего нужно измерять информацию?
2. Какие виды величин (масса, время, длина и т. д.), которые часто используются в повседневной жизни, вы знаете?
3. Почему неудобно измерять информацию в битах?
4. Зачем переводят информацию с большей единицы измерения в меньшую? Приведите примеры.

3

Анализируем и сравниваем

Сравните и проанализируйте различные единицы измерения. Заполните таблицу:

Бит	Байт	Кбайт	Мбайт
24576			
	2048		
		1,5	
34359738368			
	105960		
			0,008

4

Выполняем в тетради

Решите примеры:

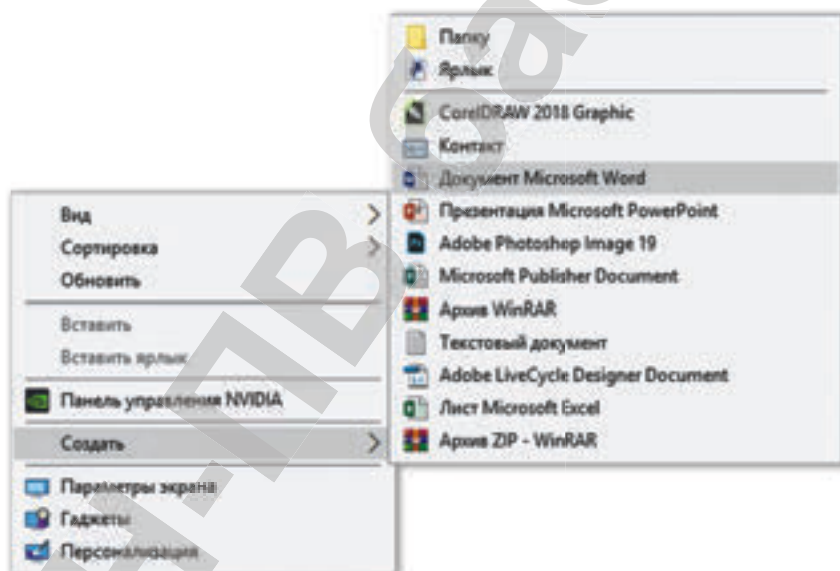
1. Переведите информацию объемом 3 Мбайт в Кбайт, а затем в байты.
2. Определите информационный объем слов «Казахстан», «Информация».
3. В книге 250 страниц. Каждая страница содержит 40 строк, а в каждой строке – 65 символов. Определите информационный объем книги и переведите объем книги в единицы измерения Кбайт и Мбайт.

4. Машина имеет носитель информации объемом 8 Гбайт. Сколько электронных версий книги в 200 страниц можно сохранить на этом носителе информации, если каждая страница книги содержит 50 строк, а каждая строка – 60 символов?
5. Во сколько раз информация в 4 Тбайт больше информации в 4 Збайт?

5

Выполняем на компьютере

1. Создайте новый объем информации.
 - 1) Щелкните правой кнопкой мыши на рабочем столе компьютера. Выберите документ Microsoft Word в появившемся окне «Создать».



- 2) Наберите следующий текст в документе MS Word:
Пока не знаешь – молчи.
Пока блуждаешь – молчи.
В бесцельных днях и в ночи
Пустых забав не ищи.
Чтоб человеком ты был
И вровень с веком ты был,
Ты пятерых побори
И пятерых избери.

(Абай Кунанбаев)

- 3) Сохраните набранный текстовый документ, выбрав строку «Свойства».
 - 4) В появившемся окне «Свойства» найдите размер документа. Определите, в каких единицах измерения представлен объем документа.
 - 5) Перенесите определенный объем документа из одной единицы измерения в другую. Вы создали новый объем информации.
2. Определите единицы измерения информации.
- 1) Выберите любую папку или файл на рабочем столе компьютера.
 - 2) Нажав на правую кнопку мыши, вызовите контекстное меню и выберите команду «Свойства».
 - 3) В открывшемся окне определите объем информации из окна Размер.
 - 4) Переведите определенный объем информации из одной единицы измерения в другую.
 - 5) Сделайте перевод определенного объема информации из одной единицы измерения в другую.

6

Делимся мыслями

Как вы думаете, можно ли использовать полученные на этом уроке знания при решении повседневных задач в будущем?

7

Домашнее задание

Ответьте на вопросы письменно:

1. Какая единица измерения больше – 1 Мбайт или 1 Кбайт?
2. Сколько бит в одном байте?
3. Сколько байт в одном килобайте?
4. Сколько мегабайт в одном гигабайте?
5. Сколько килобайт в одном мегабайте?
6. Переведите 82 байт в бит.
7. Переведите 3 Кбайт в байт.
8. Сколько байтов в 4 Тбайт?
9. Сколько байтов в 64 бит?
10. Сколько символов и сколько бит в предложении: «Республика Казахстан – 9-я по величине страна в мире»?

§ 2. Практикум. Единицы измерения информации

Уровень А

Задание 1. Заполните пропуски в таблице единицами измерения информации:

1 Эбайт	= ...	= 1024 Пбайт
1 ...	= 2^{10} байт	= 1024 байт
1 Йбайт	= 2^{80} байт	= ...
1 Мбайт	= ...	= 1024 Кбайт
1 Тбайт	= 2^{40} байт	= 1024 ...
... бит	= ... байт	
1 Гбайт	= ...	= 1024 Мбайт
1 ...	= 2^{50} байт	= 1024 Тбайт
1 Збайт	= 2^{70} байт	= 1024 Эбайт

Задание 2. Напишите ваши Ф.И.О. и определите общее количество символов. Переведите информацию из одной единицы измерения в другую.

Фамилия, имя, отчество: ...
... бит, ... байт, ... Кбайт, ... Мбайт.

Задание 3. Переведите из одной единицы измерения в другую следующую информацию.

16 бит = ... байт
3,2 Мб = ... байт
4 Гб = ... Кбайт
7 байт = ... бит
16 Кб = ... бит

88 бит = ... байт
1,8 Мб = ... байт
11,5 Гб = ... Кбайт
13 байт = ... бит
64 Мб = ... байт

Уровень В

Задание 1. Определите информационный объем:

1. На рабочем столе компьютера выберите значок «Мой компьютер».
2. В выбранном окне щелкните правой кнопкой мыши, отметив жесткий диск компьютера. Выберите команду «Свойства».

3. Определите из появившегося окна количество информации, равное объему информации в памяти жесткого диска (рис. 3):



Рис. 3. Память жесткого диска

Задание 2. Переведите единицу измерения объема выявленной информации в другую единицу измерения.

Уровень С

Задание. Рассчитайте:

1. Во сколько раз информация в 32 Тбайт меньше информации в 3 Пбайт?
2. Существует устройство для передачи информации, составляющей 10 Гбайт. Сколько книг, состоящих из 300 страниц, на каждой странице по 40 строк, а в каждой строке по 60 символов, могут храниться на этом устройстве?
3. У Азамата есть устройство передачи информации емкостью 256 Мбайт, а у Лауры – устройство емкостью 128 Гбайт. Определите, во сколько раз емкость устройства передачи информации Лауры больше емкости устройства Азамата.
4. Посчитайте, сколько информации хранится в книге, состоящей из 200 страниц, и осуществите перевод информации из одной единицы измерения в другую (из байта в Кбайт, из Кбайта в Мбайт). Каждая страница имеет 50 строк, а каждая строка – 70 символов.

§ 3. Память компьютера

Вспомните!

- Какие единицы информации вы знаете?
- Как переводить информацию из одной единицы измерения в другую?

Вы узнаете:

- что такое память;
- о назначении типов памяти компьютера;
- об оперативном запоминающем устройстве,
- о постоянном запоминающем устройстве,
- о внешних запоминающих устройствах и кэш-памяти.

Память – Жады – Memory

Компьютерная память –
Компьютер жады –
Computer memory

Оперативное запоминающее устройство – Жедел есте сақтау құрылғысы – Random access memory

Постоянное запоминающее устройство – Тұрақты есте сақтау құрылғысы – Read only memory

Кэш-память – Кэш-жады – Cache memory

Внешняя память – Сыртқы жады – External memory

Что такое память компьютера?

Компьютерная память – это устройство хранения информации, или запоминающее устройство. Память является обязательной частью компьютера. Задача компьютерной памяти – хранение информации. Память компьютера состоит из пронумерованных ячеек, которые имеют адреса. Ячейка памяти может содержать полуслово (2 байта), слово (4 байта) или двойное слово (8 байт). Эти ячейки могут фиксировать самые разнообразные физические воздействия, и информация в них записывается в виде двух четко различимых состояниях 0 и 1 («выключено»/«включено»). Память в вычислительных устройствах имеет иерархическую структуру и состоит из нескольких запоминающих устройств, имеющих различные характеристики.

Всю память можно разделить на внутреннюю и внешнюю (схема 3).

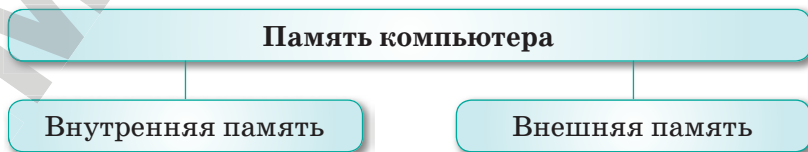


Схема 3. Типы памяти компьютера

Внутренняя память компьютера выполняет основные функции, так как процессор работает непосредственно с этой памятью.

Внутренняя память – это электронное устройство, расположенное в системном блоке компьютера. Оно предназначено для хранения информации и программ в процессе работы. Внутренняя память подразделяется на следующие элементы (схема 4):

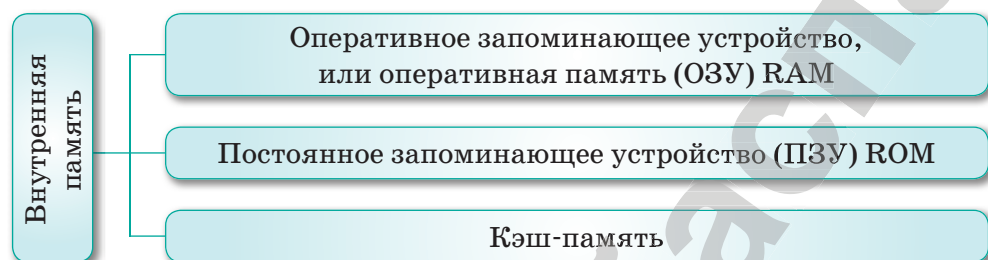


Схема 4. Элементы внутренней памяти компьютера

Оперативное запоминающее устройство, или оперативная память (ОЗУ, англ. Random Access Memory, RAM, память свободного доступа) – устройство записи, чтения и быстрого временного хранения данных для компьютерных вычислений (рис. 4). Любая информация, требующая обработку, записывается сначала из внешней памяти компьютера (магнитных дисков) в оперативную

память. Процессор получает информацию из оперативной памяти и обрабатывает ее. Оперативная память разделяется на сегменты при записи данных, эффективно размещая и передавая данные через сегмент памяти. Оперативная память используется для временного хранения данных и программ. При отключении компьютера данные в оперативной памяти будут потеряны.

Например, если вы работаете на компьютере, то данные, относящиеся к этой работе (графические изображения, функции, команды), загружаются в оперативную память и передаются в процессор, процессор их обрабатывает, а затем информация отображается на экране монитора в виде графических данных. Если компьютер отключится, то сведения в оперативной памяти будут удалены, поэтому вам придется возобновить свою работу.



Рис. 4. Оперативное запоминающее устройство

Постоянное запоминающее устройство (ПЗУ, англ. Read only memory, ROM-память только для чтения) – устройство для хранения постоянной программной и справочной информации. Данные в ПЗУ заносятся при изготовлении. Информацию, хранящуюся в ПЗУ, можно только считывать, но не изменять (рис. 5). Важной микросхемой постоянной памяти является модуль BIOS (англ. basic input/output system) – базовая система ввода-вывода, совокупность программ ввода-вывода, организующих взаимодействие между компьютерными устройствами BIOS.



Рис. 5. Постоянное запоминающее устройство

При запуске компьютера или ноутбука в первую очередь начинает загружаться модуль BIOS, который автоматически проверяет все элементы устройства компьютера, отслеживает и сообщает, правильно ли работают устройства компьютера или есть ли отклонения. Только после проверки всех аппаратных компонентов операционная система будет загружена (рис. 6).

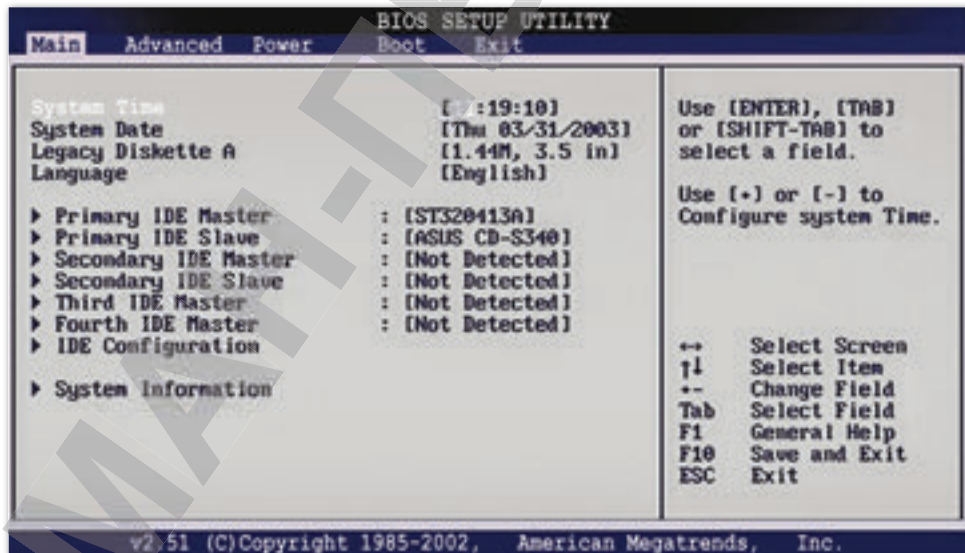


Рис. 6. Модуль BIOS

Цель BIOS:

- проверка работоспособности компьютерного оборудования;
- загрузка операционной системы;

- предоставление API (интерфейс прикладного программирования, т. е. описание методов, при которых одна компьютерная программа может взаимодействовать с другой программой) для работы с оборудованием;
- использование компьютерного оборудования.

Кэш-память (англ. *cache* – «хранилище», «секретное место») используется при обмене данными между процессором и оперативной памятью (рис. 7). Целью кэш-памяти является снижение частоты доступа процессора к оперативной памяти, что увеличивает производительность компьютера. Кэш-память отличается от оперативной памяти скоростью работы. На практике скорость работы кэш-памяти в десять раз выше оперативной памяти простого компьютера.



Рис. 7. Кэш-память

Второй вид памяти компьютера – внешняя память.

Внешняя память – устройство для длительного хранения программ и информации (магнитные, оптические диски и флеш-памяти) с помощью носителей информации (схема 5).



Схема 5. Внешняя память компьютера

Магнитные диски

Жесткий диск (англ. *Hard Disk Drive, HDD*, накопитель на жестких магнитных дисках, винчестер) – запоминающее устройство большой емкости. В таком диске носители информации представляют собой круглые алюминиевые пластины, две из которых покрыты слоем магнитного материала. Емкость информации составляет до сотни Гбайт.

Гибкий диск (англ. *floppy disk*) – носитель информации небольшой емкости, представляющий собой гибкий пластиковый диск в защитной оболочке. Для передачи данных с одного компьютера на другой и передачи программного обеспечения. Емкость информации составляет 1,44 Мбайт.

Флеш-память

Флеш-память (англ. *flash memory*) – современный носитель информации. Во флэш-памяти для записи и считывания информации используются электрические сигналы. Информация, записанная на флеш-память, может храниться очень длительное время. В отличие от магнитных и оптических дисков расходуется меньше энергии. Динамика развития емкости флеш-памяти высока. В настоящее время действуют емкости 512 Мбайт, 1 Гбайт, 2 Гбайт, 4 Гбайт, 8 Гбайт, 16 Гбайт и др.

Оптические диски

CD-R (англ. *Compact Disk Read*, диск для чтения) и *DVD-R* (англ. *Digital Video Disc – Read*, цифровой видеодиск для чтения) – оптический носитель информации, предназначенный для однократной записи информации. Различия *CD-R* и *DVD-R* зависят от емкости диска. Помимо чтения диски *CD-R* способны записывать информацию на специальные оптические диски емкостью 700 Мбайт. Объем диска *DVD-R* составляет 4,7 Гбайт.

CD-RW (англ. *Compact Disk – Read Write*, диск для чтения и записи) и *DVD-RW* (англ. *Digital Video Disc – Read Write* – цифровой видеодиск для чтения и записи) – носитель, который может записывать, удалять, переписывать информацию несколько раз. Разница в дисках в емкости информации. Вы можете переписать *DVD-RW* на диск 1000 раз. Объем такого диска составляет 1,4 Гбайт–17 Гбайт.

1

Отвечаем на вопросы

1. Что такое память?
2. Что такое память компьютера?
3. На сколько видов делится компьютерная память?
4. Что такое внутренняя память?
5. В чем отличие оперативной памяти и постоянного запоминающего устройства?
6. В чем преимущества кэш-памяти?
7. На какие виды делится внешняя память?
8. Как различить диски CD-R и CD-RW?

2

Думаем и обсуждаем

1. Для чего используется компьютерная память?
2. В чем необходимость памяти компьютера?
3. Почему память делится на внутреннюю и внешнюю?
4. Какие носители внешней памяти эффективны? Почему?

3

Анализируем и сравниваем

Составьте диаграмму Венна сходств и различий внутренней и внешней памяти компьютера.

4

Выполняем в тетради

Заполните таблицу в тетрадях:

Внутренняя память компьютера	Наименование	Описание	Функции
			
			
			

5

Выполняем на компьютере

1. Запустите программу PowerPoint.
2. Подготовьте презентацию на тему «Память компьютера». Вставьте в презентацию картинки из Интернета.
3. Под каждой картинкой вставьте название и краткое описание.
4. Установив фон и дизайн, оформите презентацию.
5. Сохраните презентацию в своей папке и защитите.

6

Делимся мыслями

Работа какого органа человека выполняет особую функцию в сохранении полученной информации?

7

Домашнее задание

Заполните таблицу:

Носители информации					
Название носителя информации					
Описание					
Применение					
Преимущества					

§ 4. Размер файлов. Сравнение размеров файлов разных форматов

Вспомните!

- Что такое память?
- Что такое память компьютера?
- На сколько видов делится компьютерная память?
- Что такое оперативная память?
- Что такое постоянное запоминающее устройство?
- Какие характеристики имеют внешняя память и кэш-память?

Вы узнаете:

- что такое концепция файла;
- что такое имя файла и расширение;
- об основных свойствах файлов, типах и файловых системах;
- о размере файла;
- как сохранить одну и ту же информацию в файлах разных форматов и сравнивать размеры.

Файл – *Файл* – File

Расширение файла – *Файлдің кеңейтілуі* – Expansion

Тип файла – *Файл түрі* – File type

Файловая система – *Файлдар жүйесі* – File expansion

Размер файла – *Файл мөлшері* – File size

Все программы и данные хранятся в виде файлов во внешней памяти компьютера. В терминологии Windows понятие файла имеет особое значение.

Что такое файл?

Файл (англ. *file* – «цепочка») – совокупность данных, хранящихся на компьютере в виде бит-цепочек.

Каждый файл характеризуется именем файла и расширением». Эти два элемента обычно разделяются точкой. Имя файла ОС Windows может содержать до 255 символов латинского или кириллического алфавита, а также цифры. Разрешается использовать пробелы и другие ранее запрещенные символы, за исключением следующих: \, /, :, *, ?, », <, >, [], { }. Эти символы являются «служебными» для системы, поэтому Windows ограничивает ненужное использование. Если первый символ имени файла содержит пробел, ОС Windows не обращает на него внимания, так как это не считается ошибкой,

воспринимает как символ. После названия файла с помощью «.» описываются файловые расширения.

Расширение – последовательность символов, предназначенная для определения типа файла. Оно состоит из нескольких (3 или 4) английских букв или цифр. Например, возьмем файл

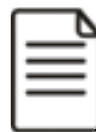
«Документ.docx», здесь «Документ» – это название файла, «.docx» – его расширение (рис. 8).

Имя файла может выбрать пользователь, а изменение расширения не допускается.

К основным свойствам файла относятся:

- 1) имя и расширение, например, «Документ.docx»;
- 2) тип файла;
- 3) размер;
- 4) дата/время;
- 5) атрибуты (скрытые, системные и другие) и права доступа.

Чтобы просмотреть основные свойства файла, выделите выбранный файл и нажмите правую кнопку мыши. В появившемся списке выберите команду «Свойства» (рис. 9).



Документ.docx

Рис. 8. Описание расширения файла

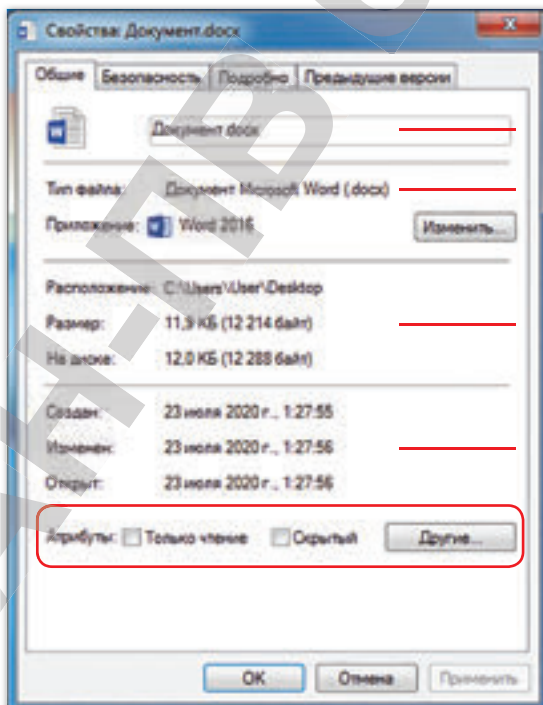


Рис. 9. Просмотр основных свойств файла

Тип файла (англ. *file type*) – описание файла, система, описывающая, к какому типу относится и в какой программе

он открывается. Отображается в основных свойствах файла. Например, «Документ.docx» (рис. 10).

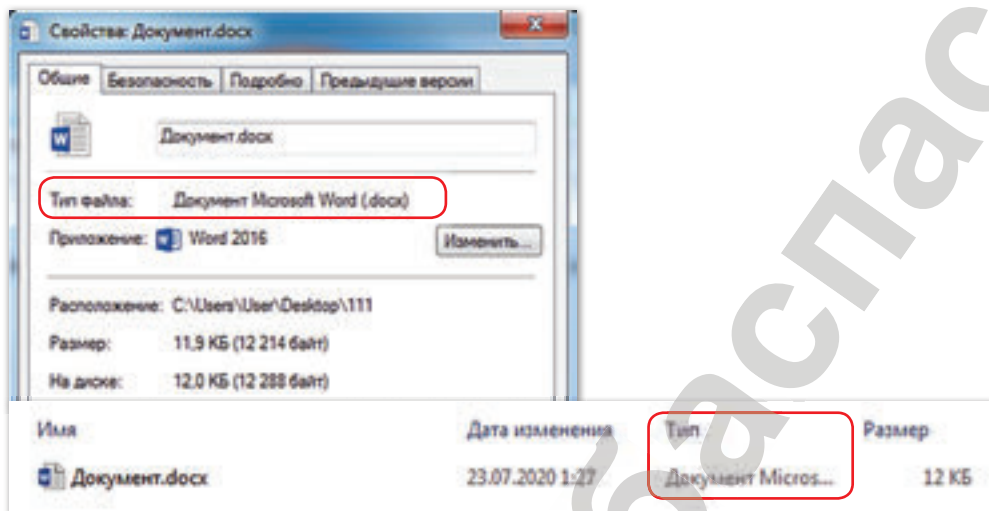


Рис. 10. Описание типа файла

Существуют сотни различных типов файлов и расширений файлов, используемых на компьютерах. Ниже представлен список распространенных расширений, разделяемых на категории по типам файлов (табл. 1).

Таблица 1. Список типов файлов и расширений

Тип файла	Расширения
Системный файл	.drv, .sys, .cab, .msi, .bak, .lnk, .cpl
Программный файл	.c, .pyw, .pyz, .java, .cpp
Архивный файл	.7z, .rar, .arj, .pkg, .tar, .gz
Текстовый файл	.doc, .docx, .pdf, .txt, .odt, .tex
Презентационный файл	.pps, .ppt, .pptx, .key, .odp
Табличный файл	.xls, .ods, .xlsm, .xlsx
Графический файл	.gif, .png, .jpg, .tiff, .bmp
Видеофайл	.avi, .mp4, .wmv, .vob, .mpg или .mpeg
Аудиофайл	.mp3, .aif, .wav, .wma, .mpa

В операционной системе Windows для того, чтобы файлы, записанные во внешнюю память компьютера, не перемешались, они сортируются с помощью типов и расширений.

Файлы хранятся в каталоге, то есть в *файловой системе*, которая определяет данные по организации, хранению, названию и ограничивает формат и доступ данных. Файловая система подразделяется на простую и иерархическую (рис. 11–12).



Рис. 11. Простая файловая система

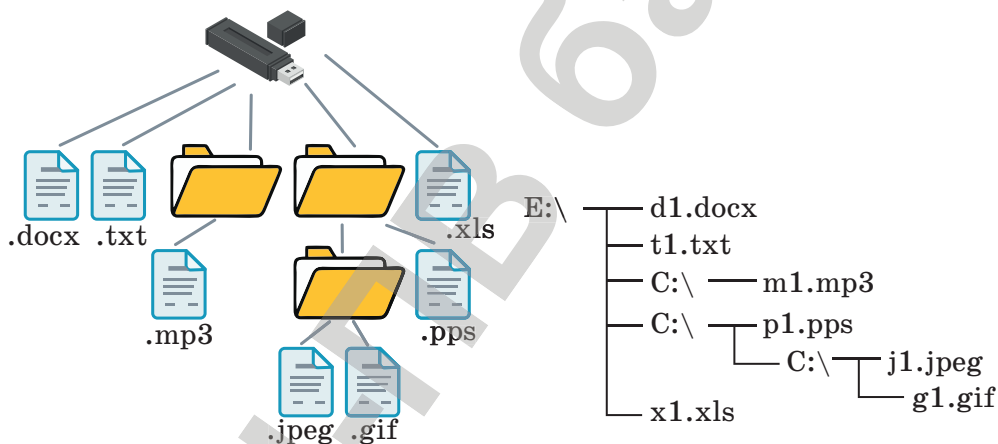


Рис. 12. Иерархическая файловая система

С помощью вышеуказанной иерархической файловой системы пользователь может создать свою систему хранения информации. Однако, чтобы сохранить файлы, важно знать размер файла.

Размер файла (объем, вес) описывает, сколько информации или сколько объема памяти занимает файл. Обычно размер файла указывается в единицах измерения: бит, байт, килобайт, мегабайт, гигабайт, терабайт, петабайт, эксабайт, зеттабайт, йоттабайт.

Файл занимает гораздо больше памяти, чем требуется при записи на диск по файловой системе. Размер файла округляется, учитывая память сектора, которая не используется на диске.

Сектор – наименьшая адресная единица хранения информации в дисковых запоминающих устройствах (жесткий магнитный диск, гибкие диски, CD – компакт-диск). Например, ниже мы наблюдаем различные размеры файла и размер записи на дисковом запоминающем устройстве (рис. 13).

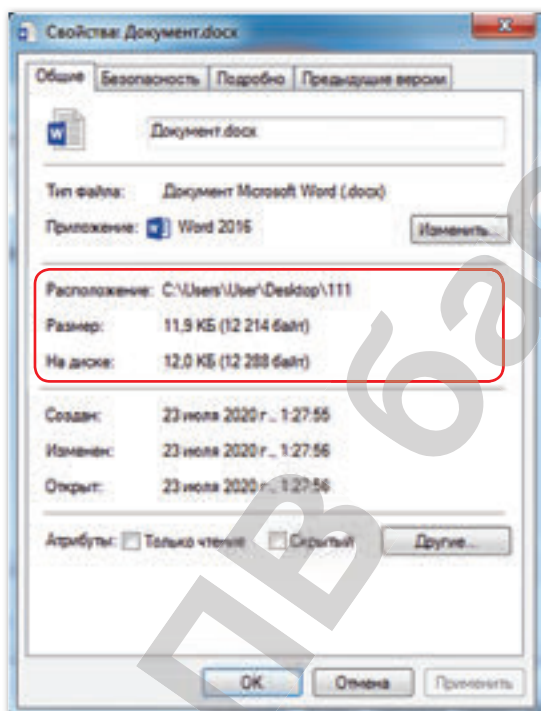


Рис. 13. Описание размера файла

Рассмотрим различные размеры файлов разных форматов, хранящие одинаковую информацию. Введите текст «Робот – это автоматизированное сложное устройство, имитирующее движение человека; машина, которая частично или полностью выполняет функции человека (иногда животного) при взаимодействии с окружающим миром» в файлы разных форматов и сохраните. Для этого:

1. Нажмите кнопку Пуск, щелкните на Блокнот из списка Стандартные и откройте Word из списка Microsoft Office (рис. 14–15).
2. В открытых текстовых файлах различных форматов введите вышеуказанный текст и сохраните на рабочем столе с одинаковым названием «Текстовый файл».

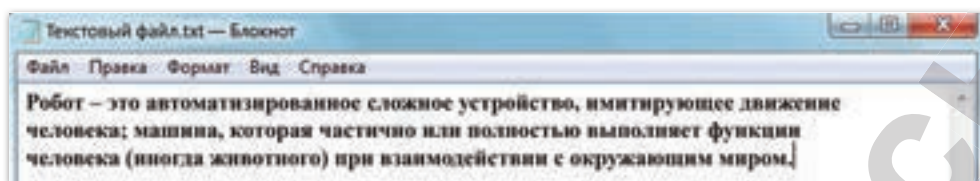


Рис. 14. Формат «.txt» текстового файла

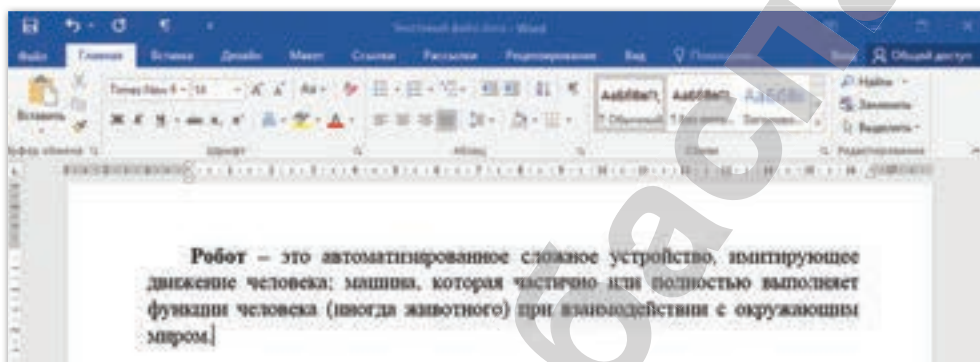


Рис. 15. Формат «.docx» текстового файла

3. Выберите текстовые файлы на рабочем столе, при помощи правой кнопки мыши вызовите контекстное меню и откройте команду «Свойства» (рис. 16).

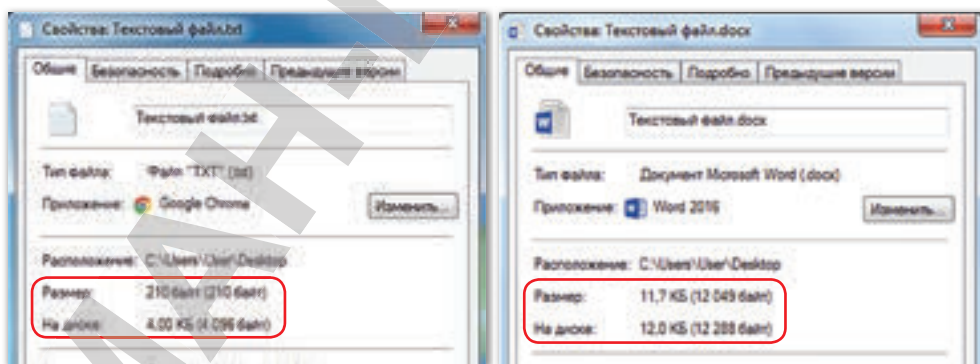


Рис. 16. Размеры файлов в форматах «.txt» и «.docx» с одинаковой сохраненной информацией

Размер файла с одинаковой сохраненной информацией в формате «.txt» – 210 байт, а на внешнем запоминающем устройстве – файл в формате «.docx» – 11,7 Кбайт (12049 байт),

округленный до 4,00 Кбайт (4096 байт), на внешнем запоминающем устройстве – файл, округленный до 12,0 Кбайт (12288 байт).

Если сравнить размеры файлов в формате «.docx» и «.txt», в которых сохранена одинаковая информация, то можно заметить, что размер файла в формате «.docx» весит больше.

Вы убедились, что размеры текстовых файлов разных форматов имеют различия, и данные единицы измерения округляются в памяти Диска.

Можно сохранить информацию в одном файле в различных форматах. Для этого:

1. Нажмите кнопку Пуск и откройте Word из списка Microsoft Office.
2. Введите указанный выше текст в открывшемся текстовом файле. Обратите внимание на ход сохранения (рис. 17). Можно сохранить один файл с одинаковым именем в разных форматах.

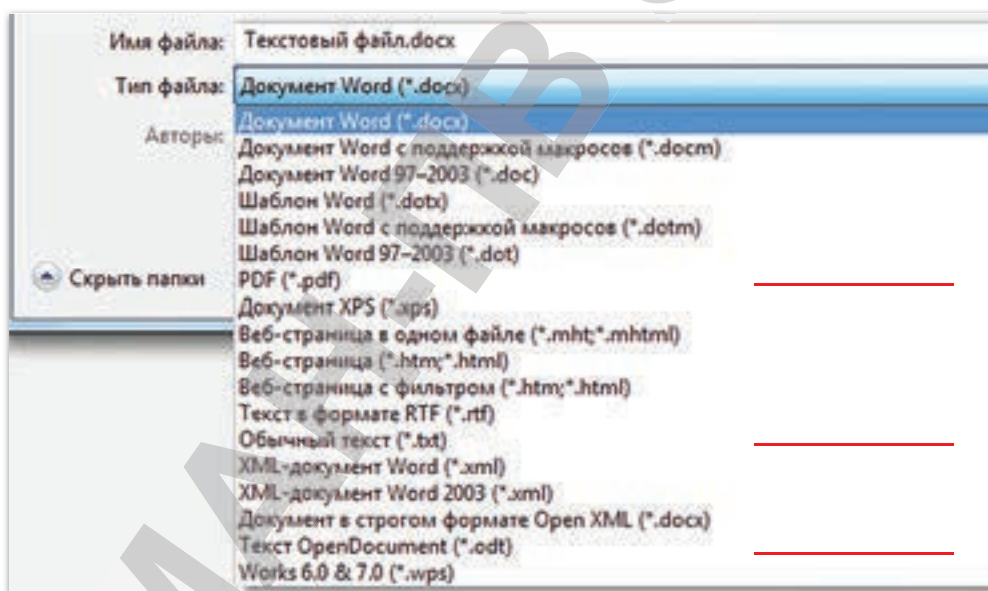


Рис. 17. Способ сохранения одной информации в разных форматах в одном файле

3. Выберите текстовые файлы на рабочем столе, потом при помощи правой кнопки мыши вызовите контекстное меню и откройте команду «Свойства» (рис. 18–19).

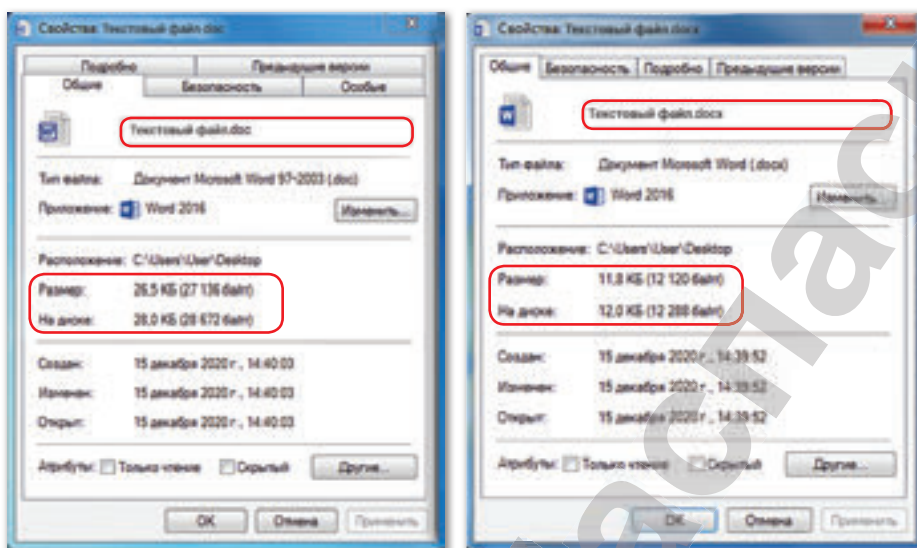


Рис. 18. Размеры файлов в форматах «.doc» и «.docx» с одинаковой информацией

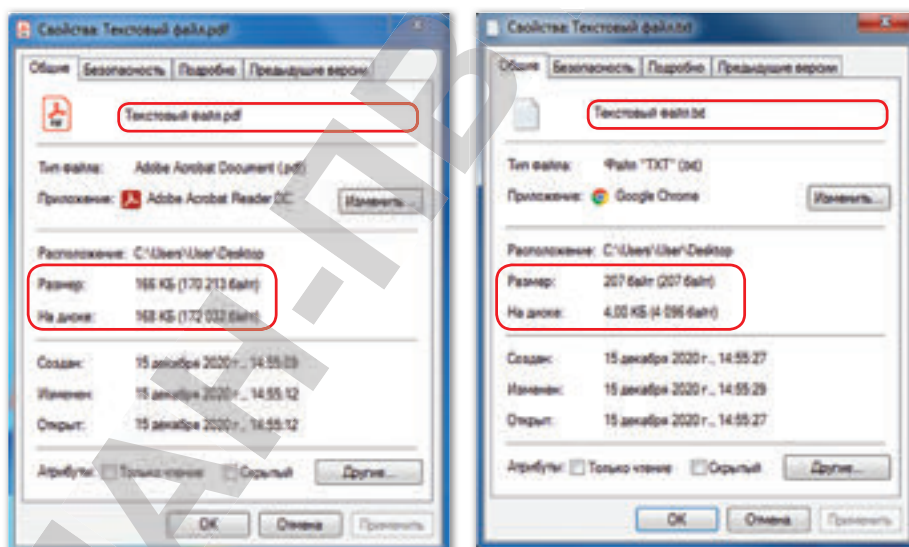


Рис. 19. Размеры файлов в форматах «.pdf» и «.txt» с одинаковой информацией

Вы научились сравнивать размеры файлов разных форматов, которые хранят одинаковую информацию. В зависимости от единиц измерения, можно определить, какой тип файла более эффективен.

1

Отвечаем на вопросы

1. Что такое файл?
2. Как описывается имя и расширение файла?
3. Что относится к основным свойствам файла?
4. Как просмотреть основные свойства файла?
5. Что такое тип файла?
6. Сколько типов файловой системы вы знаете?
7. Что такое размер файла?

2

Думаем и обсуждаем

1. Для чего нужно знать размер файла?
2. Как сохранить файлы с одинаковой информацией в разных форматах и как это влияет на их размер?
3. Как вы думаете, почему информация, записанная на компьютер, не смешивается друг с другом?

3

Анализируем и сравниваем

Проанализируйте и сравните файловые системы:



4

Выполняем в тетради

1. Заполните таблицу:

Типы файлов	Расширения
	.mp3, .aif, .wav, .wma, .mpa
Табличный файл	
	.7z, .rar, .arj, .pkg, .tar, .gz
Текстовый файл	
	.pps, .ppt, .pptx, .key, .odp
Программный файл	
	.gif, .png, .jpg, .tiff, .bmp
Видеофайл	
	.drv, .sys, .cab, .msi, .bak, .lnk, .cpl

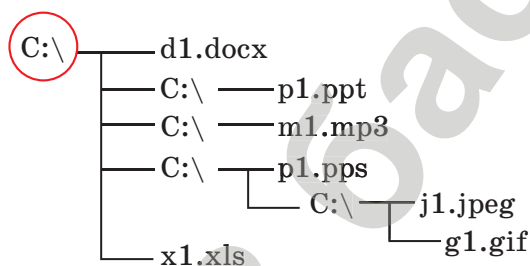
2. Приведите в соответствие:

«Реферат»	Расширение файла
«.ppt»	Имя файла

5

Выполняем на компьютере

1. Обратите внимание на следующую иерархическую файловую систему. Создайте на рабочем столе папки и типы файлов по цепям иерархической файловой системы и из списка «Свойства» определите их размер:



2. Сравните размеры файлов разных форматов, в которых хранится одинаковая информация. Рассмотрите расширения презентационных файлов. В качестве одинаковой информации необходимо составить 5–10 предложений на тему «Моя школа». Откройте файловое расширение «.ppt» и сохраните презентацию в формате нескольких презентационных файлов. Сравните размеры сохраненных файлов.

6

Делимся мыслями

Можно ли использовать в повседневной жизни полученные знания по типам и расширениям файлов?

7

Домашнее задание

Дополните предложения:

1. Файл – 2. Расширение файла – 3. Основные свойства файла – 4. Тип файла – 5. Размер файла –

§ 5. Размер файлов. Создание и открытие архивов различных форматов

Вспомните!

- Что такое файл?
- Что относится к основным свойствам файла?
- Что такое тип файла?
- На сколько видов делятся файловые системы?
- Что такое размер файла?
- Изменяются ли размеры при хранении одинаковой информации в файлах разных форматов, какие различия существуют?

Вы узнаете:

- о сжатии информации;
- об архивных файлах;
- что такое архив;
- о степени сжатия;
- как разархивировать файлы;
- о списках программ архивации;
- как добавить новый файл в архив.

Архив – *Архив* – Archive

Сжатие информации – *Ақпараттарды сығу* – Data compression

Степень сжатия – *Сығу дәрежесі* – Compression ratio

Архивный файл – *Архив файлы* – Archive file

Разархивировать – *Архивтан шығару* – Unzip

Наиболее распространенным типом системных программ являются **программы архивации**, основной задачей которых является сохранение и сжатие объема файла без изменения.

Сжатие информации (англ. *data compression*) – процесс, в результате которого в несколько раз уменьшается размер файлов, что позволяет сэкономить память компьютера. Сжатие информации в файлах производится за счет устранения избыточности различными способами, например, за счет упрощения кодов, исключения из них постоянных битов или представления повторяющихся символов или повторяющейся последовательности символов в виде коэффициента повторения и соответствующих символов. То есть удаляет пустые символы (белый цвет) и кодирует с объединением повторяющиеся символы (рис. 20).

Исходная информация



Сжатая информация

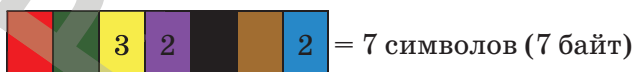


Рис. 20. Способ сжатия информации

Тип сжатия одного или нескольких файлов специальным способом называется **архивированным файлом**, или **архивом**. Например, .7z, .rar, .arj, .pkg, .tar, .GZ включают в себя архивированные расширения файлов.

Целью архивирования файлов является компактное уплотнение информации из памяти диска, длительное хранение, экономия и снижение стоимости передачи информации по каналам связи в компьютерных сетях, восстановление объема архивированных файлов. Кроме того, архивированные файлы защищают от несанкционированного доступа к информации, а также от компьютерных вирусов при копировании.

Под *степенью сжатия* понимается процентное соотношение сжатого и исходного файлов. Степень сжатия зависит от программ для архивации, методов сжатия и типа файлов. Наиболее хорошо сжимаются файлы графических образов, текстовые файлы и файлы данных, для которых степень сжатия может достигать 5–40%.

Извлечение из архива – процесс восстановления файлов из архива до первоначального состояния до загрузки в архив. Файлы, извлеченные из архива, помещаются на диск или в оперативную память. Архивация и извлечение файлов осуществляются с помощью специальных программ, которые называются **программами для архивации**. В настоящее время используются несколько десятков программ для архивации, которые отличаются своими свойствами и параметрами работы, но лучшие из них имеют одинаковые характеристики. Самые распространенные программы для архивации файлов: WinZip, WinRAR, 7Zip, WinAce, Arj (и его виды), PowerArchiver и др. (рис. 21).



Рис. 21. Основные программы-архиваторы

Чтобы заархивировать файлы разных форматов, необходимо выбрать одну из программ для архивации. Для этого ознакомимся с самой популярной программой – WinRAR.

WinRAR – это программа для архивации файлов для 32-разрядных и 64-разрядных операционных систем Windows.

Способы запуска программы-архиватора WinRAR:

1. Выполните команду: Пуск ⇒ Все программы ⇒ WinRAR.
2. Щелкните правой кнопкой мыши на любом месте рабочего стола и выберите Архив WinRAR из строки «Создать» (рис. 22).

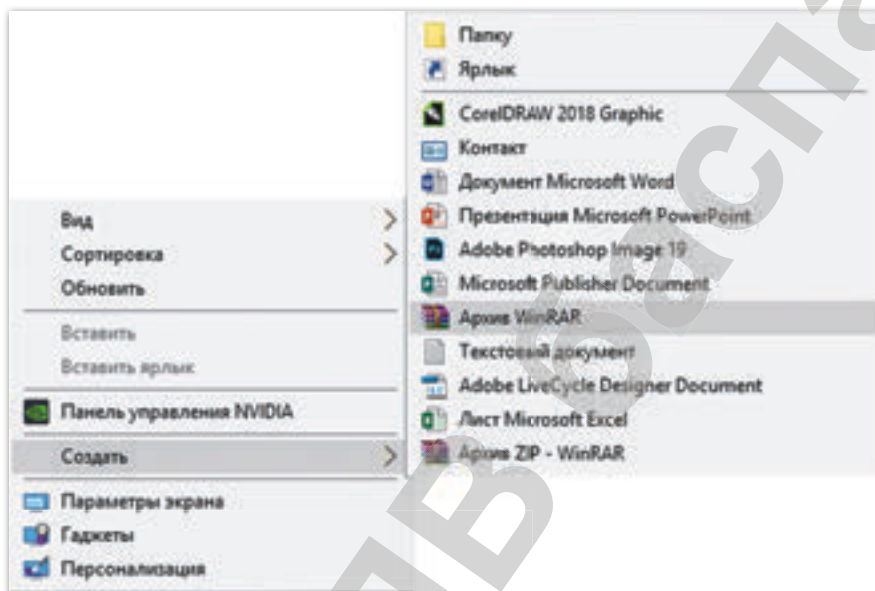


Рис. 22. Строка Создать

Познакомимся с элементами созданного окна архивации WinRAR (рис. 23) и с функциями панели инструментов (табл. 2):

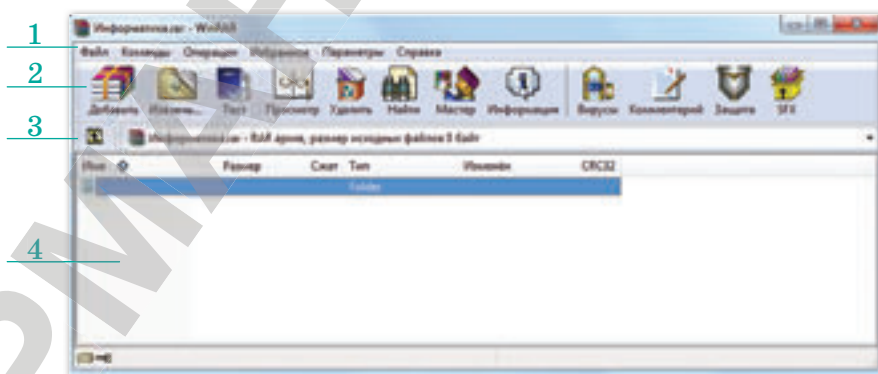











Рис. 23. Элементы окна WinRAR: 1 – строка меню; 2 – панель инструментов; 3 – адресная строка; 4 – окно папок и файлов

Таблица 2. Функции элементов панели инструментов

	Добавление нового файла в архив		Удаление файла из архива
	Извлечение файла из архива		Поиск архивированного файла
	Проверка файла в архиве		Вызов мастера для начинающих
	Просмотр содержимого архива		Информация о файле в архиве

Способы архивирования файлов различных форматов на компьютере в программе WinRAR

1. Нажмите кнопку Пуск и выберите программу WinRAR из списка «Все программы». Выполните действие «Добавить»  на панели инструментов или из списка «Команды» и выберите «Добавить файлы в архив». Достаточно указать адрес архивируемого файла и нажать кнопку «ОК» (рис. 24).

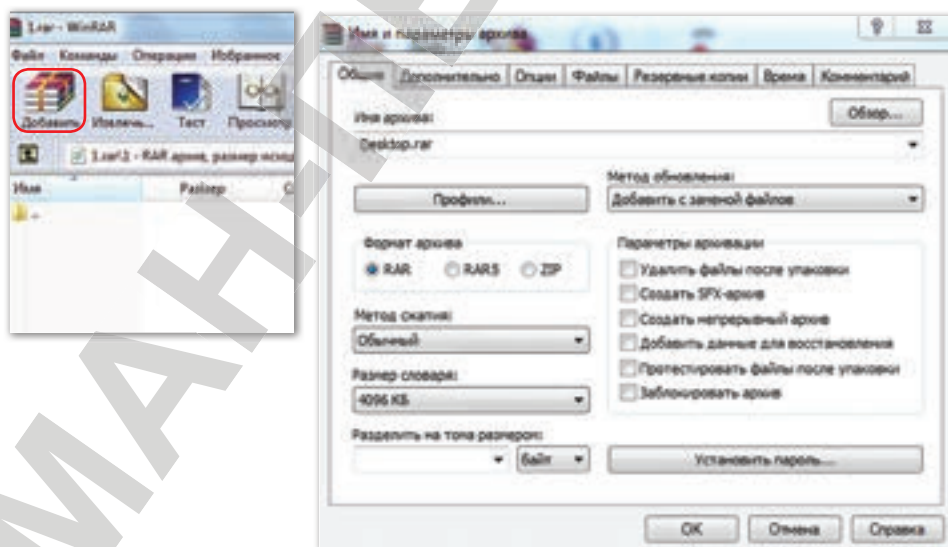


Рис. 24. Добавление нового файла в архив из панели инструментов

2. Выделите архивируемый файл, щелкните правой кнопкой мыши, из появившегося списка выберите «Добавить

в архив...» и в появившемся окне «Имя и параметры архива» нажмите на кнопку «ОК» (рис. 25).

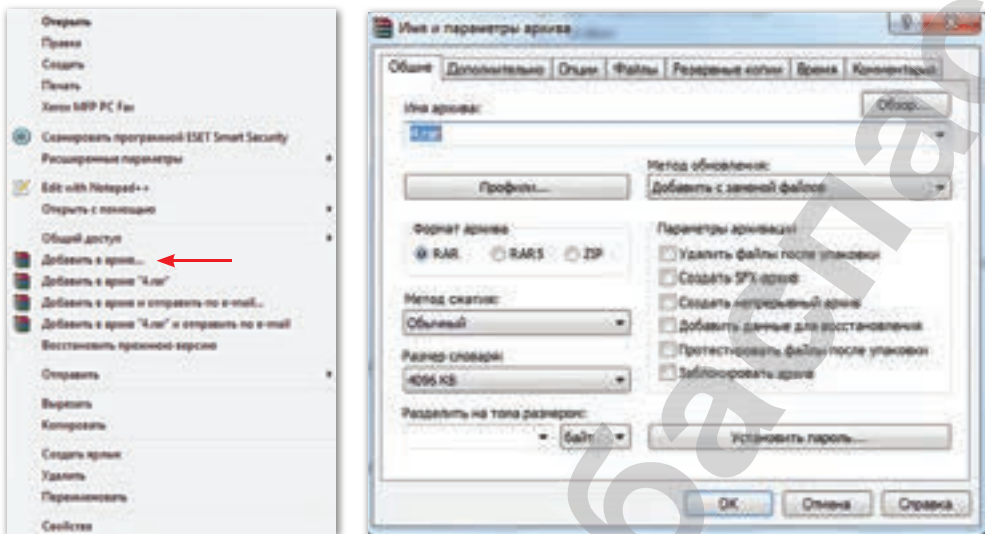


Рис. 25. Добавление нового файла в архив через контекстное меню

- Понаблюдайте, есть ли разница между исходным размером файлов разных форматов и размером после их архивирования (рис. 26).

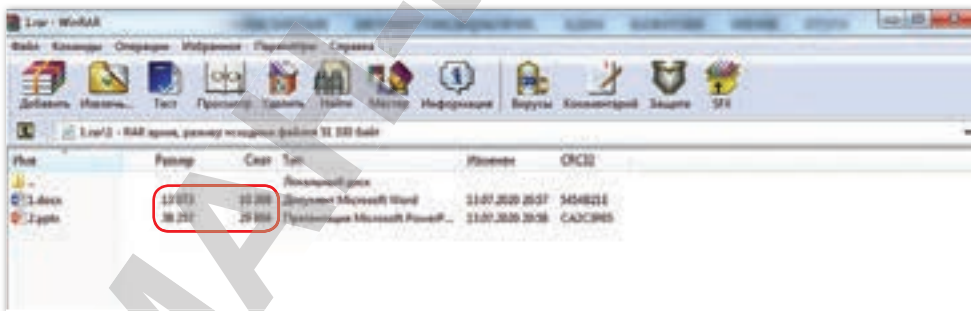



Рис. 26. Исходный размер файлов и размер после архивирования

Извлечение файла из архива WinRar

Выделите файл, который нужно извлечь из архива, и выберите на панели инструментов инструмент «Извлечь файл из архива с выбором папки назначения»  или из списка

«Команда» из списка меню «Извлечь в указанную папку». Нажмите на кнопку «ОК» с указанием места хранения извлекаемого из архива файла (рис. 27).

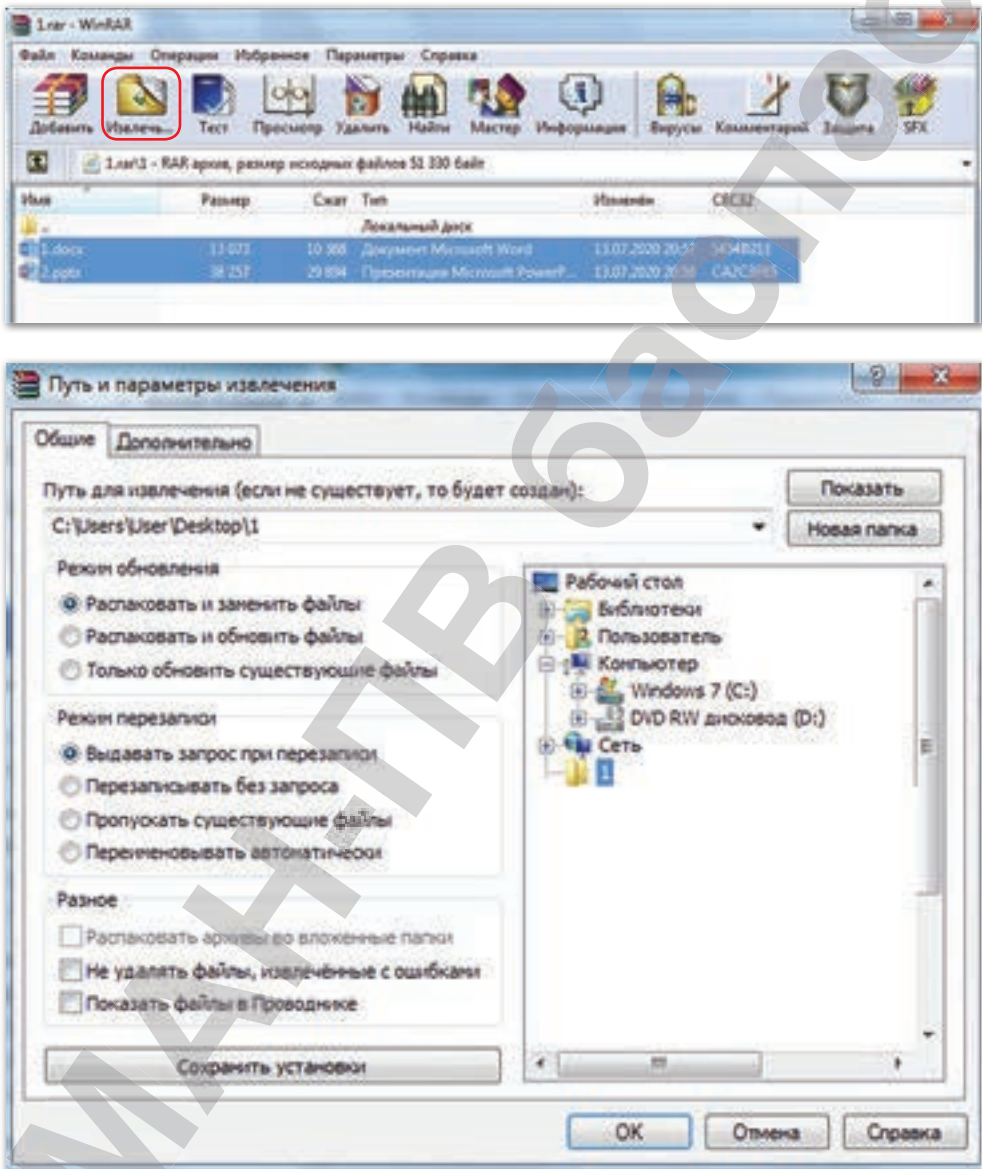


Рис. 27. Извлечение файла из архива

Как уже было сказано выше, файл из архива возвращается в исходный размер. Благодаря архивированию мы можем значительно сэкономить память компьютера.

1

Отвечаем на вопросы

1. Что такое сжатие информации?
2. Какое расширение у архивированного файла?
3. Что такое степень сжатия?
4. Что такое извлечение из архива?
5. Какие программы используются для создания и восстановления архивов?
6. Как добавить новый файл в архив?
7. Какие способы добавления файла в архив вы знаете?

2

Думаем и обсуждаем

1. Для чего мы сжимаем информацию?
2. В чем необходимость архивирования файлов?
3. Что можно сэкономить с помощью архивирования файлов?
4. Можно ли уменьшить количество файлов или передаваемых файлов, сохраненных на компьютере?

3

Анализируем и сравниваем

Разделитесь на группы. Используя дополнительные источники информации, проанализируйте приведенные ниже программы-архиваторы. Проведите сопоставление преимуществ и недостатков.









Программы-архиваторы	Описание	Преимущества	Недостатки
			
			
			

Программы-архиваторы	Описание	Преимущества	Недостатки
			
			

4

Выполняем в тетради

Запишите в тетрадях функции приведенных ниже инструментов:

Инструменты	Функции	Инструменты	Функции
			
			
			
			

5

Выполняем на компьютере

- По описанию простой файловой системы, приведенной ниже, создайте папку и типы файлов на рабочем столе с названием «Школа». Добавьте в архив созданные файлы:

```

C:\
├── Школа.docx
├── Школа.mp3
├── Школа.jpeg
├── Школа.xls
└── Школа.gif
  
```

2. Сравните размеры добавленного в задании 1 архивированного файла и исходного файла:

Файлы	Размер исходного файла	Размер архивированного файла
Школа.docx		
Школа.mp3		
Школа.jpeg		
Школа.xls		
Школа.gif		

3. Извлеките из архива созданный ранее файл под названием «Школа». Проверьте, вернулся ли размер к исходному.

6

Делимся мыслями

1. Почему размер исходного файла и размер архивированного файла не одинаковые?
2. Какими программами-архиваторами вы пользуетесь на своем компьютере? Почему?

7

Домашнее задание

Составьте алгоритм перевода по данной внизу исходной информации на сжатую (ячейка белого цвета – пробел):

Исходная информация, символ (байт) – ?



Сжатая информация, символ (байт) – ?

ЗАДАНИЯ ПО СУММАТИВНОМУ ОЦЕНИВАНИЮ К РАЗДЕЛУ 1

1. **Наименьшей единицей измерения информации является:**
 - А. Байт
 - В. Бит
 - С. Килобайт
 - Д. Мегабайт
 - Е. Число
2. **Расположите в порядке возрастания:**
 Эксабайт, Мегабайт, Килобайт, Бит, Йоттабайт, Зеттабайт, байт, Гигабайт, Терабайт, Петабайт
3. **Заполните пропуски:**
 Компьютер имеет ... и ... память.
4. **Переведите из одной единицы измерения в другую следующую информацию:**

128 бит = ... байт

3 Тбайт = ... бит

17 Эбайт = ... Збайт

140 байт = ... бит

1612540 Кбайт = ... Гбайт

800 бит = ... байт

2,8 Мбайт = ... байт

22,5 Гбайт = ... Кбайт

68 байт = ... бит

256 Мбайт = ... байт

5. **Установите соответствие:**

1	
2	
3	
4	
5	
6	
7	

А	Жесткие диски
В	Флеш-память
С	Гибкие диски
Д	Оперативная память
Е	CD-R диски
Ф	Постоянное запоминающее устройство
Г	Кэш-память

ИТОГИ 1 РАЗДЕЛА

В разделе «Измерение информации и компьютерная память» вы подробно разобрались в единицах измерения информации, узнали о переводе информации из одной единицы измерения в другую, назначении памяти компьютера и ее видов (оперативное запоминающее устройство, постоянное запоминающее устройство, внешняя память, кэш-память), сравнивали объемы файлов разных форматов, хранящих одну и ту же информацию, узнали о способах создания и открытия архивов разных форматов.

Также были рассмотрены практикумы по определению единиц измерения информации, используемой в повседневной жизни и переводу из одной единицы измерения в другую. Вы изучили типы и цели работы памяти компьютера, размеры файлов, примеры сравнения размеров файлов разных форматов и способы архивирования и извлечения из архива информации разных форматов.

Мы надеемся, что знания, полученные из этого раздела, вы будете использовать в обычной жизни или в будущем в соответствии с нужными целями.

Раздел 2

СЕТИ И БЕЗОПАСНОСТЬ

Изучив раздел, вы научитесь:

- классифицировать компьютерные сети;
- защищать компьютер от вредоносных программ;
- оценивать эргономичность пользовательских интерфейсов.

§ 6. Компьютерные сети и их классификация

Вспомните!

- Чему равна наименьшая единица измерения информации?
- Сколько битов в 1 байте?
- На сколько видов делится компьютерная память?
- Что такое файл?
- Что такое сжатие информации?
- Какие программы-архиваторы вы знаете?

Вы узнаете:

- о компьютерных сетях;
- о классификации компьютерных сетей.

Компьютерная сеть –
Компьютерлік желі – Computer network

Локальная сеть – *Жергілікті желі* – Local area network

Региональная сеть – *Аймақтық желі* – Wide area network

Глобальная сеть – *Ауқымды желі* – Global area network

Сетевая топология – *Желілік топология* – Network topology

С появлением компьютеров возник вопрос о передаче данных между персональными компьютерами и рациональном распределении компьютерных ресурсов.

С развитием аппаратной и программной базы компьютеров начали совершенствоваться и сетевые технологии. Изначально системы передачи данных создавались в коммерческих, военных и научных целях.

В настоящее время использование компьютерных сетей остается неотъемлемой частью нашей жизни и охватывает все сферы человеческой деятельности.

Что такое компьютерная сеть?

Компьютерная сеть (англ. *computer network*) – разветвленная система обработки информации, предназначенная для общения с помощью средств связи двух или более компьютеров, которые могут обмениваться данными друг с другом.

В зависимости от особенностей распространения компьютерные сети классифицируются на 4 вида (*схема 6*).

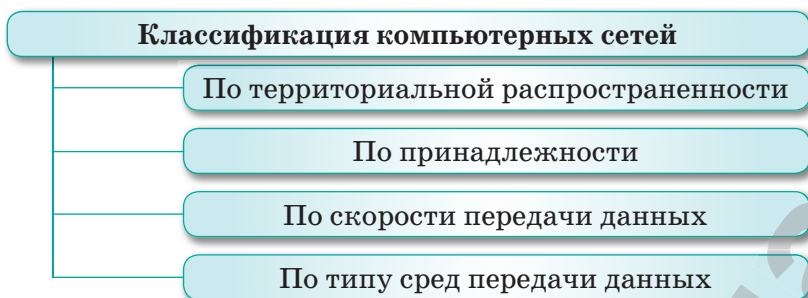


Схема 6. Классификация компьютерных сетей

По территориальной распространности, в зависимости от области, которую они охватывают, компьютерные сети делятся на локальные, региональные и глобальные.

Локальная сеть (LAN – Local Area Network) – объединение нескольких компьютеров, которые должны использовать обмен данными или общие данные только в ограниченной зоне. Например, интеграция считываемых компьютеров в одном помещении, здании или в комплексе близлежащих зданий (рис. 28).

Региональная сеть (WAN – Wide Area Network) находится в городе или регионе. Например, обмен информацией между банками в одном городе (рис. 29).

Глобальная сеть (GAN – Global area network) – это любая сеть связи, которая охватывает весь мир. Позволяет подключить любой компьютер независимо от его географического положения. Например, всемирная паутина Internet (рис. 30).

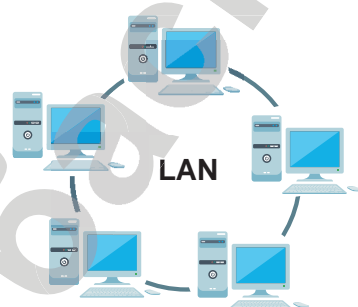


Рис. 28. Локальная сеть

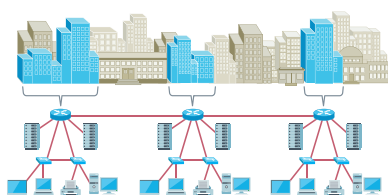


Рис. 29. Региональная сеть

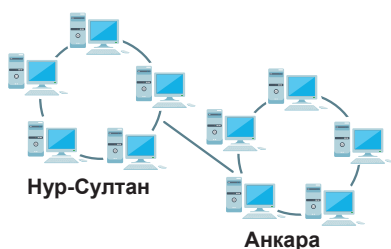


Рис. 30. Глобальная сеть

По принадлежности различают ведомственные и государственные сети.

- Ведомственные сети принадлежат к одной организации и располагаются на ее территории. Например, цирковые, театральные кассы и другие.
- Государственные сети – это сети государственных органов и учреждений. Например, Министерства информации и общественного развития РК, здравоохранения и другие.

По скорости передачи данных компьютерные сети делятся на:

- низкоскоростные (до 10 Мбит/с);
- среднескоростные (до 100 Мбит/с);
- высокоскоростные (свыше 100 Мбит/с).

Единица измерения скорости передачи данных определяется по количеству бит в 1 секунду (1 бит/с).

По типу сред передачи данных компьютерные сети делятся на:

- проводные (коаксиальные, витая пара, оптические);
- беспроводные (радиосвязь, Wi-Fi, WiMAX) (рис. 31).



Рис. 31. Среды передачи данных

Одной из характеристик локальной вычислительной сети является **топология (или архитектура) сети**. Топология сети – геометрическая форма и физическое расположение компьютеров по отношению друг к другу. Наиболее распространенные типы сетевых топологий:

Топология шина – это тип, в котором несколько компьютеров объединены в сеть с помощью одного кабельного соединения (рис. 32). Простая сеть соединена коаксиальными кабелями. В случае сбоя одного компьютера в кабельном соединении все компьютеры остаются без связи.

Топология кольцо – это простой тип, в котором сетевой сигнал распространяется в одном направлении по кругу (рис. 33). Если один из компьютеров выходит из строя на канале связи, сеть перестает работать.

Топология звезда – это тип, в котором все компьютеры в сети подключены к центральному компьютеру или концентратору (рис. 34). Подключить новый компьютер к сети такого типа легко, и в случае выхода из строя одного компьютера в сети это не влияет на другие и не прерывает соединение.

Общая топология шины – это процесс обмена информацией через большой канал связи, называемый общественной шиной (рис. 35).

Топология дерево – звездообразная связь (рис. 36), каждый узел которой высокого уровня связан с узлом нижнего уровня. Это топологию также называют иерархической звездой. К примеру, если организация является

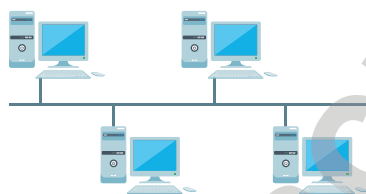


Рис. 32. Топология шина



Рис. 33. Топология кольцо

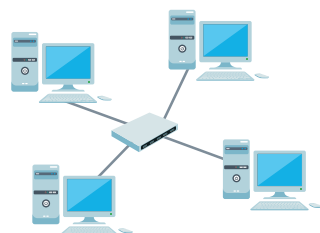


Рис. 34. Топология звезда



Рис. 35. Общая топология шины

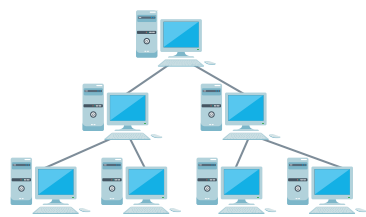


Рис. 36. Топология дерево

многоуровневой, у нее будет общий узел, который соединяет отдельные узлы каждого этажа.

1

Отвечаем на вопросы

1. Что такое компьютерная сеть?
2. Как классифицируются компьютерные сети?
3. Что такое локальная сеть?
4. Что такое региональная сеть?
5. Что такое глобальная сеть?
6. На какие типы делятся сети по ведомственной принадлежности?
7. На какие виды делятся сети по средам передачи данных?
8. Что такое топология?
9. Какие типы топологии сетей вы знаете?

2

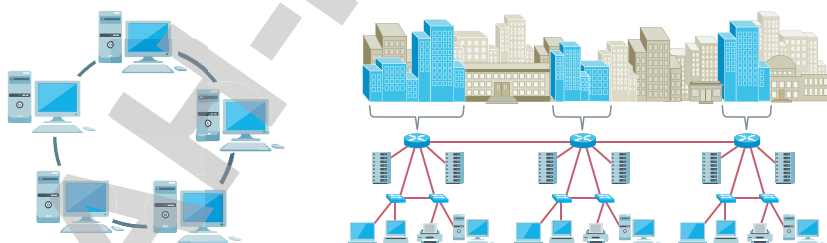
Думаем и обсуждаем

1. Как происходит обмен информацией между компьютерами?
2. В чем необходимость использования компьютерных сетей?
3. Как можно объяснить важность компьютерных сетей?
4. В чем разница между топологиями звезда и шина?

3

Анализируем и сравниваем

Проанализируйте и сравните преимущества и недостатки компьютерных сетей по территориальной распространенности:



4

Выполняем в тетради

Напишите в тетрадях, на какие типы классифицируются компьютерные сети и опишите каждый из них:



5

Выполняем на компьютере

1. Узнайте и определите, по какой топологии сети расположены компьютеры в вашем кабинете информатики.
2. С помощью Интернета узнайте, какие сетевые компоненты необходимы для построения компьютерной сети.

6

Делимся мыслями

Что нового вы узнали на уроке? Поделитесь своими мыслями с одноклассниками. Каким образом можно использовать в повседневной жизни полученные на уроке знания и навыки? Приведите примеры.

7

Домашнее задание

1. Алмас и Айбек работают в одном городе, но в разных отделениях одного банка. Они обмениваются деловыми письмами через Интернет. Как вы думаете, к какому типу распространения принадлежит эта сеть? Объясните почему и запишите свои мысли в блокнот.
2. Медет учится в Кембриджском университете в Великобритании. Однако осенние экзамены он сдал онлайн из города Алматы. Какой из типов распространения сети был использован в данном случае? Почему?

§ 7. Антивирусная безопасность

Вспомните!

- Что такое компьютерная сеть?
- Как классифицируются компьютерные сети?

Вы узнаете:

- о компьютерных вирусах;
- об обнаружении компьютерных вирусов;
- о классификации компьютерных вирусов;
- о методах защиты от компьютерных вирусов;
- об антивирусных программах.

Вирус – *Virus* – Virus

Компьютерный вирус –
Компьютерлік вирус –
Computer virus

Антивирусная программа –
Антивирустық программа –
Antivirus program

Антивирусная безопасность – *Антивирустық қауіпсіздік* – Antivirus security

Среди многих компьютерных программ имеются специально разработанные программы, которые вызывают повреждение компьютеров. Эти программы называются **компьютерными вирусами**.

Что такое компьютерные вирусы?

Компьютерный вирус – специальная программа, которая обеспечивает скрытый доступ к компьютеру (коду программ, памяти системы, секторам загрузки), блокирует работу компьютера или наносит вред данным и системам работы. Основной целью этой программы является ее распространение и внедрение в систему. Также она выполняет основные функции, связанные с нарушением работы программно-аппаратных комплексов, удалением файлов, даже удалением операционной системы, приведением в непригодность структуры размещения данных, блокированием работы пользователя и др. Компьютерные вирусы, как правило, находятся в памяти скрытно и питаются ресурсами системы.

Появление вируса на компьютере можно наблюдать с помощью следующих признаков:

- замедление работы компьютера, частые зависания или сбои;
- невозможность загрузить операционную систему;
- повреждение, изменение или полная потеря файловой системы;
- увеличение количества файлов на диске и уменьшение объема оперативной памяти;
- появление нежелательных сообщений на экране монитора.

В настоящее время существует большое количество компьютерных вирусов, которые можно классифицировать по следующим признакам (табл. 3).

Таблица 3. Классификация компьютерных вирусов

По способам заражения	По среде обитания	По алгоритму работы вирусов
Резидентные	Файловые вирусы	Паразитические
	Загрузочные вирусы	Вирусы-черви
		Вирусы-невидимки
Нерезидентные	Макровирусы	Вирусы-мутанты
	Сетевые вирусы	Троянские

Классификация по способам заражения:

- **Резидентные** – вирусы, которые остаются в памяти компьютера в течение длительного времени. Вирус этого типа проникает в запоминающее устройство и контролирует действия пользователя, повреждает вызванную программу.
- **Нерезидентные** – вирусы, которые не заражают память компьютера и являются активными лишь ограниченное время. Вирус этого типа ищет неповрежденные программы и записывается на них, затем запускается при вызове поврежденных программ.

Классификация по среде обитания:

Файловые вирусы используют файловую систему, заражают файлы, которые запускаются на любой ОС.

Загрузочные вирусы – вирусы, которые внедряются в загрузочный сектор диска или в сектор, содержащий системный загрузчик винчестера.

Макровирусы – это тип макроса, который встроен в пакеты прикладных программ, такие как Microsoft Office. Большинство из этих вирусов написаны для MS Word.

Сетевые вирусы – вирусы, которые распространяются в компьютерной сети, не изменяя файлы или сектора на дисках.

Классификация по алгоритму работы вирусов:

Паразитические вирусы – это файловые вирусы, изменяющие содержимое файла, добавляя в него свой код. Файл остается

непригодным для переноса. Данные вирусы быстро обнаруживаются и удаляются.

Вирусы-черви распространяются по компьютерным сетям, создают свои копии по сетевым адресам компьютера.

Вирусы-невидимки трудно обнаруживать и обезвреживать, так как они перехватывают обращение операционной системы к пораженным файлам и секторам и подменяют их незараженными участками.

Вирусы-мутанты состоят из алгоритмов шифрования-дешифрования, копии этих вирусов не содержат повторяющихся частей и их трудно обнаружить.

Троянский вирус (троянский конь) – вредоносный вид, основное отличие которого от классического вируса состоит в методе распространения: обычно он проникает в систему под видом обычной, легитимной программы.

Методы защиты от компьютерных вирусов:

1. Устанавливайте на компьютер только лицензионное ПО.
2. Используйте функцию для блокировки всплывающих окон в браузере.
3. Установите антивирусное программное обеспечение для защиты от вирусов.

Антивирусная программа (или антивирусное средство защиты, средство обнаружения вредоносных программ, антивирус) – специальная программа, предназначенная для выявления компьютерных вирусов, а также нежелательных (считающихся вредоносными) программ, восстановления зараженных (модифицированных) файлов и предотвращения заражения файлов или операционной системы вредоносным кодом.

Для обнаружения, удаления и защиты от компьютерных вирусов разработано несколько видов антивирусных программ:

- программы-детекторы;
- программы-доктора, или фаги;
- программы-ревизоры (инспектора);
- программы-фильтры (мониторы);
- программы-вакцины, или иммунизаторы.

Программы-детекторы осуществляют поиск характерных для конкретных вирусов сигналов в оперативной памяти и в файлах, и при обнаружении выдают соответствующее сообщение. Недостатком таких антивирусных программ является

то, что они могут находить только те вирусы, которые известны разработчикам таких программ (рис. 37).



Рис. 37. Программы-детекторы

Программы-доктора, или фаги, а также программы-вакцины не только находят зараженные вирусами файлы, но и «лечат» их, то есть удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние. В начале своей работы фаги ищут вирусы в оперативной памяти, уничтожая их, и только затем переходят к «лечению» файлов. Среди фагов выделяют полифаги, то есть программы-доктора, предназначенные для поиска и уничтожения большого количества вирусов. Наиболее популярными из них являются: Kaspersky Anti-Virus, Norton ANTIVIRUS, Dr.Web (рис. 38). Из-за появления новых вирусов программы-детекторы и программы-доктора быстро устаревают, поэтому их необходимо постоянно обновлять.



Рис. 38. Программы-доктора, или фаги

Программы-ревизоры относятся к самым надежным средствам защиты от вирусов. Они проверяют данные на диске на предмет вирусов-невидимок, изучают, не проник ли вирус в файлы, нет ли посторонних в загрузочном секторе жесткого диска, несанкционированных изменений реестра Windows. Как правило, сравнение с исходным состоянием включается

сразу после загрузки операционной системы. Обнаруженные изменения отображаются на экране монитора. Программы проверяют длину файла, код циклического отслеживания (количество отслеживаемых файлов), дату, время изменения и другие параметры во время сравнения. Одной из самых популярных программ-ревизоров является Kaspersky Monitor.

Программы-фильтры («сторожа») представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов. Такими действиями могут являться: попытки коррекции файлов с расширениями COM, EXE; изменение атрибутов файла; прямая запись на диск по абсолютному адресу; запись в загрузочные сектора диска; загрузка резидентной программы. В случае обнаружения подозрительных действий, характерных для каких-либо вирусов, программа направляет пользователю сообщение, предлагающее запретить вирусы. Программы-фильтры очень полезны, так как заранее предупреждают о вирусных атаках. Правда файлы и диски они не лечат. Для лечения вирусов используются другие антивирусные программы.

Вакцины, или иммунизаторы – это резидентные программы, предотвращающие заражение файлов. В настоящее время программы-вакцины почти не используются.

1

Отвечаем на вопросы

1. Что такое компьютерные вирусы?
2. Как обнаружить компьютерные вирусы?
3. Как классифицируются компьютерные вирусы?
4. Как защитить компьютер от вирусов?
5. Что такое антивирусные программы?
6. На какие типы делятся антивирусные программы?

2

Думаем и обсуждаем

1. Откуда берутся компьютерные вирусы?
2. Почему вирусы опасны для компьютеров?
3. Почему вирус классифицируется по алгоритму?
4. Для чего используется антивирусная программа?
5. Можно ли работать с компьютером без антивирусной программы?

3

Анализируем и сравниваем

Проанализируйте и сравните различия между типами антивирусных программ:

Программы-детекторы	↔	Программы-ревизоры
Программы-доктора, или фаги	↔	Программы-фильтры
Вакцины, или иммунизаторы	↔	Программы-детекторы

4

Выполняем в тетради

1. Напишите классификацию вирусов по среде обитания.
2. Закончите предложения:
 - Паразитические вирусы – ...
 - Вирусы-черви – ...
 - Вирусы-невидимки – ...
 - Вирусы-мутанты – ...
 - Троянские вирусы (троянские кони) – ...

5

Выполняем на компьютере

1. Используя интернет-ресурсы, заполните таблицу типами дополнительных антивирусных программ:

№	Наименование программы	Особые функции	Алгоритм удаления вируса
1			

2. Узнайте, какие антивирусные программы установлены на компьютерах в вашем кабинете информатики. Дайте дополнительные комментарии к этим программам.

6

Делимся мыслями

Сможете ли вы применить ваши знания в области информационной безопасности в будущем для решения повседневных задач?

7

Домашнее задание

Опишите классификацию вирусов по способам заражения.

§ 8. Пользовательский интерфейс

Вспомните!

- Что такое компьютерные вирусы?
- Как обнаружить компьютерные вирусы?
- На какие типы классифицируются компьютерные вирусы?
- Как защитить компьютер от вирусов?
- Что такое антивирусные программы?
- Какие типы антивирусных программ вы знаете?

Пользовательский интерфейс – Пайдаланушы интерфейсі – User interface

Командный интерфейс – Командалық интерфейс – Command interface

Графический интерфейс – Графикалық интерфейс – Graphical interface

Вы узнаете:

- что такое пользовательский интерфейс;
- что такое командный интерфейс;
- что такое графический интерфейс.

С понятием «эргономика», как науки, изучающей взаимодействие человека и других элементов системы, и сферы деятельности по применению этой

науки для обеспечения благоприятных условий труда на рабочем месте, а также правилами техники безопасности при работе с компьютером, действиями по оказанию первой медицинской помощи, вы знакомились в 6 классе. В этом году вы будете изучать правила взаимодействия человека и компьютера, типы пользовательских интерфейсов.

Пользовательский интерфейс – это совокупность средств и правил взаимодействия человека и компьютера.

По мере развития и совершенствования компьютерной техники разрабатывались различные варианты **пользовательских интерфейсов** (схема 7).

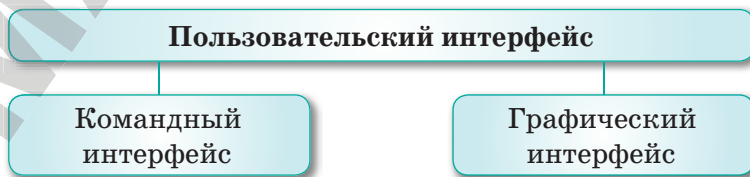


Схема 7. Виды пользовательского интерфейса

Взаимодействие с компьютером через его команды называется **командным интерфейсом**.

На компьютерах, оперировавших только числами и символами, был реализован командный интерфейс (рис. 39).

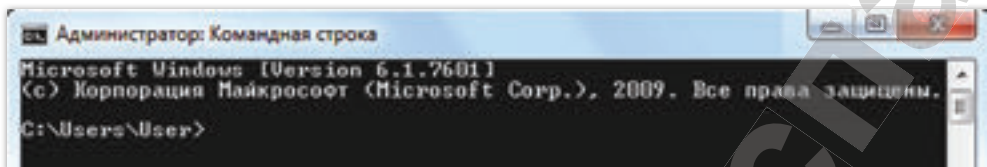


Рис. 39. Командная строка

Алгоритм работы командного интерфейса:

- команда подается с помощью последовательности символов (командной строки);
- компьютер сопоставляет поступившую команду с имеющимся в его памяти набором команд;
- выполняется действие, соответствующее поступившей команде.

Чтобы взаимодействовать с компьютером, человеку необходимо знать множество команд. Поэтому для управления устройствами и оперирования данными существуют специальные команды (табл. 4).

Таблица 4. Команды Windows

Команды	Описание
HELP	Информационно-справочная помощь Windows
DATE	Показать или изменить текущую дату
DIR	Показать список файлов и каталогов
...	...

Попробуем запустить команду информационно-справочной помощи Windows. Для этого:

1. Нажмите кнопку «Пуск» и выберите «Командная строка» из списка «Стандартные».
2. В появившемся окне введите «HELP».

3. В командной строке появится окно HELP (рис. 40). Аналогично все действия в компьютерной системе выполняются путем ввода специальных команд.

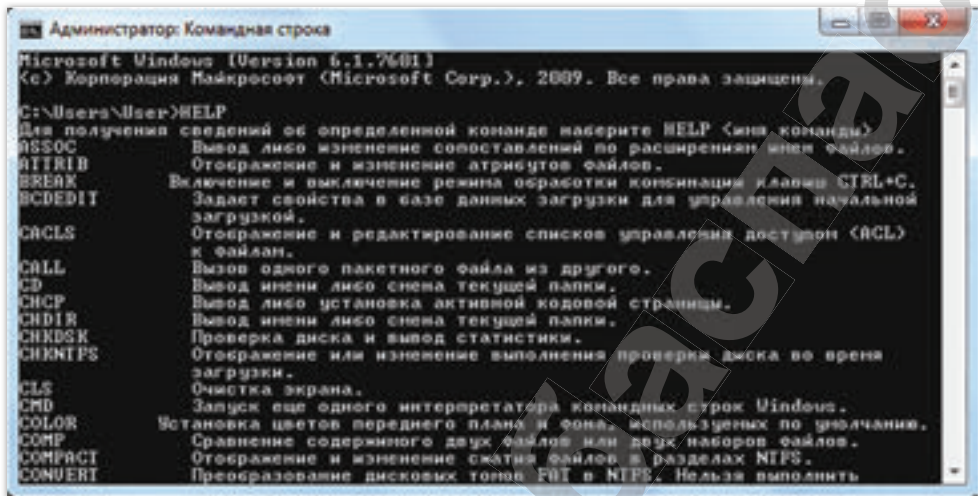


Рис. 40. Команда HELP

Работа с режимом командной строки обеспечивает минимальный расход памяти, к тому же, в современном программном обеспечении имеется большое число команд, многие из которых используются крайне редко. Набор такой команды в командной строке осуществляется гораздо быстрее, чем, например, навигация по меню.

Необходимость запоминать многочисленные команды отпала с появлением **графических интерфейсов**. Они обеспечивали возможность выполнять некоторые команды с помощью клавиш или манипулятора «мышь», которая является основным элементом аппаратного управления компьютером. Манипулятор отображается на экране в виде указателя – небольшого графического объекта, который перемещается по экрану при движении мыши. Он указывает на различные объекты, запускает программы, выбирает пункты меню, перетаскивает объекты и т.д.

Часто указатель выглядит, как маленькая стрелка, но его вид меняется в зависимости от операций, которые выполняет пользователь. Так, если указатель имеет форму маленьких песочных часов, это означает, что компьютер занят выполнением какой-то

операции. А если указатель мыши принимает вид перечеркнутого круга, это означает, что запрашиваемая операция в данном случае невозможна. Управление основными элементами мыши при сохранении эргономичности пользовательских интерфейсов показано в таблице 5.

Таблица 5. Управление основными элементами мыши

Элементы мыши	Описание
	щелчок – быстрое нажатие и отпускание левой кнопки мыши
	двойной щелчок – два щелчка, выполненные с малым интервалом времени между ними
	перетаскивание – перемещение экранного объекта, на котором установлен указатель
	протягивание – происходит изменение формы экранного объекта
	наведение указателя мыши на значок объекта – на экране появляется всплывающая подсказка, кратко характеризующая свойства объекта

Возможности графического интерфейса:

- *подводить* курсор к той или иной части экрана;
- *выделять* на экране имя файла или команду другим цветом;
- *оперировать* выделенными данными независимо от других.

Графический интерфейс имеет 2 основных элемента: окно и меню. Существуют различные типы этих элементов (*схема 8*).



Схема 8. Основные элементы графического интерфейса

Рабочий стол – основное окно графического интерфейса ОС, занимающее все пространство экрана (*рис. 41*).



Рис. 41. Рабочий стол

Окно папок – область экрана, в которой графически представлено содержимое папки (рис. 42).



Рис. 42. Окно папок

Диалоговое окно – окно, отображаемое на экране в случае необходимости ввода информации или отображения параметров для создания действия (рис. 43).

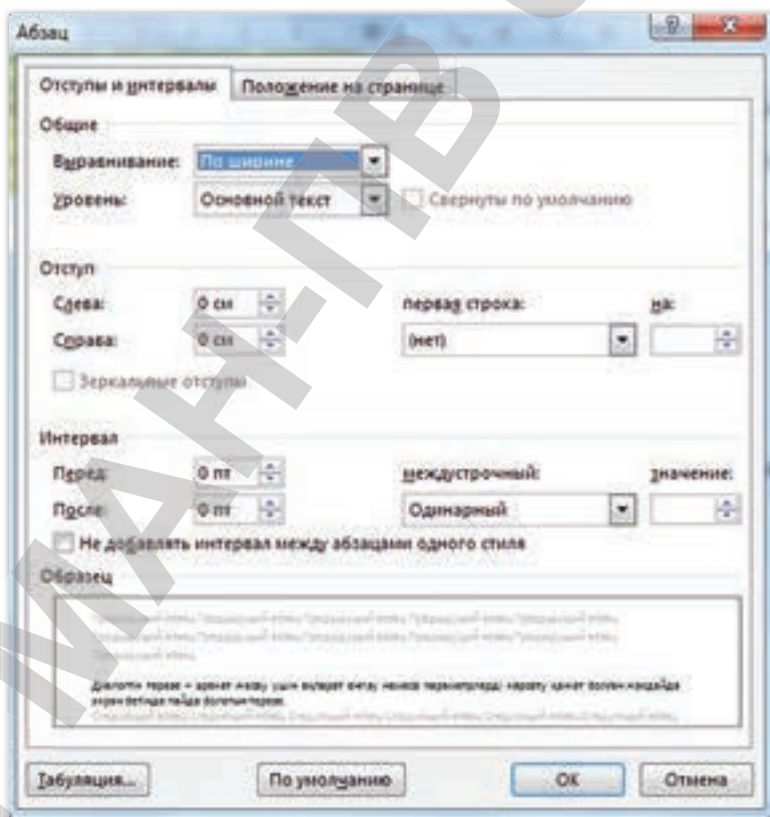


Рис. 43. Диалоговое окно

Окно приложения – область экрана, в которой представлено запущенное на выполнение приложение (рис. 44).

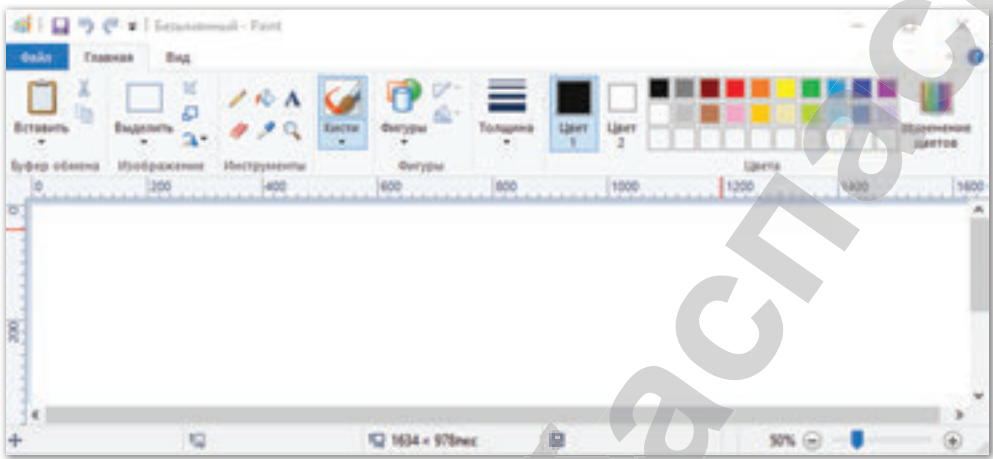


Рис. 44. Окно стандартного приложения Windows Paint

Окно документа – личная папка пользователя. В папке хранятся личные файлы пользователя (рис. 45).

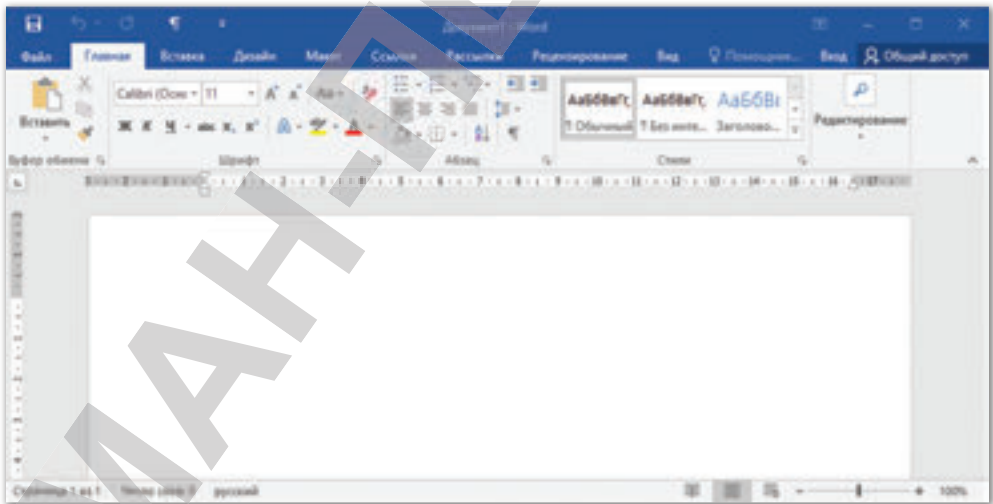


Рис. 45. Окно документа

Контекстное меню – список команд, относящихся к текущему объекту. Оно появляется при щелчке правой кнопкой мыши на выделенном объекте (рис. 46).

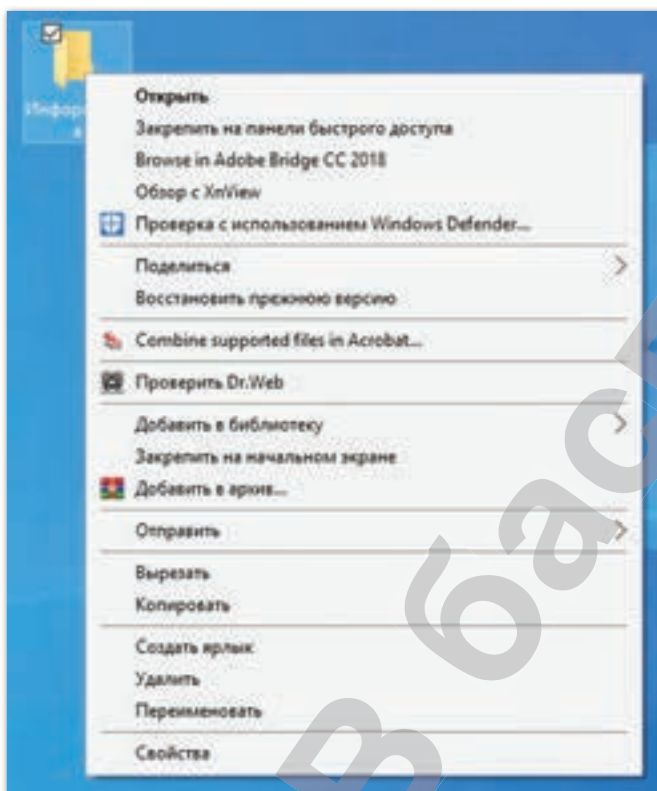


Рис. 46. Контекстное меню

В наши дни взаимодействие человека и компьютера строится на основе объектно-ориентированного графического интерфейса, в котором:

- все объекты представляются в виде значков (иконок, пиктограмм), выбор которых с помощью, например, мыши приводит к активизации соответствующих объектов;
- оперирование объектами осуществляется в окнах, заранее заданных, очерченных рамками, частях экрана;
- основным элементом программного управления является меню – выводимый на экран список команд, которые можно задать компьютеру;
- основным элементом аппаратного управления являются различные манипуляторы (мышь, трекбол, сенсорная панель и т.д.). Действительно, при выполнении какой-либо операции пользователь сначала выбирает объект, а затем – действие над этим объектом; мышь позволяет осуществлять прямое манипулирование выбранным объектом.

1

Отвечаем на вопросы

1. Что такое пользовательский интерфейс?
2. Что такое командный интерфейс?
3. Какие основные возможности реализованы в современных графических интерфейсах?
4. Какие основные элементы графического интерфейса вы знаете?
5. Что такое Рабочий стол?
6. Какие основные элементы окон папок бывают? Какие из них присутствуют в окнах известных вам приложений?

2

Думаем и обсуждаем

1. Почему командный пользовательский интерфейс не способствовал широкому распространению компьютерной техники?
2. Почему специалисты до сих пор используют интерфейс командной строки?
3. Почему современные пользовательские интерфейсы можно считать объектно-ориентированными?

3

Анализируем и сравниваем

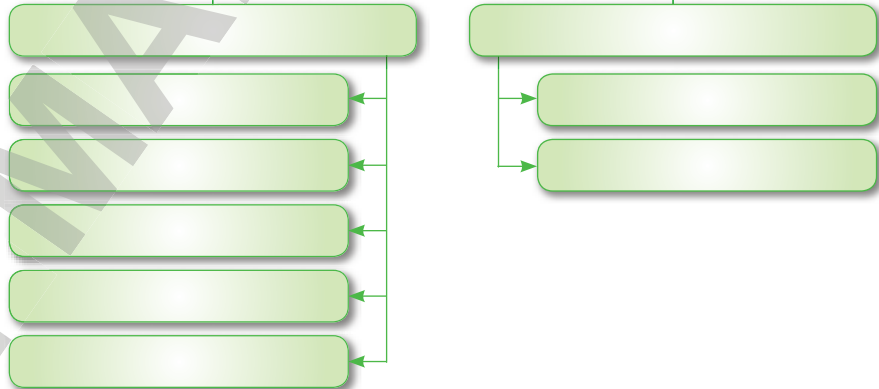
Создайте диаграмму Венна по видам пользовательского интерфейса.

4

Выполняем в тетради

Нарисуйте в своих тетрадях схему и напишите описание основных элементов графического интерфейса:

Основные элементы графического интерфейса



5

Выполняем на компьютере

1. Управление основными элементами мыши при сохранении эргономичности пользовательских интерфейсов:
 - щелкните по левой кнопке мыши один раз;
 - выделите объект на рабочем столе и перетаскивайте его из одного места в другое путем одинарного клика и удержания левой кнопки мыши, увеличьте и уменьшите масштаб;
 - дважды щелкните по левой кнопке мыши;
 - ознакомьтесь с командами контекстного меню, которые появляются при щелчке правой кнопкой мыши по любому объекту на рабочем столе.
2. Эргономика выполняемых действий на рабочем столе:
 - проверьте, какие объекты находятся на рабочем столе;
 - выберите объект «Мой компьютер» на рабочем столе и щелкните по правой кнопке мыши. Просмотрите команды в появившемся контекстном меню;
 - нажмите кнопку Пуск и откройте два приложения из ряда Стандартные, увеличьте и уменьшите появившиеся окна на рабочем столе.

6

Делимся мыслями

Что нового вы узнали на уроке? Поделитесь своими размышлениями с одноклассниками. Каким образом можно использовать в повседневной жизни полученные на уроке знания и навыки? Приведите примеры.

7

Домашнее задание

Узнайте, какие аналогии с повседневной жизнью реализованы в современном пользовательском интерфейсе?

§ 9. Практикум. Пользовательский интерфейс

Уровень А

Задание 1. Работа с окном командной строки.

1. Нажмите кнопку Пуск, из ряда Стандартные выберите Командную строку.
2. В появившемся окне введите следующие команды и посмотрите, какие изменения произошли:

Команды	Описание
DATE	
DIR	
TIME	

Задание 2. Опишите команды, которые выполняются с помощью командной строки:

Команды	Описание
COLOR	
MD	
FIND	
EXIT	

Уровень В

Задание 1. Эргономика выполняемых действий на Рабочем столе.

1. Щелкните правой кнопкой мыши пустую область рабочего стола, выберите Вид в контекстном меню, по очереди выберите «Крупные значки, Обычные значки, Мелкие значки» и посмотрите, какие изменения произошли.
2. Точно так же выберите Сортировка в контекстном меню.
3. Измените фон Рабочего стола, выполняя те же действия.

Задание 2. Объясните изменения по выполненным действиям на Рабочем столе.

Уровень С

Настройка параметров компьютера.

Задание 1. Выберите Пуск ⇒ Панель управления ⇒ Мышь и ознакомьтесь с видами указателей курсора (рис. 47).

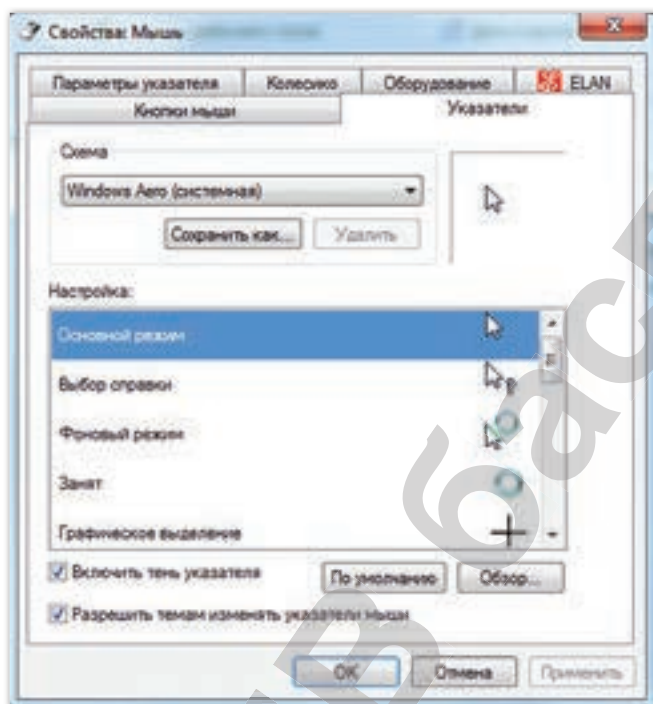


Рис. 47. Свойства компьютерной мыши

Задание 2. Выберите Пуск ⇒ Панель управления ⇒ Гаджеты Рабочего стола и ознакомьтесь с видами гаджетов (рис. 48).

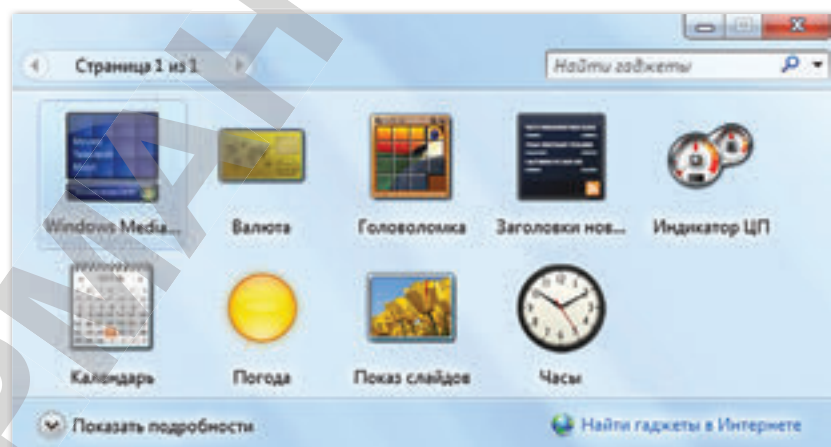
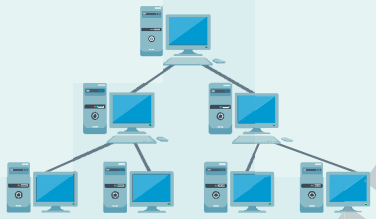
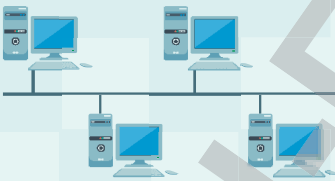
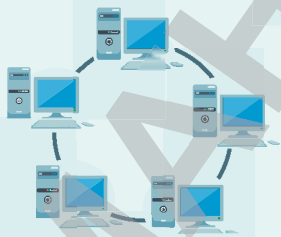
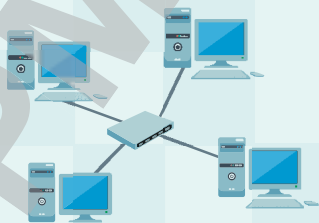



Рис. 48. Виды гаджетов

ЗАДАНИЯ ПО СУММАТИВНОМУ ОЦЕНИВАНИЮ К РАЗДЕЛУ 2

1. Закончите предложение:
Компьютерная сеть – это ...
2. Типы компьютерных сетей, в зависимости от территориальной распространенности, – ...
 - А. Локальная сеть, региональная сеть, глобальная сеть.
 - В. Ведомственная сеть, государственная сеть.
 - С. Топология шина, топология кольцо, топология звезда, топология дерево.
 - Д. Проводная, беспроводная.
 - Е. Витая пара, коаксиальный кабель, оптический кабель.
3. Опишите сетевые топологии по методу геометрической связи:

Типы топологии	Наименование	Описание типа связи
		
		
		
		

Типы топологии	Наименование	Описание типа связи
		

4. **Единица измерения скорости передачи данных в сети:**
- Количество бит в секунду (бит/с)
 - Количество байт в секунду (байт/с)
 - Количество бит в минуту (бит/м)
 - Количество байт в минуту (байт/м)
 - Количество бит в час (бит/час)
5. **Признаки появления вирусов на компьютере:**
- ...
 - ...
 - ...
6. **Заполните пропуски:**
- _____ – вирусы, которые остаются в памяти компьютера в течение длительного времени.
 - _____ – вирусы, которые не заражают память компьютера и являются активными лишь ограниченное время.
7. **Специальная программа, предназначенная для выявления компьютерных вирусов, а также нежелательных программ, восстановления зараженных файлов и предотвращения заражения файлов или операционной системы вредоносным кодом:**
- Антивирусная программа
 - Компьютерные вирусы
 - Файловые вирусы
 - Прикладная программа
 - Системная программа
8. **Методы защиты от компьютерных вирусов:**
- ...
 - ...
 - ...
9. **Совокупность средств и правил взаимодействия человека и компьютера:**
- Пользовательский интерфейс
 - Командный интерфейс
 - Графический интерфейс
 - Объектно-ориентированный графический интерфейс
 - Интерфейс
10. **Возможности графического интерфейса:**
- ...
 - ...
 - ...

ИТОГИ 2 РАЗДЕЛА

В разделе «Сети и безопасность» вы подробно ознакомились с компьютерными сетями и их классификацией, защитой компьютера от вредоносных программ, оценкой эргономичности пользовательских интерфейсов. Вы узнали о важности компьютерных сетей и безопасности в сети Интернет.

В данном разделе мы подробно остановились на понятии «компьютерные сети» и видах их классификации. Кроме того, вы ознакомились с разработанными компьютерными вирусами и антивирусными программами. Выполнили практическую работу по эргономичности пользовательских интерфейсов как совокупности правил и средств взаимодействия человека и компьютера.

Надеемся, что знания, полученные в этом разделе, пригодятся вам в будущем.

РЕШЕНИЕ ЗАДАЧ С ПОМОЩЬЮ ЭЛЕКТРОННЫХ ТАБЛИЦ

Изучив раздел, вы научитесь:

- форматировать элементы таблицы в текстовом процессоре;
- форматировать элементы электронной таблицы;
- создавать диаграммы в электронной таблице;
- использовать условное форматирование в электронной таблице;
- использовать различные типы данных в электронной таблице;
- создавать формулы для вычислений в электронной таблице.

§ 10. Таблицы в текстовом процессоре

Вспомните!

- Что такое компьютерные сети?
- Какие компьютерные вирусы бывают?
- Что такое антивирусные программы?
- Что такое пользовательский интерфейс?
- Что такое командный интерфейс?
- Каковы основные элементы графического интерфейса?

Вы узнаете:

- как форматировать элементы таблицы в текстовом процессоре;
- о кнопках раздела Таблица.

Таблица – Кесте – Table

Ячейка – Ұяшық – Cell

Столбец – Баған – Column

Строка – Қатар – Row

С текстовым процессором вы знакомились в 6 классе. Вы работали с приложением Microsoft Word, которое позволяет создавать, редактировать и печатать текстовые документы. Вы знаете, что все необходимые документы обрабатываются и создаются в текстовом редакторе.

Теперь давайте рассмотрим возможности создавать и форматировать таблицы в этом текстовом редакторе.

Что такое таблица?

Таблица – способ представления данных путем размещения их на пересечении строк и столбцов. Таблица состоит из упорядоченного набора строк и столбцов. Пересечение строки и столбца называется **ячейкой таблицы**. В ячейке можно вставить цифры, тексты, встроенные таблицы и графические символы. Для успешного преобразования ячеек таблицы предусмотрены различные функции форматирования. В каждом столбце таблицы должна содержаться однотипная информация.

Как создать таблицу в текстовом процессоре?

Для того чтобы создать таблицу в текстовом процессоре, выполняем команду из меню Вставка ⇒ Таблицы (рис. 49). В открывшемся окне есть разные варианты вставки таблиц. Выберите один из параметров и укажите размер таблицы (Число столбцов и Число строк) и типы автоматического выбора.

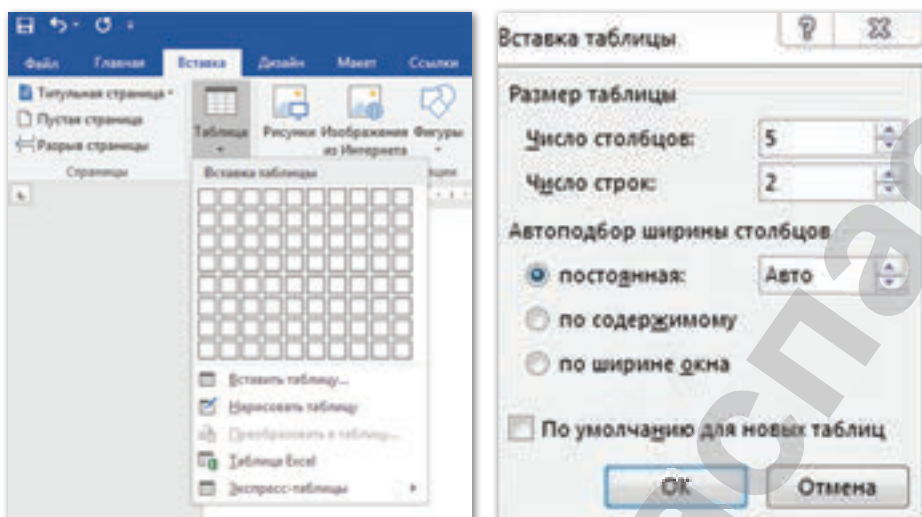


Рис. 49. Создание таблицы в текстовом процессоре

Выделите созданную таблицу и преобразуйте с помощью различных видов форматирования из вкладок **Конструктор** и **Макет** (рис. 50–51).

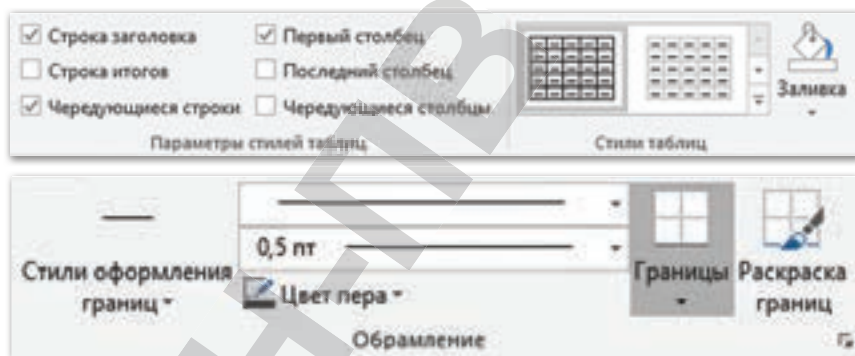


Рис. 50. Содержимое вкладки Конструктор

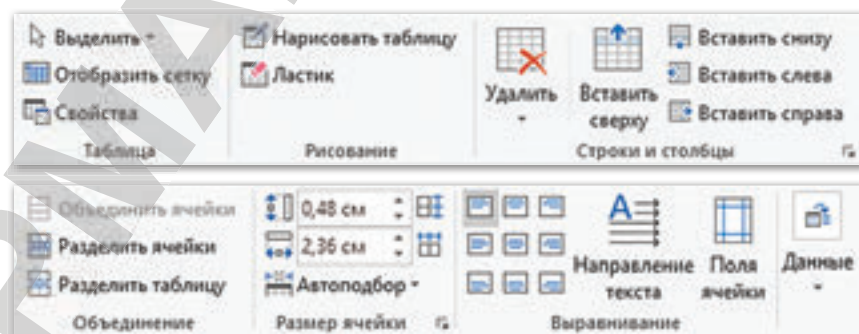


Рис. 51. Содержимое вкладки Макет

Теперь познакомимся с содержимым вкладки **Конструктор**.

Первая группа, расположенная во вкладке **Конструктор**, называется **Параметрами стилей таблиц**. С помощью инструментов этой группы можно установить или снять флажки рядом с теми элементами таблицы, которые требуется отобразить или скрыть в выбранном стиле (рис. 52).

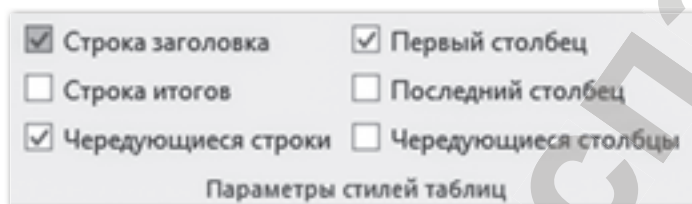


Рис. 52. Параметры стилей таблиц

Вторая группа вкладки **Конструктор** – **Стили таблиц**. Здесь используется комбинация цветов границ для быстрого визуального изменения стиля таблицы, внешнего вида стиля каждой таблицы (рис. 53–54).



Рис. 53. Стили таблиц



Рис. 54. Визуальное изменение стиля таблицы

Следующая группа вкладки **Конструктор** – **Обрамление**. Здесь вы можете отформатировать границы таблицы с дополнительными настройками для толщины и цвета (рис. 55–56).



Рис. 55. Параметры вкладки **Обрамление**



Рис. 56. Обрамление границ таблицы

Теперь ознакомьтесь с инструментами вкладки **Макет**.

Первая группа вкладки **Макет** – **Таблица**. С помощью инструментов этой группы можно выделить все ячейки строки,

столбцов или ячеек в таблице, показать или скрыть поля ячеек в таблице, просмотреть свойства таблицы (рис. 57).

Группа **Рисование** вкладки **Макет** создает таблицы, рисуя границы строк и столбцов ячеек. Здесь также имеется ластик для таблицы (рис. 58).

Группа **Строки и столбцы** вкладки **Макет** позволяет вставить или удалить новые строки и столбцы в таблице (рис. 59).

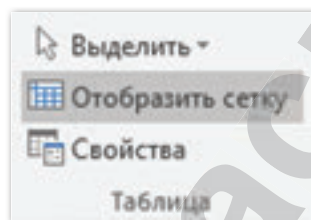


Рис. 57. Функция Свойства таблицы

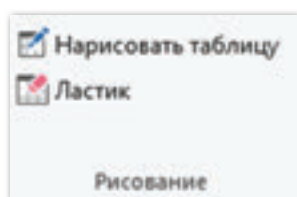


Рис. 58. Функция Нарисовать таблицу

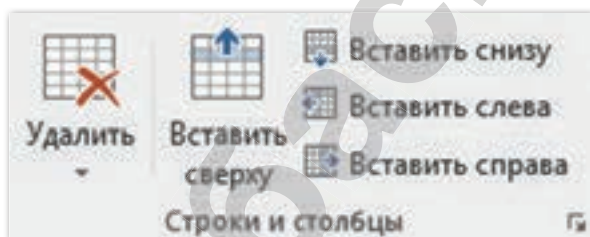


Рис. 59. Функция Вставить и удалить строки и столбцы

Следующая группа вкладки **Макет** – **Объединение**. Здесь представлены функции объединения выбранных ячеек, разделения текущей ячейки на несколько ячеек или одной таблицы на две (рис. 60).

В группе **Размер ячейки** вкладки **Макет** автоматически указывается высота строки таблицы и ширина столбцов или тип выборки (рис. 61).

В группе **Выравнивание** вкладки **Макет** можно редактировать текст в ячейке по порядку расположения, изменять направление выделенных текстов, интервал между ячейками и поля ячеек (рис. 62).

Последняя группа вкладки **Макет** – **Данные**. Она позволяет отсортировать выделенный фрагмент по алфавиту или числовому значению, ввести формулы

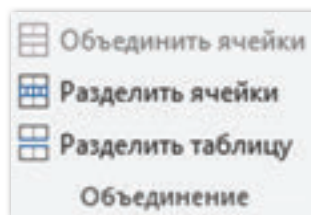


Рис. 60. Функции Объединить и Разделить ячейки таблицы

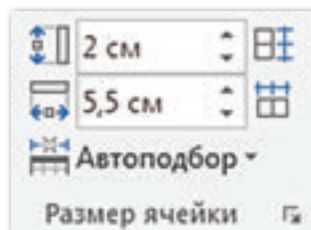


Рис. 61. Функция Размер ячейки

в ячейку, преобразовать текст, повторить строки заголовков (рис. 63).

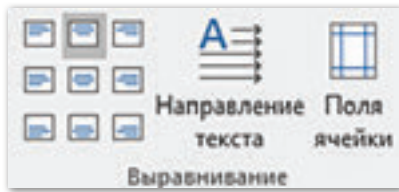


Рис. 62. Функция Выравнивание

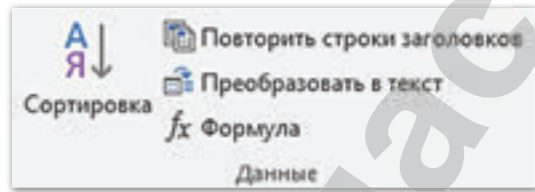


Рис. 63. Функция Данные

В текстовых документах часто используется эффективная, компактная организация и представление данных с помощью таблицы.

1

Отвечаем на вопросы

1. Что такое таблица?
2. Что такое ячейка?
3. Какую строку меню нужно выбрать для создания таблицы в текстовом процессоре?
4. Из каких групп состоит форматирование таблиц?
5. Из каких групп состоит вкладка Конструктор?
6. Какую функцию выполняет группа Строки и столбцы?

2

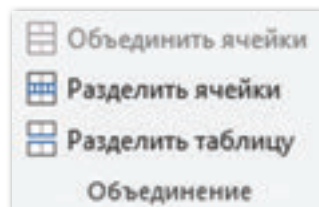
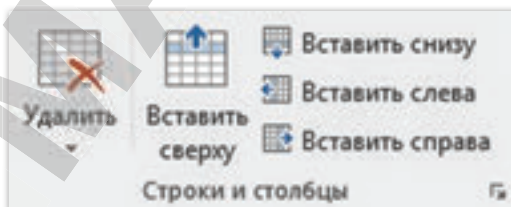
Думаем и обсуждаем

1. Для чего создают таблицы в текстовом процессоре?
2. В чем эффективность представления данных в виде таблиц?

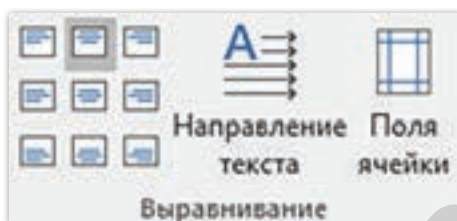
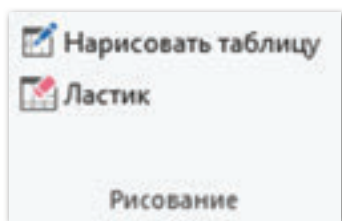
3

Анализируем и сравниваем

1. Сравните группы **Строки и столбцы** и **Объединение**. Какие различия можно отметить?



2. Сравните группы **Рисование** и **Выравнивание**. Какие различия можно отметить?



4

Выполняем в тетради

Напишите функции следующих инструментов:

№	Инструменты	Функции
1		
2		
3		

5

Выполняем на компьютере

1. Работа с инструментами вкладки Конструктор при создании таблицы:
 - 1) Откройте текстовый процессор Word.
 - 2) Создайте таблицу, которая должна состоять из 8 строк и 9 столбцов.
 - 3) Измените внешний вид стиля таблицы путем быстрого визуального изменения инструментов группы **Стили таблиц**. Используйте комбинацию цветов границ ячеек.
 - 4) С помощью инструментов группы **Обрамление** отформатируйте границы таблицы с дополнительными настройками для толщины и цвета.
2. Работа с инструментами вкладки Макет при создании таблицы:

- 1) Создайте дополнительные ячейки с помощью инструментов группы **Рисование** и удалите созданные ячейки с помощью **Ластика**.
 - 2) С помощью инструментов группы **Строки и столбцы** добавьте новые строки и столбцы. С помощью инструментов группы **Объединение** объедините созданные строки и столбцы или разделите на несколько ячеек.
 - 3) Напишите название дней недели в каждой ячейке столбца. Выровняйте текст по возрастанию и измените интервал и поля между ячейками.
 - 4) С помощью инструментов группы **Данные** отсортируйте дни недели по алфавиту и преобразуйте текст.
3. Проведите исследование на тему «Физиологические нормы потребности детей и подростков в питательных веществах (грамм в сутки)». Результаты исследования представьте в виде таблицы, созданной в текстовом процессоре:

Возраст	Белки (г/кг)	Жиры (г/кг)	Углеводы	Энергетическая потребность (ккал)
7–10	2,3	1,7	330	2550
10–15	2,0	1,7	375	2900
16 и старше	1,9	1,0	475	3100

6

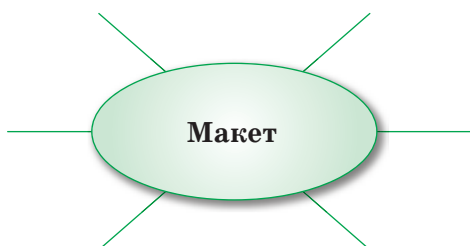
Делимся мыслями

Как вы думаете, какие знания вы можете использовать в повседневной жизни, чтобы научиться создавать таблицы для текстовых документов?

7

Домашнее задание 

Расскажите о группах вкладки **Макет**:



§ 11. Практикум. Таблицы в текстовом процессоре

Уровень А

Создание таблиц и форматирование границ таблицы в текстовом процессоре.

1. Создайте таблицу в текстовом процессоре по образцу:

Ряд	Наименование предметов	Понедельник	Вторник	Среда	Четверг	Пятница
1	Математика					
2	Казахский язык					
3	Физика					
4	Английский язык					
5	Физическая культура					
6	Литература					


2. Настройте толщину и цвет границ таблицы с помощью инструментов группы **Обрамление**  вкладки **Конструктор**.

Уровень В

Форматирование строк и столбцов таблицы.

1. Создайте таблицу в текстовом процессоре и объедините ячейки таблицы по заданному шаблону:

№							
1							
2							
3							
4							
5							
6							

2. Добавьте новые строки и столбцы в созданную таблицу.
3. В первый столбец введите текст. Выделите введенный текст и измените его направление с помощью инструментов группы  вкладки Макет.

Уровень С

Использование формул в ячейках таблицы.

1. Найдите сумму товара по данным в таблице:

Товар	Яблоки (кг)	Клубника (кг)	Морковь (кг)	Малина (кг)	Виноград (кг)	Огурцы (кг)	Сумма (кг)
Фрукты	20	0	0	0	15	0	
Овощи	0	0	35	0	0	7	
Ягоды	0	30	0	20	0	0	

Для этого вставьте курсор в ячейку вычисления и выберите Макет ⇒ Данные ⇒ Формула. Обратите внимание на открытое окно. Здесь рассмотрены возможности вставки формулы и функции, формата числа (рис. 64).

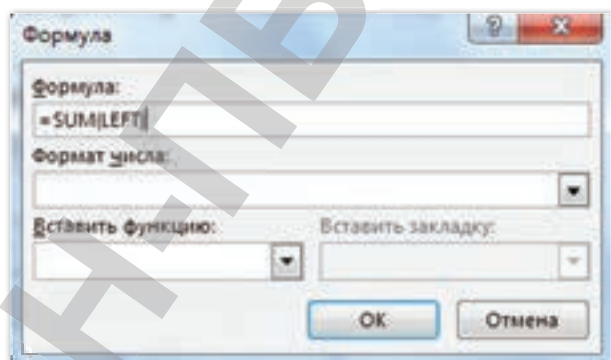


Рис. 64. Формула

Функция =SUM(LEFT) вычисляет сумму ячеек, которые находятся в левой стороне. Направления вправо, влево, вниз, вверх описаны на английском языке: =SUM(ABOVE), =SUM(BELOW), =SUM(RIGHT), =SUM(LEFT;RIGHT), =SUM(LEFT;BELOW) и др. Сумма следующего товара будет реализована именно по этому шагу.

2. Приведите дополнительные примеры, используя формулы таблицы.

§ 12. Форматирование элементов электронных таблиц

Вспомните!

- Что такое таблица?
- Что такое ячейка?
- Какую строку меню нужно выбрать для создания таблицы в текстовом процессоре?
- Из каких групп состоит форматирование таблицы?
- Из каких групп состоит вкладка Конструктор?
- Из каких групп состоит вкладка Макет?

Вы узнаете:

- о таблице Microsoft Excel;
- о рабочем окне Microsoft Excel;
- что такое рабочая книга, рабочий лист, ячейка, диапазон;
- об основных элементах электронной таблицы.

ется прикладная программа Microsoft Excel, интегрированная в пакет Microsoft Office.

Microsoft Excel – это прикладная программа, которая предназначена для создания электронных таблиц (рис. 65).

Электронная таблица –
Электронды кесте – Spreadsheet
Диапазон – Ауқым – Range

Работа с электронными таблицами очень удобна, так как для упрощения многих математических операций и сложных вычислений в табличном процессоре реализуется автоматизированная система. Работа с электронными таблицами ведется с помощью специальных программ. В настоящее время самым популярным табличным процессором является

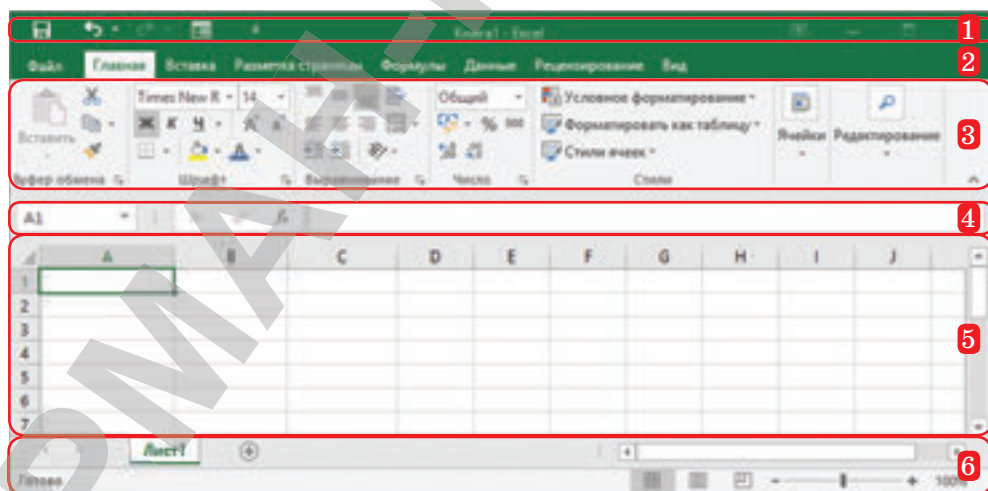


Рис. 65. Среда прикладной программы Microsoft Excel

Давайте кратко рассмотрим среду приложения Microsoft Excel.

1. *Строка заголовка с кнопками управления средой Microsoft Excel* – состоит из кнопок, которые позволяют увеличить, уменьшить и закрыть окно.
2. *Строка меню* – тип меню, позволяющий получить доступ ко всем функциям программы.
3. *Набор инструментов* – состоит из различных инструментов для форматирования и редактирования.
4. *Строка формул* – состоит из адресов ячеек и частей вызова функций.
5. *Рабочий лист* – группа ячеек, состоящих из ряда строк и столбцов.
6. *Строка состояния* – состоит из списка рабочих листов, полосы прокрутки и индикатора режимов работы.

Что относится к основным элементам электронной таблицы?

Файл, созданный в табличном процессоре, называется **Рабочей книгой**. В каждом файле может содержаться несколько листов – электронных таблиц. То есть название каждого листа устанавливается в нижней части окна, как Лист 1, Лист 2, Лист 3 При запуске окна активируется Лист 1. Вы можете удалить, добавить рабочий лист или перейти из одного рабочего листа в другой.

Электронная таблица состоит из столбцов и строк. Заголовки столбцов обозначаются латинскими буквами или сочетаниями букв (A, B, C, ...), заголовки строк – числами (1, 2, 3, ...). Всего в рабочем листе – 16 384 столбцов (от A до XFD). Строки нумеруются от 1 до 1 048 576.

Ячейка – пересечение столбца и строки.

Адрес ячейки – это заголовок столбца и номер строки, на пересечении которых расположена ячейка (рис. 6б). Например, если ячейка находится в столбце B и строке 2, она имеет адрес B2. Аналогично A9, D21, F5, G7, L16.

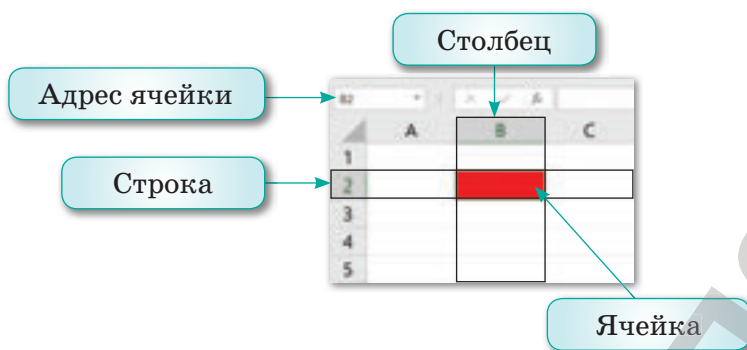


Рис. 66. Адрес ячейки

Набор из двух или более ячеек принято называть **диапазоном**. Любой диапазон, точно так же, как и ячейка, имеет свой адрес. Адрес диапазона состоит из числа выделенных строк и столбцов (рис. 67). Такой диапазон называют **смежным**, или **непрерывным**. Например, диапазон, который состоит из ячеек B2, B3, B4, и C2, C3, C4, будет записан как 3Rx2C.

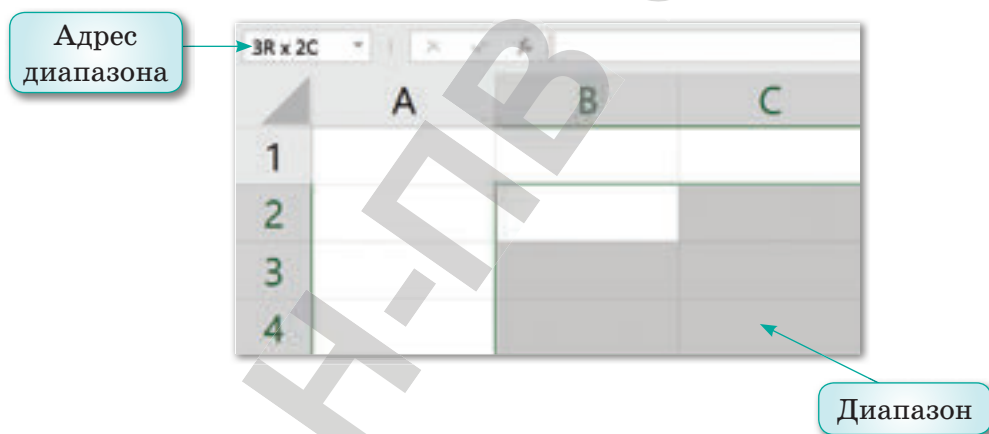


Рис. 67. Диапазон ячеек

Кроме того, чтобы выбрать диапазон, состоящий из отдельных (не смежных) ячеек, вы нажимаете клавишу Ctrl и выбираете любую ячейку, которую хотите добавить в диапазон. Например, диапазоны A1, B3, C5, D7 можно выделить клавишей Ctrl (рис. 68).

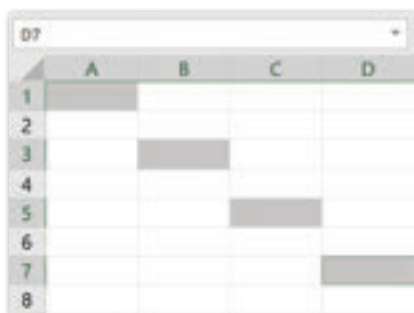


Рис. 68. Диапазон, состоящий из отдельных ячеек

Форматирование элементов таблицы

Одна из ячеек таблицы всегда является активной, а активная ячейка выделяется рамкой.

- Чтобы сделать ячейку активной, необходимо клавишами управления курсором подвести рамку к этой ячейке или щелкнуть по ней мышью;
- для выделения нескольких смежных ячеек необходимо установить указатель мыши в одну из ячеек, нажать левую кнопку мыши и, не отпуская ее, растянуть выделение на всю область;
- для выделения нескольких несмежных групп ячеек следует выделить одну группу, нажать клавишу Ctrl и, не отпуская ее, выделить другие ячейки;
- чтобы выделить целый столбец или строку таблицы, необходимо щелкнуть мышью на его имени.

Оформление внешнего вида ячеек, внесение корректировок в данные различного характера, объединение ячеек и другие функции происходят путем форматирования ячеек.

Окно **Формат ячеек** можно открыть двумя способами.

Способ 1. Нажав на кнопку в правом нижнем углу раздела Число в Главном меню (рис. 69).

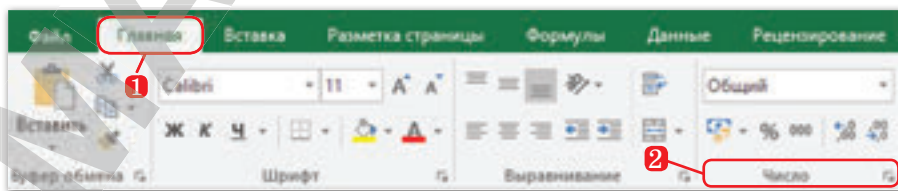


Рис. 69. Открытие окна Формат ячеек (Сп. 1)

Способ 2. Выбрать команду **Формат ячеек** из контекстного меню выделенной ячейки (рис. 70).

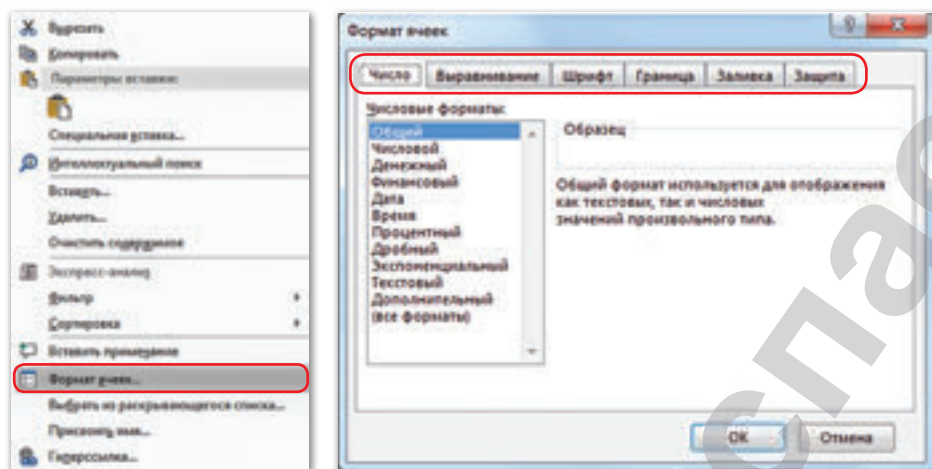


Рис. 70. Открытие окна *Формат ячеек* (Сп. 2)

Окно **Формат ячеек** имеет шесть вкладок: Число, Выравнивание, Шрифт, Граница, Заливка, Защита.

На вкладке **Число** окна **Формат ячеек** можно назначать числовые форматы ячейкам электронных таблиц. Значение в ячейке можно представить как обычное число, проценты, дату и т.д. К числовым форматам относятся: Общий, Числовой, Денежный, Финансовый и т.д. (рис. 71).

Во вкладке **Выравнивание** можно выполнить такие операции, как объединение ячеек, выравнивание и направление текста в ячейках, перенос по словам и т.д. (рис. 72).

С помощью инструментов на вкладке **Шрифт** можно преобразовать тип данных в ячейке, изменять размер или цвет шрифта (рис. 73).

	A
1	12,32
2	35 142,00 KZT
3	11.6.21
4	13:38:41
5	100000,0000%
6	1,243E+05
7	информатика
8	124-5843
9	

Рис. 71. Вкладка *Число*

	A	B	C
1	Понедельник	Вторник	Среда

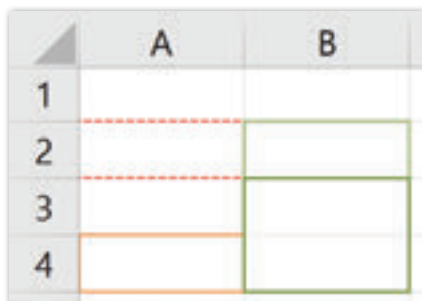
Рис. 72. Выравнивание ячеек

	A	B	C
1	Понедельник	Вторник	Среда

Рис. 73. Шрифт в ячейке

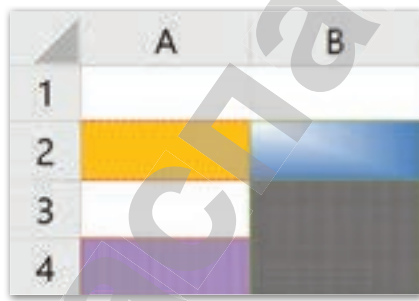
Вкладка **Граница** содержит дополнительные параметры, состоящие из настройки цвета и толщины границ таблицы (рис. 74).

Вкладка **Заливка** содержит выбор ячеек таблицы с типом цветов и выражений (рис. 75).



	A	B
1		
2		
3		
4		

Рис. 74. Граница ячеек



	A	B
1		
2	Yellow	Blue
3		Dark Grey
4	Purple	

Рис. 75. Заливка ячеек

Вкладка **Защита** позволяет защитить данные, записанные в ячейке, от удаления или изменения их значения.

1

Отвечаем на вопросы

1. Что вы знаете о программе Microsoft Excel?
2. Для чего предназначен Рабочий лист?
3. Из чего состоит Рабочий лист?
4. Сколько строк и столбцов имеется в одном рабочем листе?
5. Что такое ячейка?
6. Что такое диапазон?
7. Как выделяется диапазон?
8. Как открыть окно Формат ячеек?
9. Из каких вкладок состоит окно Формат ячеек?

2

Думаем и обсуждаем

1. Почему используются табличные процессоры?
2. В чем заключается эффективность табличных процессоров?

3

Анализируем и сравниваем

1. Проанализируйте основные элементы таблицы.
2. Сравните 6 вкладок, в которых вы можете отформатировать элементы электронной таблицы.

4

Выполняем в тетради

Запишите в тетради функции и области применения основных окон среды Microsoft Excel:

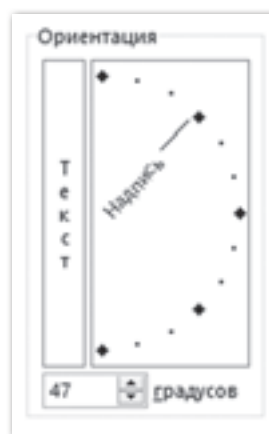
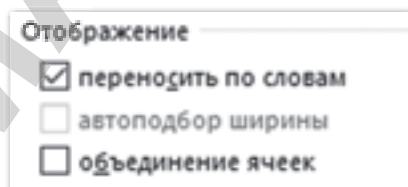
Основные окна среды MS Excel	Функции	Область применения
Строка заголовка		
Строка меню		
Набор инструментов		
Строка формул		
Рабочий лист		
Строка состояния		


5

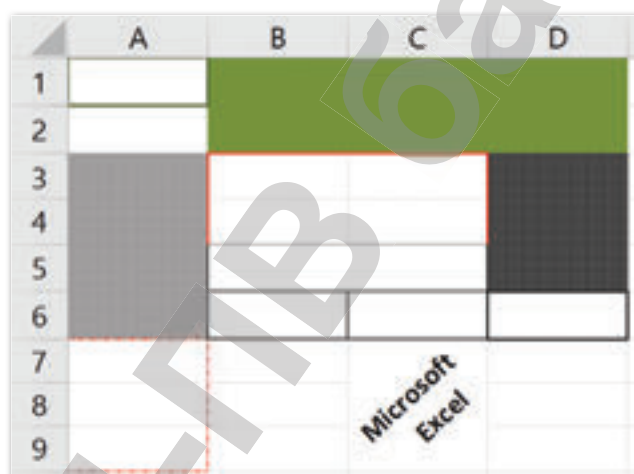
Выполняем на компьютере

1. Работа с основными элементами окна Microsoft Excel:

- 1) Запустите программу Microsoft Excel. Для этого выберите Пуск ⇒ Все программы ⇒ папка MS Office ⇒ MS Excel. Можно вызвать контекстное меню в пустом месте Рабочего стола и выбрать Создать лист MS Excel.
- 2) Наберите в ячейку С4 – наименование школы, в ячейку D4 – класс, а в ячейку E4 – вашу фамилию и имя.
- 3) Выделите ячейку и откройте диалоговое окно Формат ячеек из контекстного меню.
- 4) Выберите вкладку Выравнивание и вставьте галочку на «переносить по словам». Измените ориентацию текста. Следите за тем, какие изменения произошли в ячейке:



- 5) Выберите вкладки Граница и Заливка, измените толщину, цвет текста.
 - 6) Скопируйте тексты ячеек C4, D4, E4 на F5, G5, H5.
 - 7) Выделите диапазон ячеек B9:E14, с помощью вкладки Выравнивание объедините ячейки и измените толщину и цвет границ таблицы.
2. Работа с Рабочим листом.
- 1) Откройте новые рабочие листы с помощью кнопки , которая расположена в строке состояния MS Excel.
 - 2) Переименуйте Рабочие листы, щелкнув по ярлычку листа два раза левой кнопкой мыши и введя новое название.
 - 3) Измените расположения рабочих листов между собой.
 - 4) Преобразуйте Лист 2, как на данном рисунке:



6

Делимся мыслями

Что нового вы узнали на уроке? Поделитесь своими размышлениями с одноклассниками. Каким образом можно использовать полученные на уроке знания и навыки в повседневной жизни? Приведите примеры.

7

Домашнее задание

Закончите предложения:

Рабочий лист С помощью рабочего листа Число столбцов Число строк Ячейка – это Диапазон – это С помощью диалогового окна Формат ячеек

§ 13. Типы данных. Использование различных типов данных в электронной таблице

Вспомните!

- Какие основные функции программы MS Excel вы знаете?
- Из каких частей состоит окно Microsoft Excel?
- Расскажите о рабочей книге, рабочем листе, ячейке.
- Каковы основные элементы электронной таблицы?

Вы узнаете:

- о типах данных;
- как использовать различные типы данных в электронной таблице.

Текст – Мәтін – Text

Число – Сан – Number

Формула – Формула – Formula

Что такое типы данных?

Типы данных – это определенная характеристика содержания в таблице Excel, благодаря которой программа идентифицирует информацию и понимает, каким образом ее следует обрабатывать и отображать.

В ячейки таблицы можно ввести следующие типы данных (схема 9):

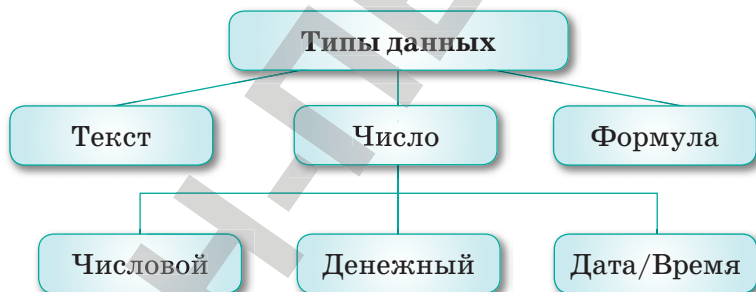


Схема 9. Типы данных

Однако ячейки таблицы в процессе работы принимают только один из трех типов данных.

Тип данных Текст может включать в себя алфавитные, числовые и специальные символы. В одну ячейку можно ввести до 32767 символов, что составляет 30 страниц файла стандартного текстового редактора. Тип данных Текст не предназначен для выполнения расчетов и дается только в качестве информационного характера.

По умолчанию любой текст, введенный в ячейку рабочего листа Excel, выравнивается по левому нижнему краю ячейки, а числа – по правому нижнему краю (рис. 76). Если текст выступает от края ячейки, то есть состоит из длинного предложения, то он накладывается на соседнюю ячейку (рис. 77). В этом случае нужно выделить текст и из контекстного меню выбрать **Формат ячеек** ⇒ **Выравнивание** и вставить галочку на **Переносить по словам** либо можно выполнить команду **Главная** ⇒ **Выравнивание** ⇒ **Перенести текст**.

	A
1	Excel
2	
3	12345
4	
5	

Рис. 76. Содержание текста

	A	B
1		
2	Тип данных	
3		
4		

Рис. 77. Длинный текст

Некоторые действия в Microsoft Excel приходится повторять по несколько раз, что отнимает большое количество времени. При заполнении таких данных можно воспользоваться функцией «Автозаполнение», которая поможет ускорить ввод данных на рабочий лист.

Для этого необходимо воспользоваться маркером автозаполнения, который позволяет быстро скопировать и вставить данные (рис. 78). Выделите ячейку, данные которой необходимо продублировать. В правом нижнем углу выделенной ячейки появится маленький квадрат – это маркер автозаполнения. Нажмите и, удерживая левую кнопку мыши, перетащите маркер автозаполнения, пока все требуемые ячейки не будут выделены. Разом вы можете заполнять ячейки либо столбца, либо строки. Отпустите кнопку мыши, чтобы заполнить выбранные ячейки.

Маркер автозаполнения может быть использован всякий раз, когда необходимо заполнить данные, имеющие

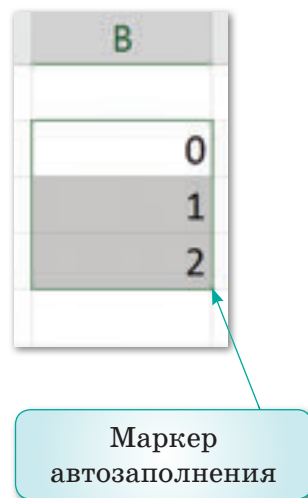


Рис. 78. Маркер автозаполнения

последовательный порядок. Например, последовательность чисел (1, 2, 3) или дней (понедельник, вторник, среда) (рис. 79).

A1	
	A
1	01.01.2021
2	02.01.2021
3	03.01.2021
4	04.01.2021
5	05.01.2021
6	06.01.2021
7	07.01.2021
8	08.01.2021
9	09.01.2021
10	10.01.2021

A1	
	A
1	1
2	3
3	5
4	7
5	9
6	11
7	13
8	15

A1	
	A
1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16

Рис. 79. Функция Автозаполнение

Тип данных **Число** применяется в различных расчетах. Для открытия этой ячейки в среде MS Excel вызовите контекстное меню и выберите команду **Формат ячеек**. Также можно выполнить команду **Главная** \Rightarrow **Число** \Rightarrow **Числовые форматы**. Здесь предполагается работа с числами, для которых может быть задан один из следующих форматов ячеек: числовой, денежный, финансовый, процентный, дробный и др. (рис. 80).

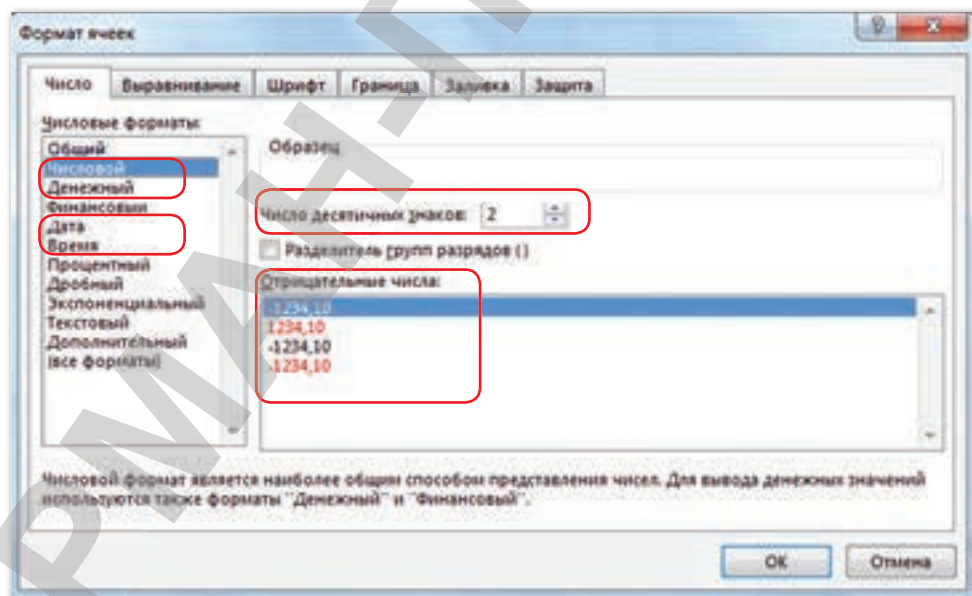


Рис. 80. Вкладка типа данных Число

Числовой – формат, который является наиболее общим способом представления чисел. Например, число 124576 отформатировано, как 124576,00 (рис. 81).

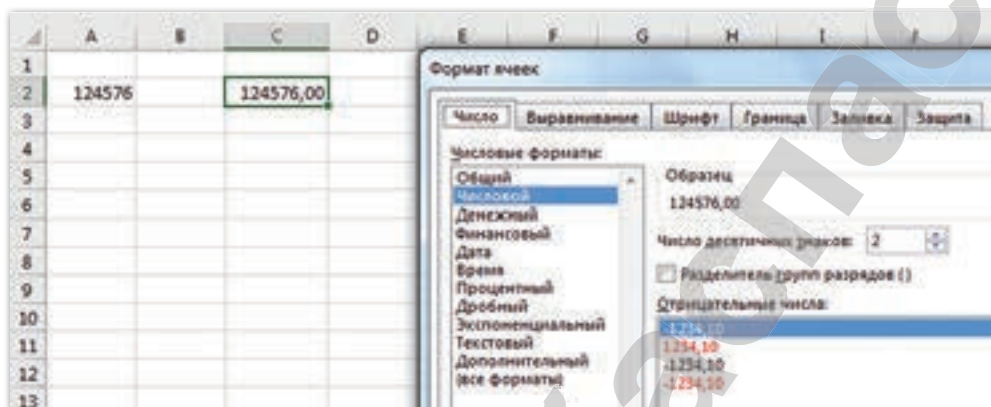


Рис. 81. Форматирование чисел

Денежный – используется для форматирования денежных единиц. Например, 124576 тенге отформатировано как 124 576,00 тенге (рис. 82).

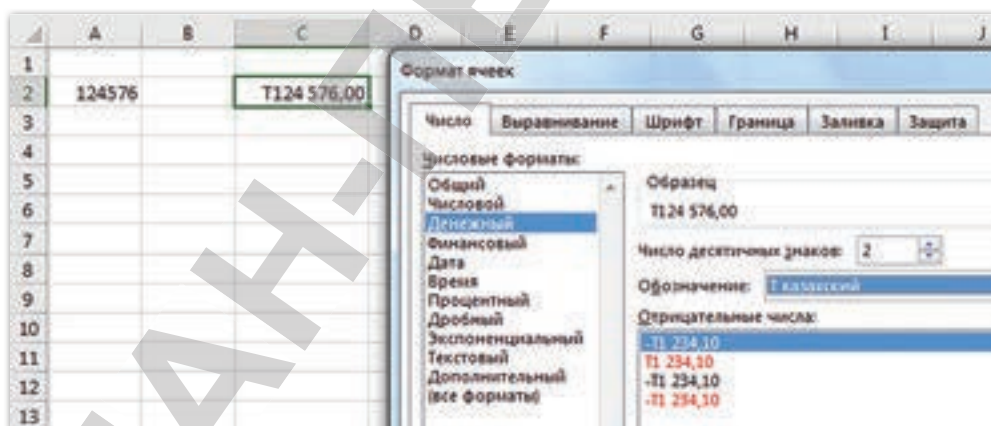


Рис. 82. Форматирование денежных единиц

Дата – служит для отображения дат и времени, представленных числами, в виде дат. Например, ДД.ММ.ГГГГ; ГГГГ-ММ-ДД; ДД-ММ и т.д. (рис. 83).

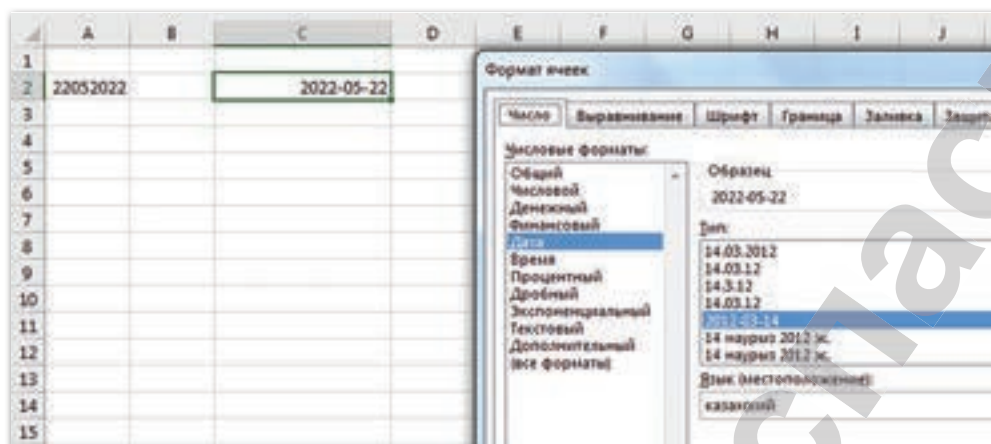


Рис. 83. Форматирование дат и времени

Время – отображает число в формате времени. Например, чч:мм:сс; чч:мм АМ/РМ; чч:мм:сс АМ/РМ и т.д. (рис. 84).

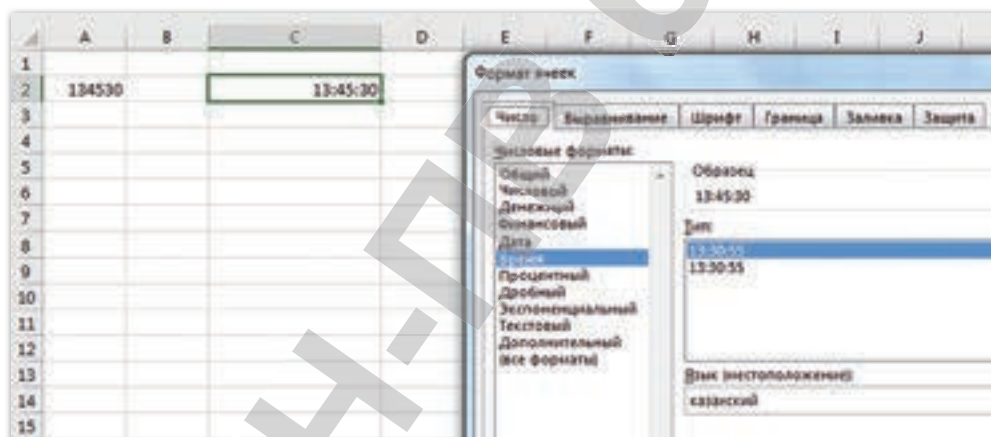


Рис. 84. Форматирование времени

Тип данных Формула. Если содержимое ячейки начинается со знака «=» (иногда «+» или «-»), тогда значение этой ячейки принимается как формула. При выводе вычислений результат и значения вычислений отображаются в ячейке, а не в формулах. Например, результат суммы значений в ячейках А2 и В2 записываются в следующую ячейку С2. И строка формул `=A2+B2` описывает всю информацию в ячейках. Если ссылка на ячейки не указана, на строке формул отображается результат (рис. 85), а если ссылка указана, на строке формул отображается адрес ячейки (рис. 86).

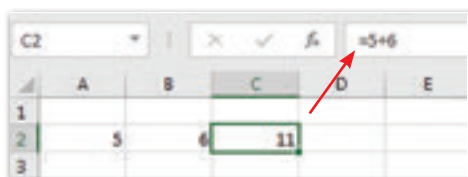


Рис. 85. Формула без указания ссылки

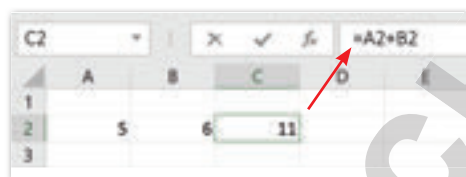


Рис. 86. Формула с указанием ссылки

Формулы работают простым способом: начинается с символа формулы «=», затем для расчета отображаются ячейки значений данных (щелкните левой кнопкой мыши по ним) и заканчивается нажатием клавиши ENTER.

В результате вы можете получить требуемый расчет по простой формуле. Последовательность действий в формуле соответствует математическим правилам: сначала умножить и разделить, затем сложить и вычесть. Действия в скобках выполняются раньше остальных.

Для типа данных Формула можно использовать маркер «Автозаполнение». Например, сумма значений в ячейках A2 и B2 отображает результат формулы в ячейке C2. Достаточно отметить результат формулы в ячейке C2 и переместить получившийся значок маркера в нужную ячейку, не отпуская левую кнопку мыши (рис. 87–88).

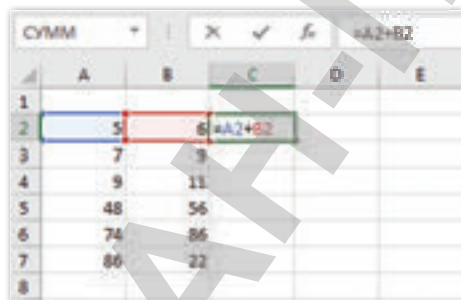


Рис. 87. Ввод формулы

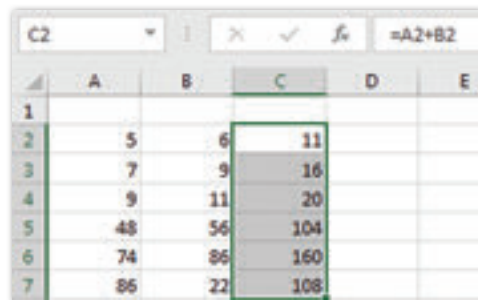


Рис. 88. Ввод с помощью маркера Автозаполнение

1

Отвечаем на вопросы

1. Что такое типы данных?
2. Какие типы данных вы знаете?
3. Что такое автозаполнение?
4. Что такое числовой тип данных?

5. Какие типы числовых данных вы знаете?
6. Как вы понимаете тип данных Формулы?

2

Думаем и обсуждаем

1. Почему типы данных делятся на текстовые, числовые и формулы?
2. Для чего нужно автозаполнение?
3. В чем причина разделения числовых типов данных на числовые, денежные и время?
4. Почему формулы начинаются со знака «=»?

3

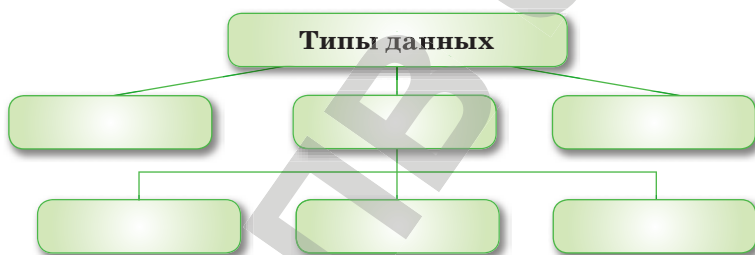
Анализируем и сравниваем

Создайте диаграмму Венна по типам данных. Приведите примеры и сравните.

4

Выполняем в тетради

Заполните схему:



5

Выполняем на компьютере

1. Работа с текстовыми и числовыми типами данных.
 - 1) Запустите программу Microsoft Excel.
 - 2) Заполните следующую таблицу:

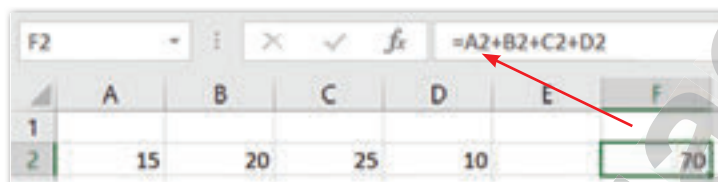
п/п	ФИО учеников класса	Дата рождения	Класс

- 3) С помощью окна Формат ячеек преобразуйте тексты.
2. Работа с типом данных Формула.

Введите числа в ячейки A2, B2, C2 и D2 и выполните вычисления по формулам, приведенным в таблице.

- 1) Ниже приведен тип формул. Необходимо присвоить разные значения ячейкам A2, B2, C2 и D2, чтобы результаты отображались в ячейке F2. Тип формул,

приведенный в таблице, должен быть записан в строке формул. Например,



- 2) Определите, получен ли результат задачи по математическим правилам, проверьте (сначала умножьте и разделите, затем сложите и вычтите):

1	$= A2+B2$	6	$= (A2+B2)+C2*D2$
2	$= A2+B2-C2$	7	$= (A2+B2)+C2*D2$
3	$= A2+B2*C2$	8	$= (A2*B2)+C2*D2$
4	$= A2/B2+C2$	9	$= A2+(B2+C2*D2)$
5	$= (A2-B2)*C2$	10	$= A2/(B2+C2)*D2$

3. Работа с функцией Автозаполнение.

Заполните данные в таблице при помощи маркера Автозаполнение (диапазон A1:F10):

По тексту	По дате	По времени	По целым числам	По нечетным числам	По формуле
Январь ...	ДД.ММ. ГГГГ ...	чч:мм:сс ...	12; 14 ...	3; 5 ...	$= A1+C2*D2$...

6

Делимся мыслями

Что нового вы узнали на уроке? Поделитесь своими размышлениями с одноклассниками. Каким образом можно использовать в повседневной жизни полученные на уроке знания и навыки? Приведите примеры.

7

Домашнее задание

Основываясь на примере, приведенном в этом параграфе, заполните таблицу Excel «Мои достижения», состоящую из списка дисциплин и показателей знаний.

§ 14. Типы данных. Создание формулы для вычислений в электронной таблице

Вспомните!

- Что такое типы данных?
- Сколько типов данных существует?
- Что такое текстовые типы данных?
- Как описать числовые типы данных?
- Как программа различает типы данных Формула?

Вы узнаете:

- что такое формула;
- что такое Строка формул;
- об операторах формул;
- о создании формул для вычислений в электронной таблице.

Строка формул – Формула жолы – Formula bar

Создание формул – Формулалар құру – Creating formulas

На прошлом уроке вы кратко ознакомились с типами данных Формула электронной таблицы. Вы знаете, что в электронной таблице все расчеты выполняются в формулах и написание формулы начинается со знака «=». Также было упомянуто о том, что состав

активной ячейки (формулы или значения) на странице таблицы описывается в **Строке формул**.

Что такое Строка формул?

Строка формул – панель, используемая для ввода и изменения текста и формул в ячейках листа. Если в ячейку заносится «4+3», то программа воспринимает их как простые числа и отображает в строке формулы. Если вставить в ячейку «=4+3», то программа воспримет это как формулу и будет описываться в Строке формул (рис. 89).

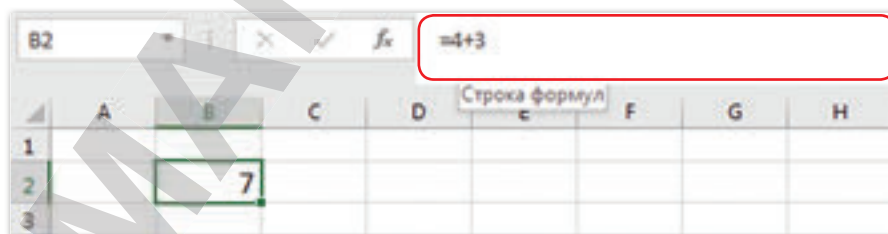


Рис. 89. Строка формул

В электронной таблице используются различные операторы формулы (арифметические и др.): текст, ссылки на ячейки,

арифметические операторы, логические операторы, справочные операторы, операторы, объединяющие два текста и скобки. Программа Excel описывает формулы в скобках (). Формулы, как и математические системы, следуют стандартному порядку расчета. Например, в формуле $=5+2*3$ первым выполняется умножение (операция $2*3$), затем результат прибавляется к числу 5.

Использование текста в формулах

В поле формулы необходимо ввести текст в двойные кавычки «». Например, если в поле формулы вводится «=Компьютер», программа Excel выводит ошибку, а запись в виде «="Компьютер"» является правильной формулой (рис. 90).

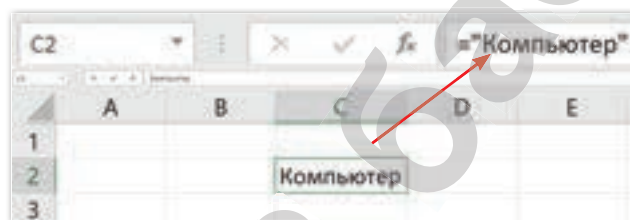


Рис. 90. Использование текста в формулах

Использование ссылок в формулах

Для того чтобы ввести адрес ячейки в Строку формул, нет необходимости вводить его вручную. После знака «=» нужно выбрать необходимую ячейку и нажать на левую кнопку мыши или выбрать нужный диапазон. Если в Формуле используется несколько ссылок, программа Excel дает каждой из них различные цвета. Пример: в ячейку C2 можно записать формулу `=A2+B2`, нажав клавишу Enter и дважды щелкнув в ячейку со ссылкой. В ячейке вы увидите формулу с цветными указателями. Цвета ячеек будут выделены одинаковыми цветами границ ячеек (рис. 91).

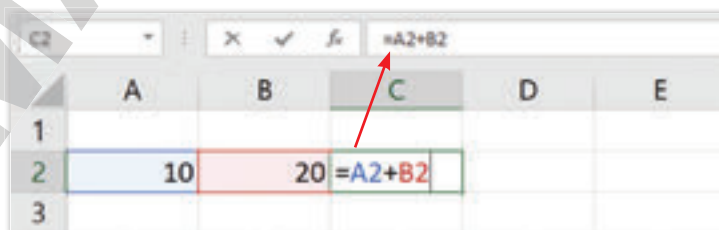


Рис. 91. Использование ссылок в Строке формул

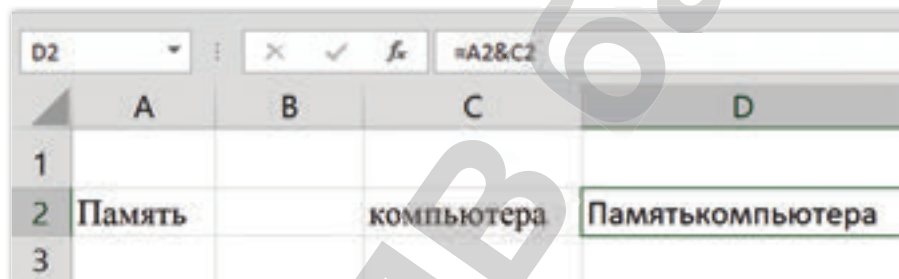
Операторы формул

В программе Excel операторы делятся на бинарные и унарные. Бинарные операторы работают с двойным значением. Например, оператор «*» умножает левое число на правое.

Унарные операторы работают с одним значением. Например, если «+» остается без изменений, то «-» возвращает значение числа справа.

Операторы, объединяющие два текста

Оператор «&» (амперсанд) используется для «склеивания» двух текстовых ячеек. Например, в ячейку A2 введен текст «Память», в ячейку C2 ввели текст «компьютера» и в ячейку D2 написали формулу «=A2&C2». В результате в ячейке D2 появится приклеенный текст «Памятькомпьютера» (рис. 92).



	A	B	C	D
1				
2	Память		компьютера	Памятькомпьютера
3				

Рис. 92. Использование оператора «&» в Строчке формул

Арифметические операторы

- «+» – сложение (например: «=5+5»);
- «-» – вычитание (например: «=5-5»);
- «*» – умножение (например: «=3*5»);
- «/» – деление (например: «=6/3»);
- «^» – возведение в степень (например: «=2^10»);
- «%» – процент (например: «=3%» преобразование 0,03) и др.

Логические операторы

- «>» – больше;
- «<» – меньше;
- «>=» – больше или равно;
- «<=» – меньше или равно;
- «=» – равно;
- «<>» – не равно (проверяет неравенство).

Справочные операторы

- «:» (двоеточие). Две ячейки ставятся между ссылками. Эти комбинации являются ссылками диапазона (A1: A15);
- «;» (точка с запятой). Объединяет ссылки нескольких ячеек в одну (SUM (A1: A15; B1: B15));
- () (пробел). Оператор пересечения сборов. Используется для ссылки на общие ячейки двух диапазонов (B7:D7_C6:C8).

С помощью постоянных и арифметических операторов можно создать простую формулу, как в *таблице 6*.

Таблица 6. Создание простых формул

Математическое написание	Описание в Excel
$x+25$	<code>=(x+25)</code>
$1+x^2$	<code>=(1+(x^2))</code>
$\frac{x}{2+x}$	<code>=(x/(2+x))</code>
$1 + \frac{x}{2 + \frac{x}{2+x}}$	<code>=(1+(x/(2+(x/(2+x))))))</code>
$(x^2+2xy+y^2)$	<code>=((x^2)+(2*x*y)+(y^2))</code>
$x^2 + y^2$	<code>=(x^2)+(y^2)</code>
$x^2 + y^2 + 1 + x$	<code>=(x^2)+(y^2)+1+x</code>

В программе Excel создадим простую формулу с помощью арифметических операторов.

1. Дана формула $x^2 + y^2$.
2. Вставьте в ячейку A1 – x и в ячейку B1 – y , а в ячейку A2 – значение x , в B2 – значение y .

	A	B	C
1	x	y	
2	10	20	
3			

3. В ячейку C2 введите формулу, описываемую в программе Excel: $=((A2^2)+(B2^2))$, где ссылки A2 и B2 являются значениями x и y .

	A	B	C	D	E	F
1	x	y				
2	10	20	500			
3		x^2+y^2				
4						

4. При изменении значений x и y созданная формула автоматически пересчитывает результат.

	A	B	C	D	E	F
1	x	y				
2	50	60	6100			
3		x^2+y^2				
4						

Создадим формулу в программе Excel с помощью логических операторов.

Пример 1.

1. Дано $15 > 29$.
2. В ячейку B2 электронной таблицы введите формулу $=15 > 29$. Выводится значение ЛОЖЬ, потому что программа автоматически выдает истину или ложь и принимает его в качестве логического оператора.

	A	B	C	D	E
1					
2		ЛОЖЬ			
3					

Пример 2.

1. Даны логические операторы в виде $15x > 29 - y$. В данном случае нужно создать ссылки для значений x и y .
2. В электронную таблицу введите формулу $=15 * x > 29 - y$.

	A	B	C	D	E	F
1	X	Y				
2	5	8		ИСТИНА		
3						

3. При изменении значений x и y программа автоматически пересчитывает результат.

	A	B	C	D	E	F
1	X	Y				
2	1	2		ЛОЖЬ		
3						

1

Отвечаем на вопросы

1. Что такое формула?
2. Что такое Строка формул?
3. Какую функцию выполняет Строка формул?
4. Какие операторы формулы используются в электронной таблице?
5. Как вы понимаете использование ссылок в формулах?
6. Как создать формулу для вычислений в электронной таблице?

2

Думаем и обсуждаем

1. Почему в электронной таблице для вычислений используются формулы?
2. В чем заключается эффективность применения формул для вычислений в электронной таблице?

3. Можем ли мы производить вычисления в электронной таблице без формул?

3

Анализируем и сравниваем

Проанализируйте описание математических выражений в Excel. Приведите примеры и сравните.

4

Выполняем в тетради

Напишите в тетради описание математических выражений в программе Excel:

Математическое написание	Описание в Excel
$2x^2 - 3y + 1$	
$\frac{x+1}{x} + 2x$	
$\frac{1}{2}x + 6y + c$	
$\frac{1}{4}x + \frac{2}{3}y + x^2$	
$4(x+3y) + 9\frac{1}{3}$	
$3\frac{3}{4}x \cdot \left(\frac{1}{3} + 0,4\right)$	
$10 - \frac{8x^2 + 9}{7}$	
$\frac{4}{9}y + \frac{5}{9}y - \frac{1}{3}y$	

5

Выполняем на компьютере

1. Найдите значения выражений в Excel, придавая переменным x , y и z различные значения. Проверьте полученные результаты:

1	$3,4x + y$
2	$(x + y)z$
3	$0,8x + 0,9y - xy$
4	$(x + z)y + 0,1x$
5	$(x - z)z + yz$
6	$\frac{30(x^3 - \frac{5}{6}y)}{5(x^3 - 6y)}$
7	$\frac{5x^{3y} - 2}{10xy} \cdot \frac{12x + y}{3x - y^{5-2x}}$

2. С помощью логических операторов создайте формулы.

6

Делимся мыслями

Что нового вы узнали на уроке? Чему научились? Поделитесь своими мыслями с друзьями. В каких условиях можно использовать в повседневной жизни полученные на уроке знания и навыки? Приведите примеры.

7

Домашнее задание 

Напишите описание данных операторов:

Арифметические операторы		Логические операторы	
«+»		«>»	
«-»		«<»	
«*»		«> =»	
«/»		«< =»	
«^»		«=»	
		«<>»	

§ 15. Условное форматирование

Вспомните!

- Что такое формула?
- Что такое Строка формул?
- Какие операторы вы знаете?

Вы узнаете:

- об условном форматировании;
- о параметрах условного форматирования;
- как применять условное форматирование в электронной таблице.

Форматирование –
Форматтау – Formatting

Условное форматирование –
Шартты форматтау –
Conditional formatting

Дополнительные функции –
Қосымша функциялар –
Additional functions

Для удобного и визуального отображения данных в электронной таблице используется условное форматирование. Это один из самых полезных инструментов Excel, который поможет сэкономить пользователю много времени и сил.

Условное форматирование – это инструмент для выделения значений таблицы по своим условным параметрам.

Инструмент Условное форматирование позволяет визуально отображать преимущества и особенности данных. Отсюда и вид визуализации значений ячеек.

Как используется инструмент Условное форматирование в электронной таблице?

Для того чтобы открыть инструмент **Условное форматирование** электронной таблицы, нужно выбрать **Главная** в поле меню, из ряда инструментов – раздел **Стили** – инструмент **Условное форматирование**.

В открывшемся окне вы увидите параметры инструмента Условное форматирование.

К ним относятся:

- Правила выделения ячеек;
- Правила отбора первых и последних значений;
- Гистограммы;
- Цветовые шкалы;
- Наборы значков (рис. 93).

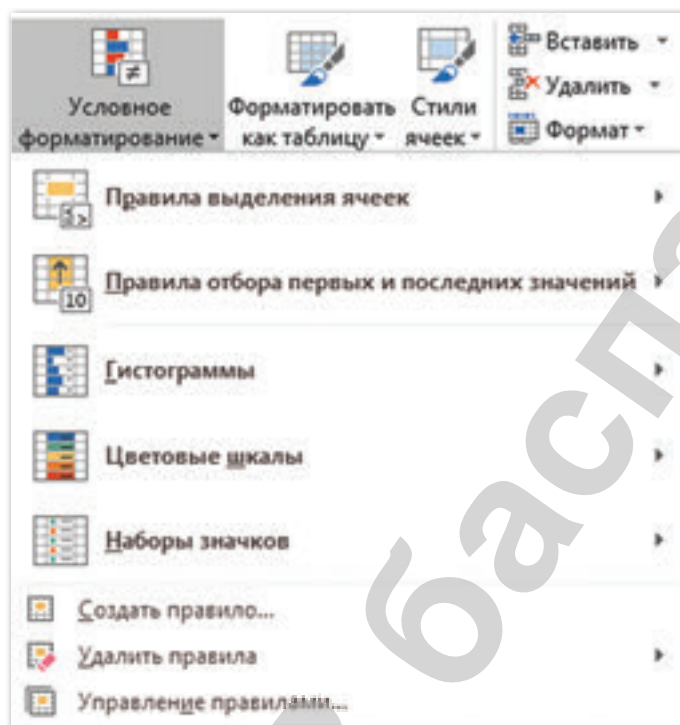


Рис. 93. Окно Условное форматирование

Теперь ознакомимся со списком параметров инструмента **Условное форматирование**.

Первый параметр – **Правила выделения ячеек**, в котором сопоставляются значения ячеек между собой и выделяются по большим, малым или равным значениям (рис. 94). Например, в диапазоне ячеек A1:A10 заданы цифры от 1 до 10. Выделяем заданный диапазон ячеек и в меню выделения ячеек инструмента **Условное форматирование** выбираем кнопку «Больше...». В открывшемся диалоговом окне щелкните по кнопке **Изменить**, после чего введите команду для окрашивания чисел

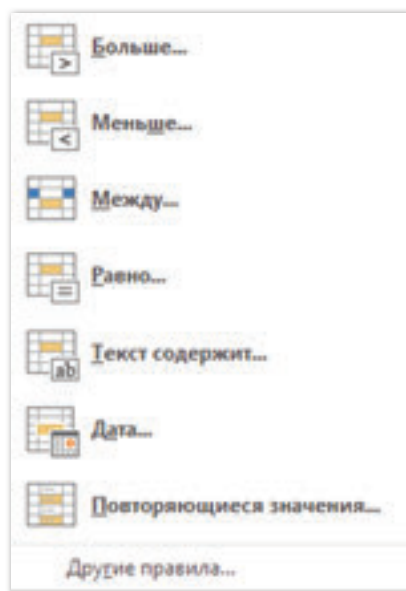

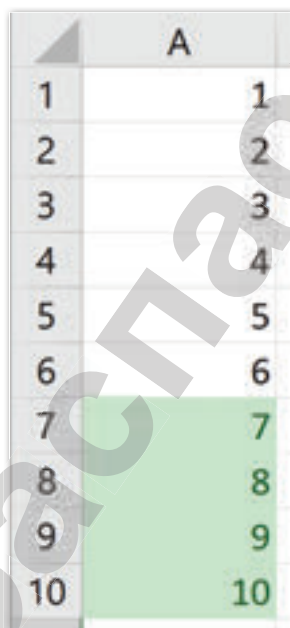


Рис. 94. Меню параметра Правила выделения ячеек

больше 6 в зеленый цвет. В результате ячейки больше 6 диапазона A1:A10 выделяются зеленым цветом (рис. 95).

Следующий параметр инструмента **Условное форматирование** – **Правила отбора первых и последних значений** (рис. 96). Инструмент позволяет выделить содержимое значений в ячейке по первому или последнему значениям разным цветом. Например, в диапазоне ячеек A1:A10 заданы цифры от 1 до 10. Из меню параметра **Правила отбора первых и последних значений** выбираем кнопку  **Последние 10 элементов...**. Выделяем диапазон ячеек A1:A10 и даем команду раскрасить последние 4 элемента в розовый цвет. В результате выбираются и выделяются последние 4 элемента из диапазона ячеек A1:A10 (рис. 97).



	A
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Рис. 95. Результат кнопки «Больше ...»

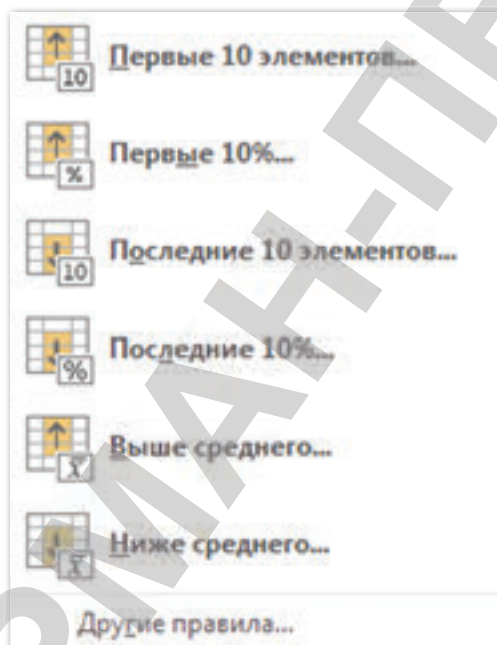
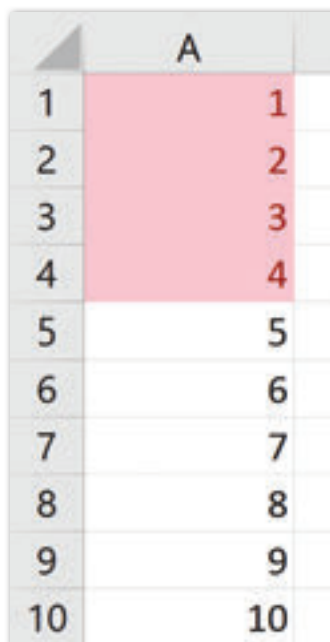


Рис. 96. Меню параметра Правила отбора первых и последних значений



	A
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Рис. 97. Результат кнопки «Последние 4 элемента...»

Параметр Гистограмма позволяет добавлять цветную гистограмму в ячейку. Длина столбца зависит от значения в ячейке. Длинный столбец означает большое значение (рис. 98–99).

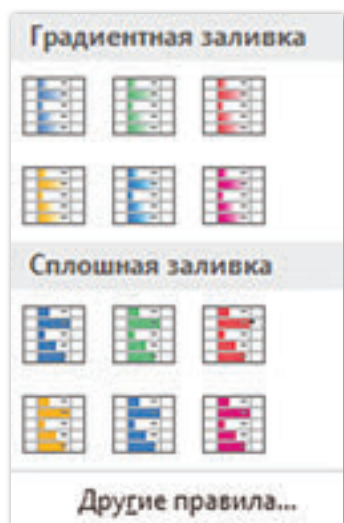


Рис. 98. Параметры гистограммы

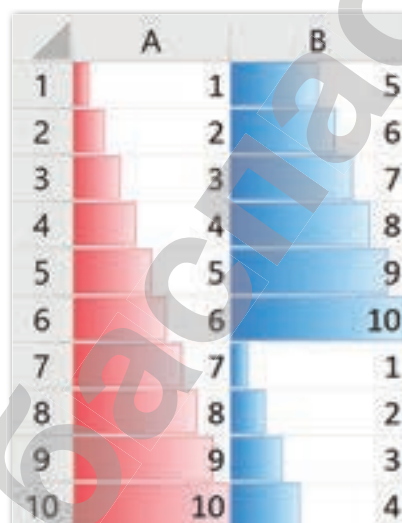
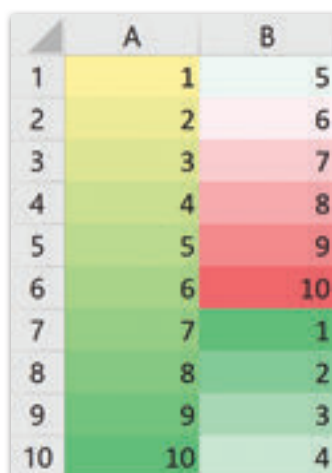


Рис. 99. Вид градиента

Следующий параметр инструмента **Условное форматование – Цветовые шкалы**. Он позволяет использовать цветные градиенты в диапазонах ячеек. Оттенок цвета зависит от величины значений в диапазоне ячеек (рис. 100).



Рис. 100. Параметр Цветные шкалы



Следующий параметр – **Наборы значков**. Он отображает указатели значений выбранных ячеек с помощью набора значков (рис. 101).

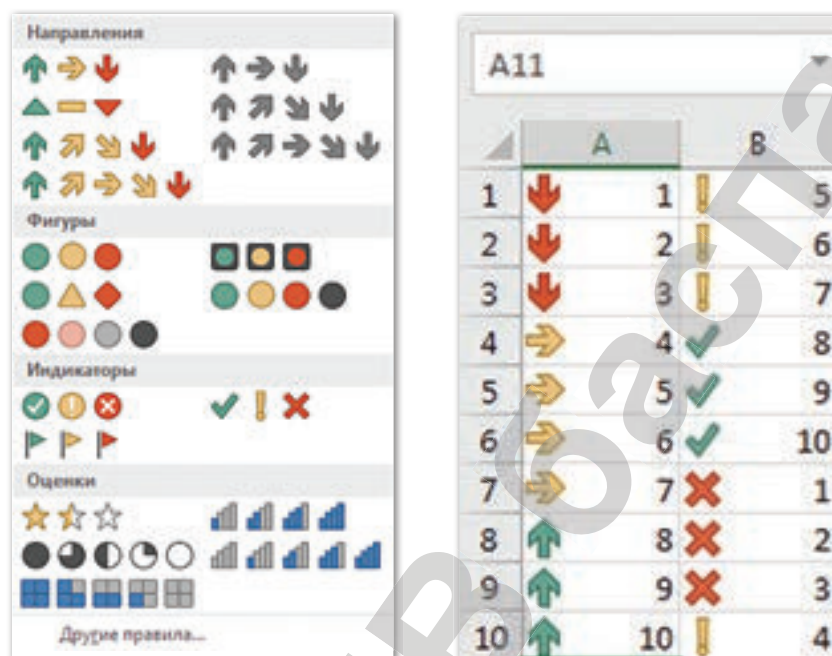


Рис. 101. Параметр Наборы значков

Инструмент **Условное форматирование** имеет ряд дополнительных функций.

Функция **Создать правило...** – предлагает пользователю условное форматирование выбранной ячейки.

Функция **Удалить правила** – позволяет пользователю удалить введенные правила условного форматирования с ячейки.

Функция **Управление правилами...** – позволяет управлять всеми правилами (создание, изменение, просмотр и удаление) на рабочем листе с помощью диспетчера правил условного форматирования.

1

Отвечаем на вопросы

1. Что такое условное форматирование?
2. Как открыть инструмент Условное форматирование?
3. Из каких параметров состоит инструмент Условное форматирование?

4. Что такое гистограмма?
5. Какие дополнительные функции имеет инструмент Условное форматирование?

2

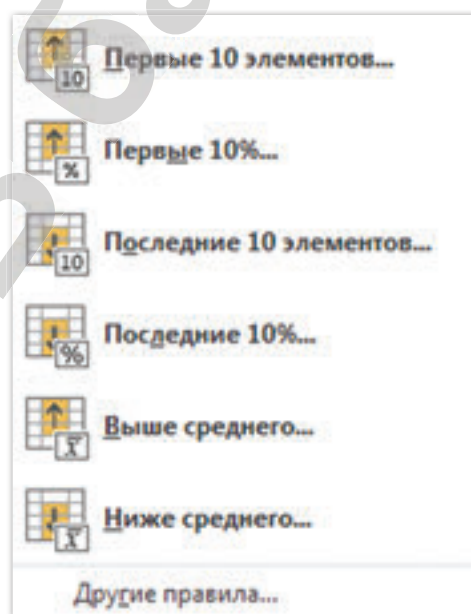
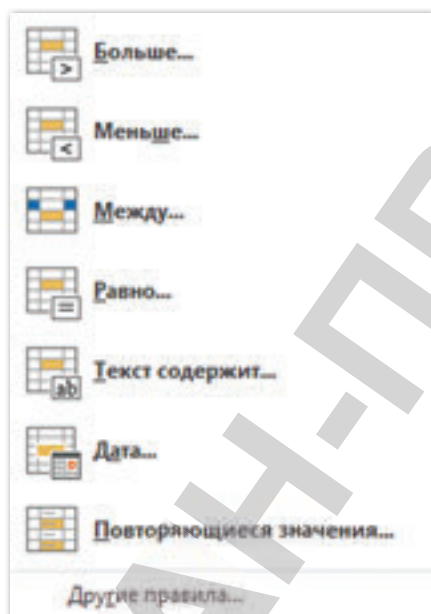
Думаем и обсуждаем

1. Для чего используется инструмент Условное форматирование?
2. В чем заключается эффективность инструмента Условное форматирование?

3

Анализируем и сравниваем

Сравните параметры «Правила выделения ячеек» и «Правила отбора первых и последних значений» по диаграмме Венна. Какие особенности имеет каждый из них?



4

Выполняем в тетради

Запишите дополнительные функции инструмента Условное форматирование.

5

Выполняем на компьютере

1. Отформатируйте таблицу «Список рек протяженностью более 1000 км на территории Казахстана»

с помощью параметров инструмента Условное форматирование.

- 1) Отформатируйте ячейки 2-го столбца с помощью параметра Правила выделения ячеек:
 - Числа меньше 2000 – красным цветом.
 - Числа между 2000–3000 – зеленым цветом.
 - Числа больше 3000 – желтым цветом.
- 2) Отформатируйте ячейки 3-го столбца с помощью параметра Правила отбора первых и последних значений:
 - Первые числа больше 3 – зеленым цветом.
 - Последние числа ниже 2 – красным цветом.
 - Числа выше среднего уровня – желтым цветом.

*Список рек протяженностью более 1000 км
на территории Казахстана*

Реки	Длина (км)	В Казахстане (км)
Иртыш	4 248	1 700
Есиль	2 450	1 400
Жайык	2 428	1 082
Сырдария	2 219	1 400
Тобыл	1 591	800
Иле	1 439	815
Шу	1 186	800

2. Отформатируйте таблицу «Список первых 10 мест по результатам международных студенческих и молодежных спортивных соревнований «Зимняя Универсиада 2017» с помощью параметров форматирования Гистограмма, Цветные шкалы и Наборы значков.
 - В 3-м столбце применить параметр Гистограмма.
 - В 4-м столбце применить параметр Цветные шкалы.
 - В 5-м столбце применить команду Индикаторы из меню параметра Наборы значков.
 - В 6-м столбце применить команду Направления из меню параметра Наборы значков.

Список первых 10 мест по результатам международных студенческих и молодежных спортивных соревнований «Зимняя Универсиада 2017»

Место	Страны	Золото	Серебро	Бронза	Всего
1	 Россия	29	27	15	71
2	 Казахстан	11	8	17	36
3	 Республика Корея	11	5	5	21
4	 Япония	6	12	10	28
5	 Польша	5	2	5	12
6	 Китай	4	4	2	10
7	 Франция	4	2	2	8
8	 Италия	4	0	0	4
9	 Беларусь	3	2	1	6
10	 Украина	2	3	4	9

6

Делимся мыслями

Что нового вы узнали на уроке? Чему научились? Поделитесь своими мыслями с друзьями. В каких условиях можно использовать в повседневной жизни полученные на уроке знания и навыки? Приведите пример.

7

Домашнее задание

Запишите параметры инструмента Условное форматирование:

Параметры
инструмента
Условное
форматирование

§ 16. Графическое представление табличных данных

Вспомните!

- Что такое условное форматирование?
- Какие параметры имеет инструмент Условное форматирование?
- В чем заключается эффективность гистограммы?

Вы узнаете:

- что такое диаграмма;
- как создать диаграмму в электронной таблице.

Диаграмма – Диаграмма – Charts

Типы диаграмм – Диаграмма түрлери – Chart types

Создание диаграмм – Диаграммалар құру – Creating charts

С помощью электронных таблиц можно не только быстро и качественно обработать большие объемы числовой информации, но и сделать числовые данные более наглядными за счет их графического представления.

Основным средством графического представления табличных данных являются диаграммы.

Что такое диаграмма?

Диаграмма – графическое изображение, наглядно демонстрирующее соотношение каких-либо величин. Диаграмма создается на основе любых данных. Из одного цифрового источника можно создать несколько типов диаграмм.

Какие типы диаграмм существуют?

Для создания диаграммы в табличном процессоре необходимо, прежде всего, ввести используемые данные. В зависимости от введенных данных можно выбрать диаграмму. Чтобы ознакомиться с типами диаграммы, необходимо выделить используемые данные и в строке Меню ⇒ Вставка ⇒ Панель инструментов и выбрать раздел Диаграммы (рис. 102).



Рис. 102. Раздел Диаграммы

В открывшемся окне можно просмотреть **Все диаграммы** и **Рекомендуемые диаграммы**. В зависимости от вида, диаграммы используются для различных целей (рис. 103).





Рис. 103. Виды дополнительных диаграмм

В списке **Все диаграммы** можно увидеть виды встроенных графиков, которые можно использовать, дважды щелкнув мышью (табл. 7).

Таблица 7. Виды и типы диаграмм

Внешний вид диаграммы	Типы диаграмм	Внутренний вид
	Гистограмма	
	Графика	
	Круговая	
	Линейчатая	
	С областями	
	Точечная	
	Биржевая	
	Поверхность	
	Лепестковая	
	Древовидная	

Внешний вид диаграммы	Типы диаграмм	Внутренний вид
	Солнечные лучи	

Из каких элементов состоит диаграмма?

Диаграмма обычно состоит из стандартных элементов. Некоторые из них отображаются по умолчанию. При необходимости можно добавить другие (рис. 104).

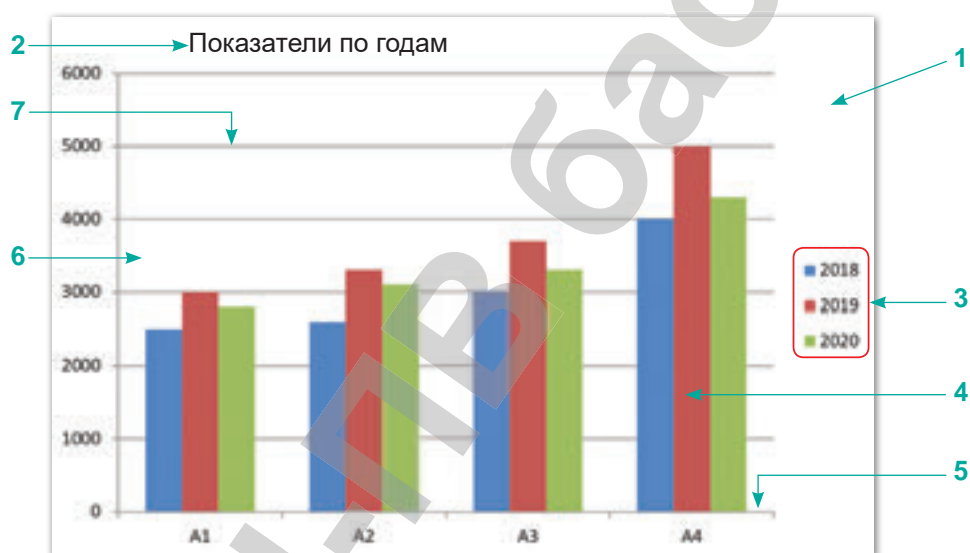


Рис. 104. Элементы диаграммы

Цифрами на рисунке обозначены следующие элементы диаграммы:

1. Область построения диаграммы
2. Название диаграммы
3. Легенда диаграммы
4. Ряд данных
5. Горизонтальная ось диаграммы
6. Вертикальная ось диаграммы
7. Сетка диаграммы

Кроме того, можно использовать элемент описания данных (Подписи данных).

Как построить диаграмму?

Построение и создание диаграммы выполняется в следующем порядке:

1. Выделите в таблице диапазон данных, по которым будет строиться диаграмма, включая, если это возможно, и диапазоны подписей к этим данным по строкам и столбцам.
2. Выбрать Строка меню ⇒ Вставка ⇒ Панель инструментов ⇒ раздел Диаграммы (рис. 105).

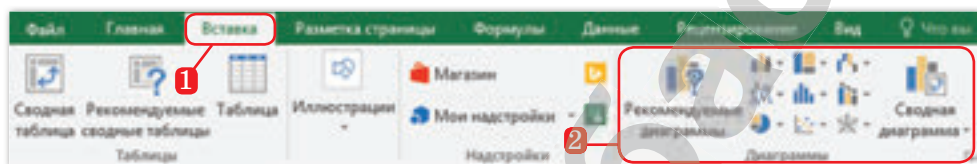


Рис. 105. Раздел Диаграммы

3. В разделе Диаграммы представлены только несколько типов графиков, предназначенных для быстрого выбора. Чтобы открыть дополнительные типы диаграмм, нажмите кнопку в правом нижнем углу и просмотрите **Рекомендуемые диаграммы** и **Все диаграммы**.
4. Выберите необходимую диаграмму и нажмите на кнопку ОК.
5. При выборе типа диаграммы следует учитывать частоту отношений между данными, так как неверно выбранный тип диаграммы приводит к неправильному отображению информации.

Как изменить диаграмму?

Созданную диаграмму можно изменить и обработать в любое время. Для внесения изменений:

1. Выделить построенную диаграмму.
2. Можно также редактировать с помощью трех кнопок (**Элементы диаграммы**, **Стиль диаграммы**, **Фильтр диаграммы**) в правом углу диаграммы, которые используются для добавления элементов, изменения внешнего вида или настройки шкалы диаграммы (рис. 106).

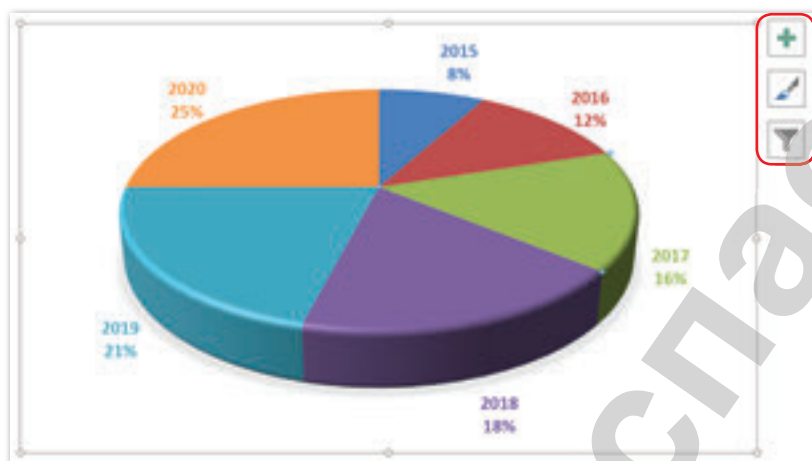


Рис. 106. Кнопки редактирования

3. Доступ к функциям дополнительной обработки выполняется с помощью панели инструментов разделов «Конструктор» и «Формат» в Строке меню (рис. 107–108).



Рис. 107. Панель инструментов раздела «Конструктор»



Рис. 108. Панель инструментов раздела «Формат»

4. С помощью панели инструментов раздела «Конструктор» можно преобразовать элементы диаграммы, изменить стиль диаграммы, диапазоны данных, расположение диаграммы.
5. На панели инструментов раздела «Формат» используются установка текущего фрагмента, ввод дополнительных фигур, отображение области выбора.

1

Отвечаем на вопросы

1. Что такое диаграмма?
2. Какие типы диаграмм вы знаете?
3. Из каких элементов состоит диаграмма?
4. Как создается диаграмма?
5. Как можно изменить и дополнить построенную диаграмму?

2

Думаем и обсуждаем

1. Для чего используются диаграммы?
2. В чем заключается эффективность диаграммы?
3. В какой сфере часто используются диаграммы? Объясните, почему.

3

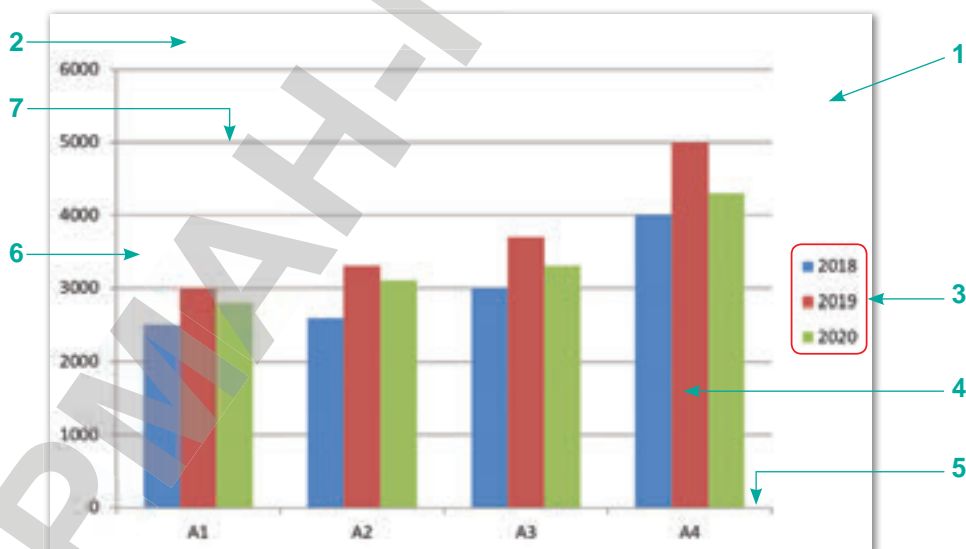
Анализируем и сравниваем

Сравните разделы «Конструктор» и «Формат». Проанализируйте, какой раздел имеет наибольшую производительность обработки.

4

Выполняем в тетради

1. Напишите в тетрадях порядок создания диаграммы.
 1. ...
 2. ...
 3. ...
2. Напишите в тетрадях описание элементов диаграммы:



5

Выполняем на компьютере

1. Приведите примеры по внешнему виду следующих графиков и постройте диаграмму:



2. По результатам статистики 2019 года выявлен список смартфонов, которые были наиболее продаваемыми по всему миру. Постройте диаграмму по данным показателям:

iPhone XR – 46,3 млн штук	iPhone 11 – 37,3 млн штук	Samsung Galaxy A10 – 30,3 млн штук	Samsung Galaxy A50 – 24,2 млн штук
Samsung Galaxy A20 – 19,2 млн штук	iPhone 11 Pro Max – 17,6 млн штук	iPhone 8 – 17,4 млн штук	Xiaomi Redmi Note 7 – 16,4 млн штук

6

Делимся мыслями

Можно ли использовать способы построения диаграммы на других уроках? Обоснуйте свои ответы.

7

Домашнее задание

Составьте ребус на слово «Диаграмма» и придумайте синквейн с этим словом.

§ 17. Практикум. Моделирование процессов в электронных таблицах

Цель работы: Преобразование и моделирование цифровых данных с помощью элементов электронной таблицы.

Инструкция по выполнению задания:

1. Используйте элементы электронной таблицы.
2. Определите типы вводимых данных.
3. Соедините строки с адресами ячеек и формулами.
4. По необходимости используйте операторы формул.
5. С помощью условного форматирования визуальнo преобразуйте результаты.
6. Смоделируйте данные с помощью типов диаграммы.

Уровень А

Задание 1. Форматирование элементов электронной таблицы. Заполните приведенную ниже таблицу Пифагора (таблицу умножения) с помощью функции Автозаполнение:

*	1	2	3	4	5	6	7	8	9	10
1	1	2	3							
2	2	4	6							
3	3	6	9							
4	4	8	12							
5	5	10	15							
6	6	12	18							
7	7	14	21							
8	8	16	...							
9	9	18	...							
10	10	20	...							

Задание 2. Приведите примеры, чтобы отличить тип данных в таблице:

Формула	Текст	Число
...
...
...

Уровень В

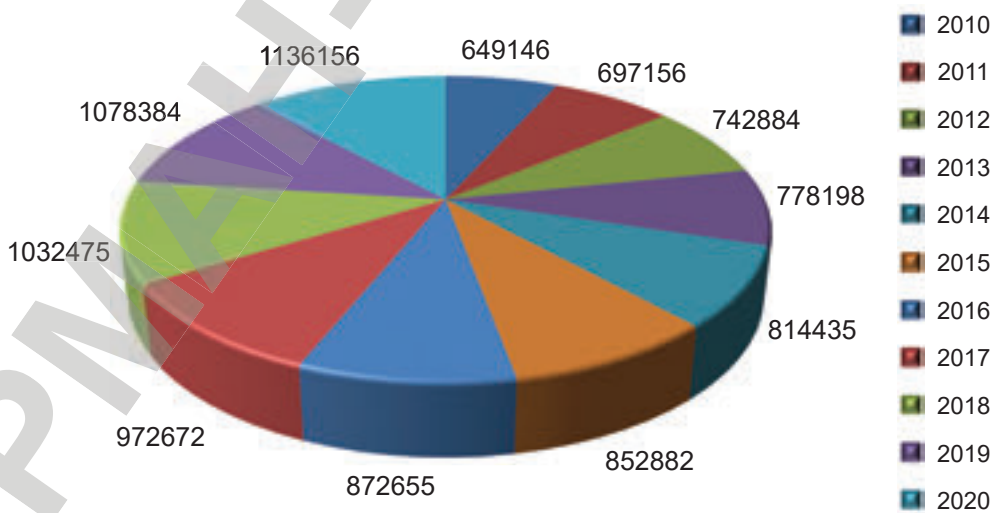
Дополните пустые графы в таблице:

Описание в программе Excel	Математическое написание
	$(x+y)(x^2 - xy + y^2)$
$= ((15*(x^2)) - ((7/12)*y))/(18*y) + (x^2)$	
	$2 \frac{80(x^3 - \frac{5}{6}y)}{5(x^3 - 6y)}$
$= ((x^3) - (y^2)) + ((1+x) + (x^{(3/2)}))$	
	$\frac{40y^3 + \frac{4}{x^2}x}{6x^2 - 18xy}$
$= ((2*(x^2)) - (3*x*y))/((x-3*y)^2) - ((7*x-4*y)^2)$	
	$x^2 + 4x + \frac{6}{8}y$

Уровень С

Задание 1. Опишите модель диаграммы:

Численность населения города Нур-Султан за последние 10 лет



Задание 2. Составьте диаграмму на основе данной информации:

Жансая Абдумалик – казахстанская шахматистка, международный мастер (2017) и гроссмейстер среди женщин (2014).

В 2007 году приняла участие в чемпионате Казахстана среди девушек до 8 лет, в котором заняла первое место.

В 2008 году в Иране – первое место на чемпионате Азии.

В 2010 году в Турции – первое место в турнире среди школьников в своей возрастной категории.

В 2011 году в Польше вновь завоевала первое место, став обладательницей титула сильнейшей шахматистки мира среди школьников.

В 2013 году в Турции заняла второе место на чемпионате мира среди девушек до 20 лет.

В 2014 году в ОАЭ заняла третье место на чемпионате Азии по блицу.

В 2015 году в Ханты-Мансийске заняла третье место на чемпионате мира среди девушек до 20 лет.

В 2017 году в Италии – заняла первое место, став чемпионкой мира среди девушек до 20 лет.

В 2021 году в Гибралтаре получила титул мужского международного гроссмейстера.

Задание 3. Приведите в таблице итоговые оценки учащихся за I четверть и составьте диаграмму:

№	ФИО ученика	Предмет 1	Предмет 2	Предмет 3	Средний балл ученика
1
2
3
4
Средний балл класса					...

ЗАДАНИЯ ПО СУММАТИВНОМУ ОЦЕНИВАНИЮ К РАЗДЕЛУ 3

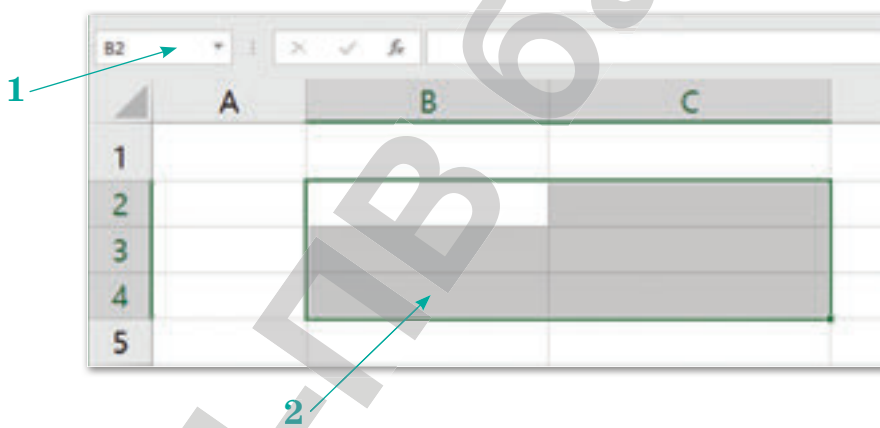
1. Прикладная программа для работы с электронными таблицами:

- A. Microsoft Excel
- B. Microsoft Word
- C. Microsoft Access
- D. PowerPoint
- E. Paint

2. Заполните пропуски:

Ряд столбцов – ... выделяются комбинациями, а строки – ... состоят из списка.

3. Запишите названия указанных элементов электронной таблицы.



4. Выполните шаги для открытия панели «Форматирование ячеек»:

- 1. ...
- 2. ...
- 3. ...

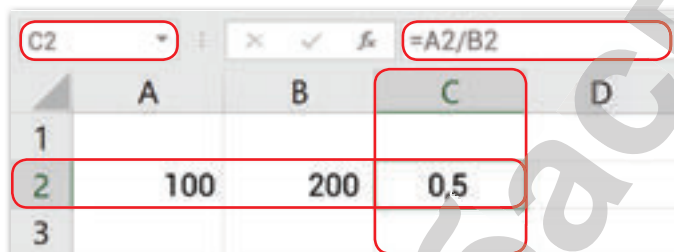
5. Панель, используемая для просмотра описания текста и формул, включаемых в ячейки листа:

- A. Формула
- B. Ячейка
- C. Диапазон
- D. Таблица
- E. Строка формулы.

6. Написание формулы начинается со знака ...

- A. «&»
- B. «;»
- C. «,»
- D. «=».
- E. «.»

7. Опишите картинку:



8. Запишите математическое написание формулы $((x^2)+(y^2))$.

9. Напишите назначение кнопки



10. Распишите шаги обработки/изменения диаграммы:

- 1. ...
- 2. ...
- 3. ...

ИТОГИ 3 РАЗДЕЛА

В разделе «Решение задач с помощью электронных таблиц» вы узнали, как форматировать таблицы в текстовом редакторе, как использовать элементы электронной таблицы, различные типы данных, условное форматирование в электронной таблице, создавать формулы для вычислений и диаграммы в электронной таблице.

Кроме того, вы выполняли практические работы по созданию таблиц в текстовом процессоре, а также по типам форматирования и основным элементам электронной таблицы.

Особенность электронных таблиц заключается в возможности применения формул для описания связи между значениями различных ячеек. Расчет по заданным формулам выполняется автоматически. Изменение содержимого какой-либо ячейки приводит к пересчету значений всех ячеек, которые с ней связаны формульными отношениями и, тем самым, к обновлению всей таблицы в соответствии с изменившимися данными.

Применение электронных таблиц упрощает работу с данными и позволяет получать результаты без проведения расчетов вручную или специального программирования. Наиболее широкое применение электронные таблицы нашли не только в экономических и бухгалтерских расчетах, но и в научно-технических задачах электронные таблицы можно использовать эффективно.

ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ НА ЯЗЫКЕ PYTHON

Изучив раздел, вы научитесь:

- осуществлять чтение и запись файла на языке программирования Python (пайтон);
- записывать разветвляющиеся алгоритмы на языке программирования Python (пайтон);
- использовать вложенные условия на языке программирования Python (пайтон);
- использовать составные условия на языке программирования Python (пайтон).

§ 18. Работа с файлами

Вспомните!

- Как форматировать элементы электронных таблиц?
- Что такое типы данных?
- Что такое условное форматирование?
- Что такое графическое представление табличных данных?
- Как можно моделировать процессы в электронных таблицах?

Вы узнаете:

- как открывать файл на языке программирования Python;
- как закрывать файл на языке программирования Python;
- как читать и записывать файлы на языке программирования Python.

Файл – *Файл* – File

Свойства файла – *Файлдың қасиеттері* – File properties

Чтение файла – *Файлды оқу* – Read file

Запись файла – *Файлды жазу* – Write file

Что такое файл?

Файл в Python (англ. *file*) – это набор данных, сохраненный в виде последовательности битов на компьютере.

Работа с файлами на языке программирования Python

На языке программирования Python существует два типа файлов: текстовые и бинарные.

Текстовые файлы

В текстовых файлах хранятся последовательности символов, которые понимает человек. Блокнот и другие стандартные редакторы умеют читать и редактировать этот тип файлов.

Текст может храниться в двух форматах: `.txt` – простой текст и `.rtf` – формат обогащенного текста.

Бинарные файлы

Бинарный файл – это любой файл на вашем компьютере. Все данные (изображения, звуки, сжатые версии других файлов, любой тип содержимого файла) на компьютере и связанных с ними носителях записаны в битах. В бинарных файлах данные отображаются в закодированной форме (с использованием только нулей (0) и единиц (1) вместо простых символов). В большинстве случаев это просто последовательности битов. Они хранятся в формате `.bin`.

Любую операцию с файлом можно разбить на три этапа: 1) открытие файла; 2) выполнение операции (запись, чтение); 3) закрытие файла.

Открытие файла. Метод `open()`

На языке программирования Python есть встроенная функция `open()`. С ее помощью можно открыть любой файл на компьютере. Технически Python создает на его основе объект.

Синтаксис для открытия файла:

```
f = open(file_name, access_mode)
```

Где,

- `file_name` – имя открываемого файла;
- `access_mode` – режим открытия файла. Он может быть: для чтения, записи и т.д. По умолчанию используется режим чтения (`r`), если другое не указано (*табл. 8*).

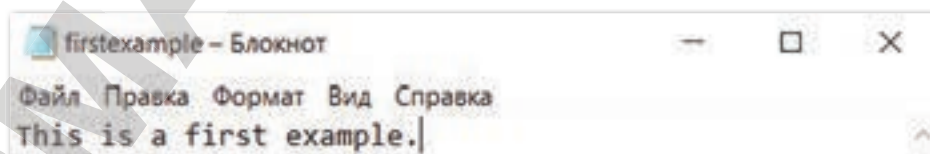
Таблица 8. Полный список режимов открытия файла

Режим	Описание
<code>r=read</code> opens for reading	Только для чтения
<code>w=write</code> opens for writing	Только для записи. Создаст новый файл, если не найдет с указанным именем
<code>rb – r=read, b=binary</code> open the binary file in read mode	Только для чтения (бинарный)
<code>wb – w=write, b=binary</code> open the binary file in write mode	Только для записи (бинарный). Создаст новый файл, если не найдет с указанным именем
<code>r+</code> <code>r=read</code> opens for reading and writing	Для чтения и записи
<code>rb+</code> <code>r=read, b=binary</code> open the binary file in reading and writing mode	Для чтения и записи (бинарный)

Режим	Описание
w+ w=write open for reading and writing	Для чтения и записи. Создаст новый файл для записи, если не найдет с указанным именем
wb+ w=write, b=binary open the binary file in writing and reading mode	Для чтения и записи (бинарный). Создаст новый файл для записи, если не найдет с указанным именем
a a=append opens for for appending	Откроет для добавления нового содержимого. Создаст новый файл для записи, если не найдет с указанным именем
a+ a=append open for reading and appending	Откроет для добавления нового содержимого. Создаст новый файл для чтения записи, если не найдет с указанным именем
ab a=append, b=binary opens a file for appending in binary mode	Откроет для добавления нового содержимого (бинарный). Создаст новый файл для записи, если не найдет с указанным именем
ab+ a=append, b=binary opens a file for both appending and reading in binary mode	Откроет для добавления нового содержимого (бинарный). Создаст новый файл для чтения записи, если не найдет с указанным именем

Пример 1.1.

Откроем Блокнот, создадим текстовый файл `firstexample.txt` и сохраним его.



Следующий код используется для открытия файла в режиме чтения:

```
File1 = open('firstexample.txt', 'r')
# открыть файл в режиме чтения
```

В этом примере `File1` – переменная – указатель на файл `firstexample.txt`.

Для открытия файла из любого каталога используется следующий код:

```
File2 = open('C:/firstexample.txt', 'r')
```

Для вывода содержимого файла используется следующий код:

```
>>> File1 = open('firstexample.txt', 'r')
>>> print(*File1)
This is a first example.
```

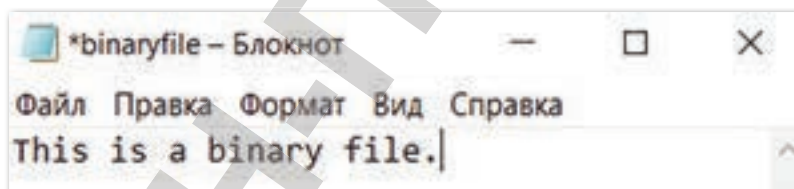
Для вывода информации о файле используется следующий код:

```
>>> print(File1)
<_io.TextIOWrapper name='firstexample.txt' mode='r'
encoding='cp1251'>
```

Стоит обратить внимание, что в Windows стандартной кодировкой является CP1251, а в Linux – UTF-08.

Пример 1.2.

Создадим бинарный файл `binaryfile.bin` и сохраним его.



Следующий код используется для открытия бинарного файла в режиме чтения:

```
File1 = open('binaryfile.bin', 'rb')
# Открыть бинарный файл для чтения
```

В этом примере `File1` – переменная – указатель на файл `binaryfile.bin`.

Для открытия бинарного файла из любого каталога используется следующий код:

```
File2=open('C:/binaryfile.bin', 'rb')
```

Следующий код используется для вывода содержимого бинарного файла и информации о нем.

```
>>> File2=open('C:/binaryfile.bin', 'rb')
>>> print(*File2)
b'This is a binary file'
```

Заккрытие файла. Метод `close()`

После открытия файла в Python его нужно закрыть. Таким образом освобождаются ресурсы. Python автоматически закрывает файл, когда объект присваивается другому файлу.

Существуют несколько способов для закрытия файла.

Способ 1.

Проще всего после открытия файла закрыть его, используя метод `close()`.

```
File1=open('firstexample.txt', 'r')
#
File1.close()
```

После закрытия этот файл нельзя будет использовать до тех пор, пока заново его не открыть.

Способ 2.

Также можно написать путь `try/finally`, который гарантирует, что если после открытия файла операции с ним приводят к исключениям, он закроется автоматически. Без него программа завершается некорректно.

Вот как сделать это исключение:

```
File1=open('firstexample.txt', 'r')
try:
    #
finally:
    File1.close()
```

Файл нужно открыть до инструкции `try`, потому что если инструкция `open` сама по себе вызовет ошибку, то файл не будет открываться для последующего закрытия.

Этот метод гарантирует, что если операции над файлом вызовут исключения, то он закроется до того, как программа остановится.

Чтение и запись файлов в Python

В Python файлы можно читать или записывать информацию в них с помощью соответствующих режимов.

Функция `read()`

Функция `read()` используется для чтения содержимого файла после открытия его в режиме чтения (`r`).

Синтаксис

```
file.read(size)
```

Где,

- `file` – объект файла;
- `size` – количество символов, которые нужно прочитать. Если это не указать, то файл будет прочитан целиком.

Пример 2.

Чтение 15 символов из файла `firstexample.txt`

```
>>> File1=open('firstexample.txt', 'r')
>>> File1.read(15)
'This is a first'
```

Интерпретатор прочитал 15 символов файла и, если снова использовать функцию `read()`, то чтение начнется с 16-го символа.

Для чтения следующих 8 символов применяется следующий код:

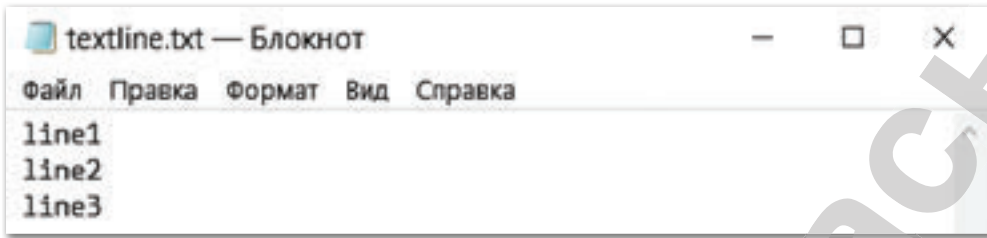
```
>>> File1.read(8)
'example'
```

Функция `readline()`

Функция `readline()` используется для построчного чтения содержимого крупных файлов. С ее помощью можно получать доступ к любой строке в любой момент.

Пример 3.

Создадим файл `textline.txt` с несколькими строками:



Посмотрим, как функция `readline()` работает в `textline.txt`. Для того чтобы прочитать первую строку, используется следующий код:

```
>>> File1=open('C:/textline.txt','r')
>>> File1.readline()
'line1\n'
```

Для того чтобы прочитать все строки, используется следующий код:

```
>>> File1=open('C:/textline.txt','r')
>>> File1.readlines()
['line1\n', 'line2\n', 'line3']
```

Функция `write()`

Функция `write()` используется для записи в файлы Python, открытые в режиме записи. Если пытаться открыть файл, которого не существует в этом режиме, то будет создан новый.

Синтаксис

```
file.write(string)
```

Пример 4.

Предположим, файла `writetext.txt` не существует. Он будет создан при попытке открыть его в режиме чтения.

```
>>> a=open("D:/writetext.txt","w")
>>> a.write("Hello Kazakhstan")
```

В результате появится запись в файле `writetext.txt`:



Переименование файлов в Python. Функция `rename()`

Функция `rename()` используется для переименования файлов в Python. Для ее использования сначала нужно импортировать модуль `os`.

Синтаксис

```
import os
os.rename(src, dest)
```

Где,

- `src` – файл, который нужно переименовать;
- `dest` – новое имя файла.

Пример 5.

```
>>> import os
>>> os.rename("D:/writetext.txt", "D:/secondexample.txt")
```

Текущая позиция в файлах Python

В Python возможно узнать текущую позицию в файле с помощью функции `tell()`. Таким же образом можно изменить текущую позицию командой `seek()`.

Пример 6.

Сначала открываем `secondexample.txt`, который мы переименовали ранее, и перейдем к 5-й позиции:

```
>>> f=open("D:/secondexample.txt")
>>> f.read(5)
'12345'
```

Для того чтобы вернуть текущую позицию, используется следующий код:

```
>>> f.tell()
5
```

Теперь вернем положение снова на 0:

```
>>> f.seek(0,0)
0
```

Таблица 9. Методы работы с файлами в Python

Методы	Описание
<code>file.close()</code>	Закрывает открытый файл
<code>file.fileno()</code>	Возвращает целочисленный дескриптор файла
<code>file.flush()</code>	Очищает внутренний буфер
<code>file.isatty()</code>	Возвращает <code>True</code> , если файл привязан к терминалу
<code>file.next()</code>	Возвращает следующую строку файла
<code>file.read(n)</code>	Читает первые <code>n</code> символы файла
<code>file.readline()</code>	Читает один ряд строки или файла
<code>file.readlines()</code>	Читает и возвращает список всех строк в файле
<code>file.seek(offset[,whene])</code>	Устанавливает текущую позицию в файле
<code>file.seekable()</code>	Проверяет, поддерживает ли файл случайный доступ. Возвращает <code>True</code> , если да
<code>file.tell()</code>	Возвращает текущую позицию в файле
<code>file.truncate(n)</code>	Уменьшает размер файл. Если указано <code>n</code> , то файл обрезается до <code>n</code> байт, если нет – до текущей позиции
<code>file.write(str)</code>	Добавляет строку <code>str</code> в файл
<code>file.writelines(sequence)</code>	Добавляет последовательность строк в файл

1

Отвечаем на вопросы

1. Что такое файл?
2. Какова функция файловой системы?
3. Какие свойства есть у файла?
4. Какие бывают режимы открытия файла?
5. Какие бывают режимы закрытия файла?
6. Какие бывают способы чтения и записи файла в Python?
7. Какие функции применяют для переименования файла?

2

Думаем и обсуждаем

1. Зачем нужна высокоорганизованная файловая система?
2. Почему важно закрыть открытый файл?
3. Определите методы работы с файлами на языке программирования Python.

3

Анализируем и сравниваем

Сравните между собой типы файлов на языке программирования Python.

	Текстовый файл	Бинарный файл
Определение		
Форматы файлов		
Режимы открытия файла		

4

Выполняем в тетради

1. Заполните таблицу описанием режимов открытия файла.

Режим	Описание	Режим	Описание
r		rb+	
a+		w	

2. Определите методы работы с файлами по их описанию.

Методы	Описание	Методы	Описание
	очищает внутренний буфер		чтение первых n символов файла
	добавляет строку в файл		добавляет последовательность строк в файл

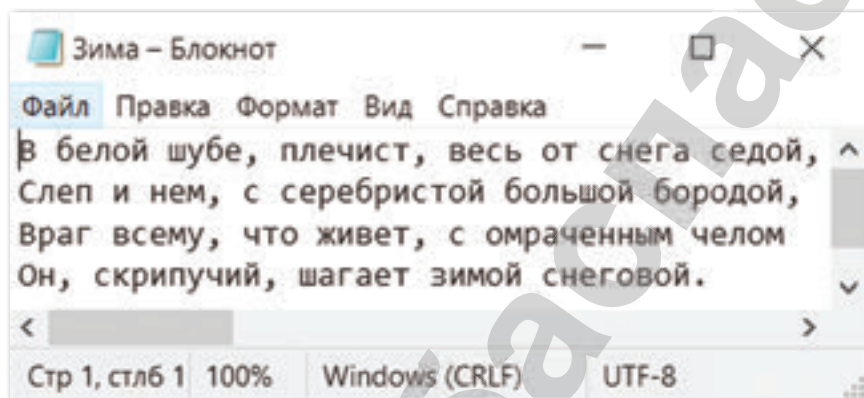
5

Выполняем на компьютере

1. Проверьте ответы примеров 1–6, приведенные в параграфе, с помощью программы Python.
2. Создайте текстовый файл из 5 строк и напишите программу, которая выводит каждую строку в файле.

3. Создайте пустой текстовый файл и напишите программу, которая записывает в него любые 4 строки стихотворения Абая Кунанбаева «Зима».

Например:



6

Делимся мыслями

Как вы думаете, можно ли использовать полученные знания по чтению и записи файла на языке программирования Python при решении каких-либо бытовых проблем в будущем?

7

Домашнее задание

В программе Блокнот наберите текст для 5 файлов под заголовком «Самая ... профессия».

Сохраните файлы под названиями «Самая сладкая профессия», «Самая зеленая профессия», «Самая смешная профессия», «Самая успешная профессия», «Самая умная профессия» (названия файлов можно изменить на другие профессии).

Используя режим для открытия файла, сравните типы профессий, приведенные в каждом файле, с типами профессий, приведенными одноклассниками и определите список «Самых ... профессий».

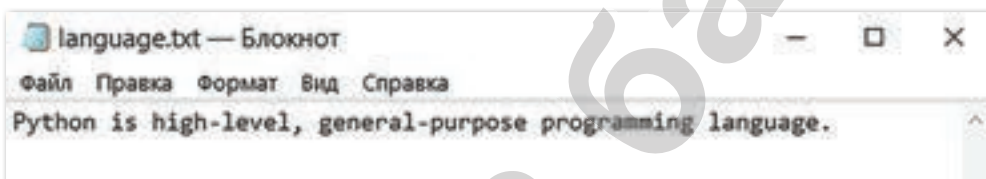
§ 19. Практикум. Работа с файлами

Уровень А

Задание 1. Создание текстового файла.

1. Нажмите на кнопку Пуск и откройте Блокнот из ряда Стандартные.
2. Сохраните текстовый файл с именем `language.txt` на диске `D:\`.
3. Введите какую-либо строку в созданном текстовом файле.

Например:



4. Снова сохраните файл после ввода текста.
5. Откройте файл `language.txt` в режиме чтения в программе Python.
6. В открывшемся окне программы Python введите следующие строки и нажмите клавишу Enter:

```
>>> f = open('D:/language.txt', 'r')
>>> print(*f)
```

7. В результате файл `language.txt` откроется в режиме чтения и внесенная ранее строка отобразится на экране.

```
>>> f = open('D:/language.txt', 'r')
>>> print(*f)
Python is high-level, general-purpose programming
language.
```


Задание 2. Вывод информации о файле.

1. Для того чтобы получить информацию о файле `language.txt`, сформированный в задании 1, введите следующие строки и нажмите клавишу Enter:

```
>>> f = open('D:/language.txt', 'r')
>>> print(f)
```

2. Заполните таблицу информацией о файле:

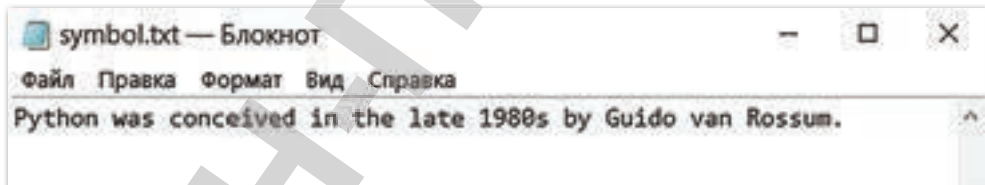
name	mode	encoding

Уровень В

Задание 1. Чтение текстового файла.

1. Нажмите на кнопку Пуск и откройте Блокнот из ряда Стандартные.
2. Сохраните текстовый файл с именем `symbol.txt` на диске `D:\`.
3. Введите какую-либо строку в созданном текстовом файле.

Например:



4. Снова сохраните файл после ввода текста.
5. Откройте файл `symbol.txt` в режиме чтения и прочитайте первые 20 символов. Для этого откройте программу Python.
6. В открывшемся окне программы Python введите следующие строки и нажмите клавишу Enter:

```
>>> f = open('D:/symbol.txt', 'r')
>>> f.read(20)
```

7. В результате возвращает строку ниже:

```
'Python was conceived'
```

8. Проверьте ответ. Соответствует ли ваша программа полученному результату? Если да, вы молодцы!

Задание 2. Чтение символов из текстового файла.

Продолжите программу 1-го задания для чтения всех символов файла `symbol.txt`, начиная с 21 символа.

Задание 3. Чтение файла по строке.

Создайте текстовый файл и введите текст стихотворения, состоящий из 4 строк. Напишите программу на языке программирования Python, которая читает первые и четвертые строки стихотворного ряда.

Уровень С

Задание 1. Запись файла.

Предположим, файла `birthday.txt` не существует. Откройте Python и напишите программу, которая создает файл `birthday.txt` с пожеланиями из 2–3 предложений, посвященных дню рождения друзей.

Задание 2. Переименование файла.

Создайте программу для переименования файла `birthday.txt` на `tomyfriend.txt`.

§ 20. Программирование алгоритмов ветвления

Вспомните!

- Как открыть файл на языке программирования Python?
- Как закрыть файл на языке программирования Python?
- Как читать файл на языке программирования Python?
- Как записать файл на языке программирования Python?

Вы узнаете:

- о разветвляющемся алгоритме;
- о структуре условного оператора;
- об использовании условного оператора;
- как записывать разветвляющиеся алгоритмы на языке программирования Python.

Алгоритм – Алгоритм –
Algorithm

Программный код –
Программалық код –
Program code

Условный оператор –
Шартты оператор –
Conditional operator

Логическое выражение –
Логикалық өрнек –
Logical expression

Разветвляющиеся алгоритмы

Последовательность выполнения программы может быть линейной, то есть команды выполняются последовательно от начала до конца друг за другом. В этом случае выполняется вся строка программного кода.

Однако в программе не всегда бывает такая ситуация. В зависимости от каких-либо условий, указанных в целях задачи, некоторые строки кода программы могут быть выполнены, а некоторые не выполнены. Иными словами, в программе может присутствовать ветвление, которое реализуется *условным оператором* – особой конструкцией языка программирования.

Приведем пример из реальной жизни. Ты живешь каждый день по расписанию. Можно сказать, расписание – это алгоритм для человека, его программный код, подлежащий выполнению. Например, по расписанию в 17.00 ты идешь в музыкальную школу по классу домбры. Однако учитель сообщил тебе об отмене занятий. В связи с тем, что сегодня ты не посещаешь музыкальную школу, тебе нужно будет изменить порядок выполнения программы расписания, одним из условий которого является посещение музыкальной школы. В противном случае необходимо выполнение других действий.

Структура if

Подобная нелинейность действий может быть реализована в компьютерной программе. Например, часть кода будет выполняться лишь при определенном значении конкретной переменной. В языке программирования Python используется следующая конструкция условного оператора:

```
if логическое_выражение:  
    выражение 1  
    выражение 2  
    ...
```

В переводе с английского слово «if» означает «если». Объяснить понятным языком можно так: если логическое выражение возвращает истину, то выражение выполняется; если логическое выражение возвращает ложь, то выражение не выполняется.

С помощью блок-схемы легко понять алгоритм выполнения программы. Вы хорошо знаете, что на языке блок-схемы каждая конструкция изображена определенной фигурой. Например, выполнение действия – прямоугольник, логическое выражение – ромб.

Структура условного оператора представлена следующей блок-схемой (рис. 109).

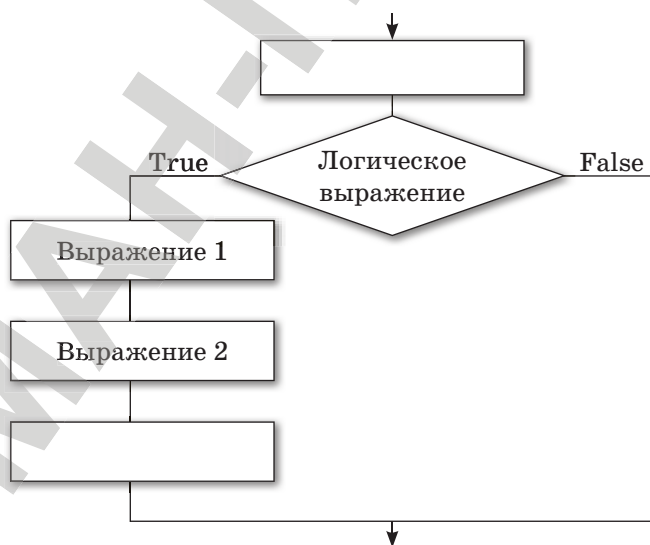


Рис. 109. Блок-схема разветвляющегося алгоритма

Структура `if_логическое выражение` является *заголовком* условного оператора. Строки, на которых расположены выражения, – *телом* условного оператора. Тело может содержать как множество выражений, так и всего одно или даже быть пустым.

Логические выражения формируются с помощью сравнений (табл. 10).

Таблица 10. Сравнения в Python

Сравнения	В Python	Значение
<	<	Меньше
≤	<=	Меньше или равно
>	>	Больше
≥	>=	Больше или равно
=	==	Равно
≠	!=	Не равно

Рассмотрим примеры использования условного оператора в языке программирования Python.

Пример 1. Дано число a . Если заданное число a удовлетворяет условию, то число a следует увеличить на 15.

Тело условного оператора:

```
if a<5:  
    a=a+15
```

```
>>> a=2  
>>> if a<5:  
        a=a+15  
  
>>> print(a)  
17
```

После логического выражения на языке программирования Python нужно поставить двоеточие «:». Тело условного оператора записывается после 4 пробелов или с клавиатуры можно использовать клавишу Tab (табуляция). В конце логического выражения на языке программирования Python после установки двоеточия «:» на новой строке автоматически проставляется 4 пробела. При работе в интерактивном режиме эти пробелы приходится ставить вручную.

В этом примере $a<5$ является логическим выражением. Если логическое выражение возвращает истинное значение, то выполняется выражение $a=a+15$. Если логическое выражение является ложным, то выражение $a=a+15$ не выполняется.

Последняя строка программного кода `print(a)` не имеет отношения к условному оператору. Это можно заметить по отсутствию пробелов перед этим рядом. Если ряд программ не встроен в условный оператор, то условный оператор считается не зависящим от оператора.

Пример 2. Программа должна попросить пользователя ввести две цифры и сравнить их. Если они равны, то необходимо сделать соответствующее сообщение о том, что логическое выражение `a=b` истинно (рис. 110).

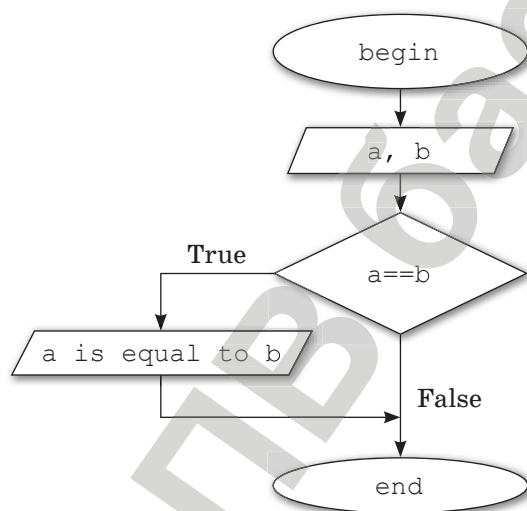


Рис. 110. Блок-схема задачи

Программный код

```
print ('input a')
a=input ()
print ('input b')
b=input ()
if a==b:
    print ('a is equal to b', a,'=',b)
```

Результат программы

```
input a
5
input b
5
a is equal to b 5 = 5
```

1

Отвечаем на вопросы

1. В каких случаях происходит разветвление?
2. Когда применяется условный оператор?
3. Какова структура условного оператора?
4. Как отображается блок выполнения действия?
5. Что вы знаете о блоке логического выражения?
6. Что такое заголовок и тело условного оператора?

2

Думаем и обсуждаем

1. Почему возникает необходимость применения разветвляющихся алгоритмов?
2. В чем заключается важность осуществления разветвления с условным оператором?

3

Анализируем и сравниваем

Сравните между собой линейные и разветвляющиеся алгоритмы в языке программирования Python.

	Линейные алгоритмы	Разветвляющиеся алгоритмы
Определение		
Структура		
Используемые блоки		

4

Выполняем в тетради

Создайте блок-схему, чтобы найти большее из двух чисел. Напишите программу в Python, находящую наибольшее из двух чисел.

Программный код

5

Выполняем на компьютере

1. Проверьте результат работы примера 2, приведенного в параграфе.
2. Составьте программу, определяющую нечетность введенного числа.

3. Дано число C . Составьте программу, которая определит, является ли число C кратным 5.

При решении задач обязательно принять во внимание приведенный шаблон:

Постановка задачи	
Блок-схема	Код программы
Результат программы	

6

Делимся мыслями

Поделитесь мыслями по игре «Цепочка профессий». Последовательность игры ведется по алгоритму ветвления. Первый участник называет профессию, например – повар. Следующий участник называет 2 профессии, которые он ассоциирует с профессией повара, например: металлург и врач. Теперь нужно объяснить, почему именно эти две профессии ассоциируются с данной профессией. Металлург – как повар, тоже имеет дело с огнем, высокими температурами, печами; врач – как повар, строго соблюдает чистоту, носит белые халаты и головные уборы и т.д. Продолжите цепочку профессий, назвав по 2 специальности каждой из двух названных профессий и объяснив причины их выбора.

7

Домашнее задание

Перечислите те действия, которые должны быть выполнены на тему «Мой день». Как вы могли бы использовать время, отведенное для этого, если планируемое действие не произошло?

§ 21. Практикум. Программирование алгоритмов ветвления

Указания для заданий уровня А

Составьте программу для приобретения товаров со скидкой. Если цена товара 2000 тенге или более, то цена товара должна быть рассчитана со скидкой 10%. Составьте блок-схему для решения задачи и заполните в соответствии с образцом оформления.

Шаблон оформления задачи

Постановка задачи	
Если цена товара 2000 тенге или более, то цена товара должна быть рассчитана со скидкой 10%. Нужно составить программу для приобретения товаров со скидкой	
Блок-схема программы	Программный код
<pre> graph TD Start([begin]) --> Input[/t/] Input --> Decision{t >= 2000} Decision -- True --> Process[t = t - t * 10 / 100] Decision -- False --> Output[/t/] Process --> Output Output --> End([end]) </pre>	<pre> print ('cost of the t') t=int (input ()) if t>=2000: t=t-t*10/100 print (t) </pre>
Результат программы	
Если логическое выражение истинно: cost of the t 6000 5400.0	Если логическое выражение ложь: cost of the t 500 500

Здесь t – цена товара. Логическое выражение – $t \geq 2000$, то есть равно 2000 тенге или больше. Если условие истинно, то товар рассчитывается со скидкой, а если условие принимает значение ложь, то цена товара остается неизменной.

Уровень А

Задание. Расстояние от Алматы до Кокшетау составляет 1614 км. Поезд двигался равномерно со скоростью 50 км/ч. За сколько времени поезд доедет до места назначения? Если время превышает 24 часа, то необходимо вывести сообщение «более 1 суток».

Составьте блок-схему и заполните ее в соответствии с образцом оформления.

Уровень В

Составьте блок-схемы и заполните их в соответствии с образцом оформления.

Задание 1. Дано одно число. Составьте программу определения принадлежности заданного числа к интервалу $[1, 10]$.

Задание 2. Даны два числа. Если два числа положительные, то создайте программу, которая найдет среднее арифметическое этих двух чисел.

Задание 3. Даны два числа. Если два числа отрицательные, то создайте программу, которая найдет сумму этих двух чисел, возводя их в квадрат.

Уровень С

Составьте блок-схемы и заполните их в соответствии с образцом оформления.

Задание 1. Дано значение сторон треугольника. Если периметр треугольника нечетный, создайте программу, которая умножит его на 3.

Задание 2. Площадь квадрата известна. Если стороны квадрата четные, создайте программу, которая уменьшает площадь в 2 раза.

Задание 3. Введите два числа с клавиатуры. Если они не равны друг другу, то составьте программу, которая выводит на результат значение «yes».

Задание 4. Введите два числа с клавиатуры. Если они равны друг другу, то составьте программу, увеличивающую два числа по отдельности в 5 раз.

Задание 5: Даны длины сторон двух треугольников: a, b, c и d, e, f . Создайте программу, которая определяет, равны ли периметры двух треугольников.

§ 22. Программирование вложенных условий

Вспомните!

- Что такое разветвляющийся алгоритм?
- Какова структура условного оператора?
- Как использовать условный оператор?
- Как записывать разветвляющиеся алгоритмы на языке программирования Python?

Если – Eger – If

Или – Әйтпесе – Else

Истина – Ақиқат – True

Ложь – Жалған – False

Вы узнаете:

- о структуре `if-else`;
- о структуре `if-elif-else`;
- как использовать вложенные условия на языке программирования Python.

Структура `if-else`

Бывают случаи, когда необходимо рассмотреть другой вариант выполнения программы. Если логическое выражение является истинным, то выполняется одно указание, а если логическое выражение

является ложным, то выполняется другое указание. Вот для этого используется структура `if-else`.

В этом случае выполняется не только одно направление условного оператора, но и оба направления. Таким образом осуществляется полное разветвление.

На языке программирования Python полное разветвление осуществляется с добавлением блока `else`. Полное разветвление выполняется со структурой `if-else` (если – иначе).

Синтаксис оператора `if-else` выглядит следующим образом:

```
логическое_выражение if:
```

```
    выражение 1
```

```
    выражение 2
```

```
    ...
```

```
    выражение n
```

```
else:
```

```
    выражение 1
```

```
    выражение 2
```

```
    ...
```

```
    выражение n
```

В случае возврата логическим выражением `false` поток выполнения программы не возвращается сразу в основную

ветку. На случай false существует другой вложенный код, отличный от случая true. Другими словами, встретившись с расширенной версией условного оператора, поток выполнения программы не вернется в основную ветку, не выполнив хоть какой-нибудь вложенный код. Полная структура условного оператора представляется следующей блок-схемой (рис. 111):

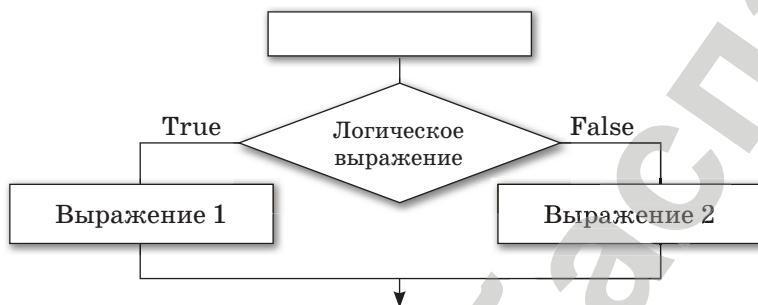


Рис. 111. Блок-схема полной структуры условного оператора

Давайте рассмотрим примеры использования else в языке программирования Python.

Пример 1. У Дидара есть 500 тенге. Он купил некоторые товары в магазине. Необходимо создать программу, которая при оплате выдаст сообщение «Достаточно денег», если стоимость товара меньше 500 тенге, иначе выдаст сообщение «Недостаточно денег» (рис. 112).

Программный код	Результат программы
<pre> print ('input t1') t1=int (input ()) print ('input t2') t2=int (input ()) t=t1+t2 if t<=500: print ('t=', t,'enough money') else: print ('t=', t,'not enough money') </pre>	<p>В случае true:</p> <pre> input t1 200 input t2 250 t=450 enough money </pre> <p>В случае false:</p> <pre> input t1 250 input t2 350 t=600 not enough money </pre>

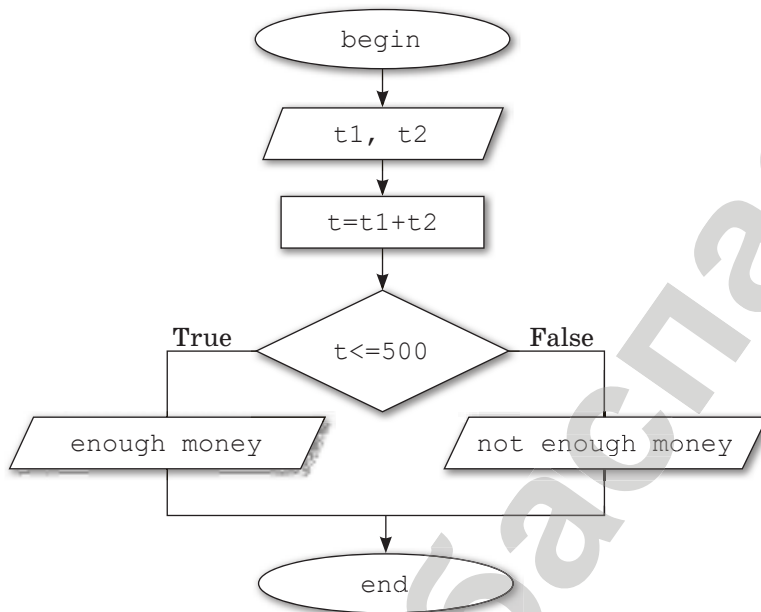


Рис. 112. Блок-схема задачи

В структуру if-else могут входить любые другие операторы. Остановимся на вложенном условном операторе, структура которого выглядит следующим образом:

```

логическое_выражение if:
    логическое_выражение if:
        выражение 1
    else:
        выражение 2
else:
    выражение 3
  
```

Условный оператор, вложенный внутри оператора if, выделен жирным шрифтом. С помощью вложенного условного оператора можно осуществлять не только два, но и несколько выборов. Вложенный условный оператор может быть также расположен после служебного слова else:

```

логическое_выражение if:
    выражение 1
else:
    логическое_выражение if:
  
```

```

        выражение 2
else:
        выражение 3

```

В таком случае после слова `else` нужно будет проверить еще одно условие. При этом вместо оператора `if` можно использовать ключевое слово `elif` (сокращенное от слов `else-if`), осуществляющее каскадное разветвление.

Структура `if-elif-else`

```

if логическое_выражение:
    выражение 1
elif:
    выражение 2
else:
    выражение 3

```

При использовании каскадного разветвления `elif` все ключевые слова `if-elif-else` располагаются на одном уровне.

Пример 2. У Ернурра есть деньги. Если у него имеется более 1000 тенге, то необходимо создать программу, которая выдаст сообщение «много денег», а если денег больше 100 тенге, но меньше 1000 тенге, то выдаст сообщение «достаточно денег», а если меньше 100 тенге, то выдаст сообщение «недостаточно денег» (рис. 113).

Программный код	Результат программы
<pre> print ('input A') A=int (input ()) if A>100: if A>1000: print ('A lot of money') else: print ('enough money') else: print ('not enough money') </pre>	<pre> A>1000 input A 1200 A lot of money 1000 < A > 100 input A 500 enough money A<100 input A 50 not enough money </pre>

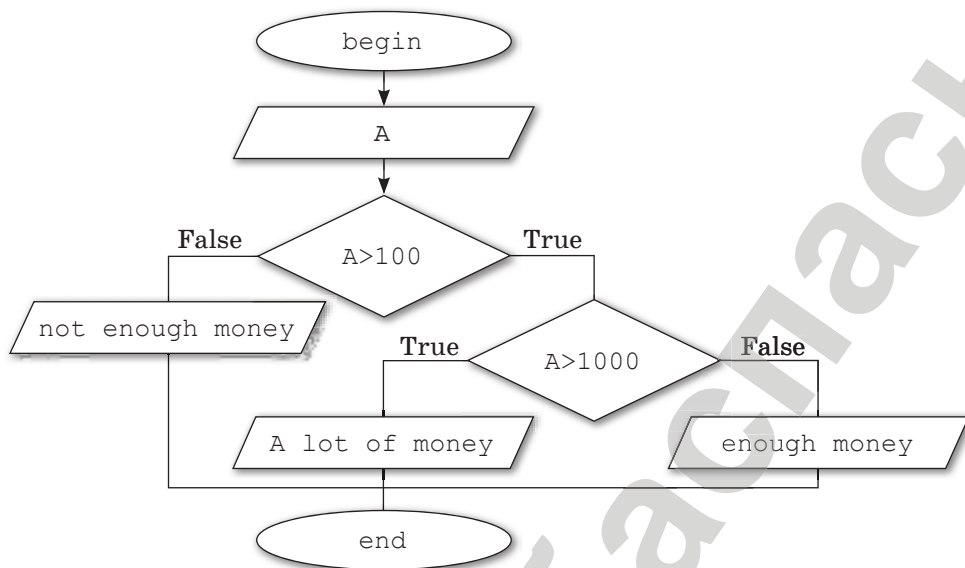


Рис. 113. Блок-схема задачи

1

Отвечаем на вопросы

1. В каких случаях используется структура if-else?
2. Каков синтаксис структуры if-else?
3. В каких случаях применяется структура if-elif-else?
4. Каков синтаксис структуры if-elif-else?
5. Какова функция алгоритма полного разветвления?
6. Какова структура вложенного условного оператора?
7. Каковы условия применения вложенного условного оператора?

2

Думаем и обсуждаем

1. Почему возникает потребность в полной структуре условного оператора?
2. В чем важность осуществления полного разветвления с помощью вложенного условного оператора?

3

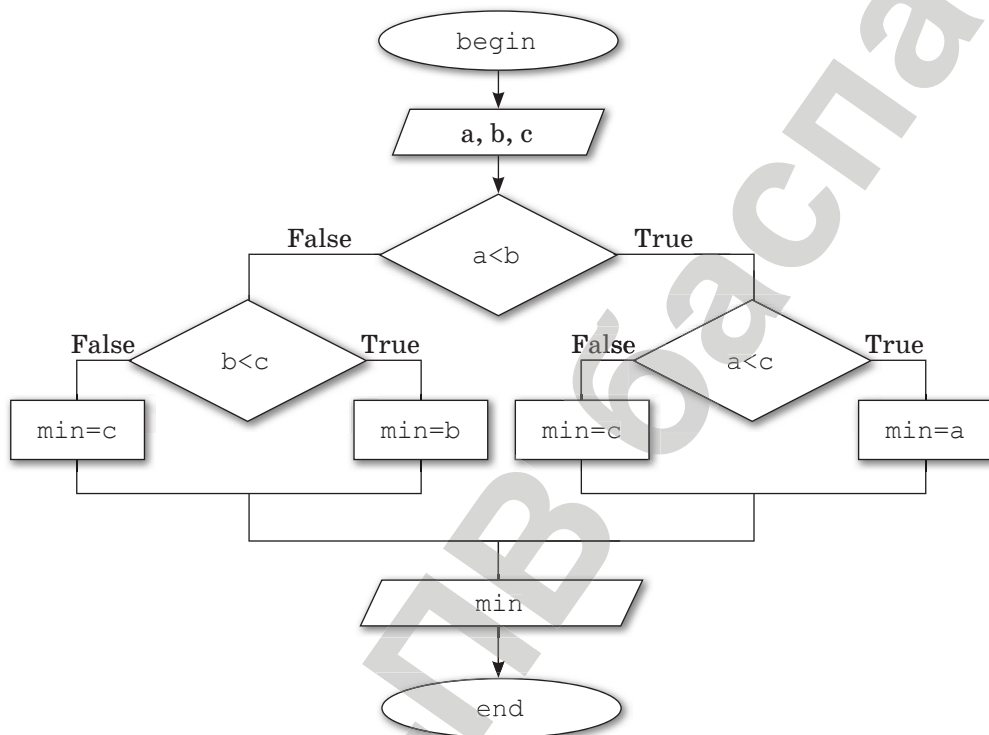
Анализируем и сравниваем

Сравните между собой структуры, осуществляющие вложенные разветвления if-else и if-elif-else.

Структура if-else

Структура if-elif-else

Даны три числа. Нужно найти наименьшее из них и вывести значение на результат. Ниже приведена блок-схема задачи, по которой составьте программу вложенных условий на языке программирования Python.



1. Проверьте в программе Python примеры 1 и 2, приведенные в параграфе.
2. Установите вложенный условный оператор после слова `else` и создайте программу на языке программирования Python.
3. Выполните вложенный условный оператор по структуре `if-elif-else`, создайте программу на языке программирования Python.
4. Дано число. Если заданное число положительное, то число следует увеличить на 5, а если отрицательное, то уменьшить на 10. Если вводимое число равно 0,

то создайте программу, которая выводит сообщение «введите число, отличное от 0».

5. Дано число. Если заданное число нечетное, то число следует увеличить на 5, а если четное, то уменьшить в 2 раза. Если введенное число равно 0, то создайте программу, которая выводит сообщение «введите число, отличное от 0».

При решении задач обязательно примите во внимание приведенный шаблон:

Постановка задачи	
Блок-схема	Программный код
Результат программы	

6

Делимся мыслями

Поделитесь мыслями по теме «Если я ..., тогда буду ..., иначе буду ...». Обязательно нужно выбрать ведущего игры. Условие ставится одним учеником, а второй и третий ученики продолжают предложение. Например, первый ученик говорит: «Если я хорошо изучу математику», далее второй ученик продолжает: «тогда я буду бухгалтером», а третий ученик продолжает: «иначе я буду юристом».

7

Домашнее задание 

Даны три стороны треугольника. Создайте программу, которая определяет, является ли треугольник равносторонним треугольником по значению введенных сторон.

§ 23. Практикум. Программирование вложенных условий

Цель: Решение задач с использованием вложенных условий на языке программирования Python.

Инструкция по выполнению задания:

Сначала составьте блок-схему для задач. В процессе их решения используйте структуры `if ... else`, `if ... elif ... else`.

Задание 1. Если число, введенное с клавиатуры, находится в диапазоне от 1 до 20, то вам нужно создать программу, которая выдает слово «yes», в противном случае слово «no». Программа приведена ниже. Заполните пробелы и проверьте полную программу в Python.

```
n = int(input())
if _____:
    if _____:
        print("YES")
    else:
        print("NO")
else:
    print("NO")
```

Задание 2. Даны значения треугольника и его углов. В зависимости от углов необходимо составить программу, определяющую, является ли треугольник остроугольным, прямоугольным или тупоугольным.

Задание 3. Создайте программу, которая определяет знак числа x . Если $x > 0$, то $\text{sign}x = 1$, если $x < 0$, то $\text{sign}x = -1$, если $x = 0$, то $\text{sign}x = 0$.

Входные данные: вводится одно число.

Выходные данные: выводится знак введенного числа.

Например:

№	Входные данные	Выходные данные
1	15	1

№	Входные данные	Выходные данные
2	-10	-1
3	0	0

Задание 4. Даны три числа. Определите, сколько чисел между ними совпадают друг с другом. Если количество соответствующих чисел три, то следует составить программу, которая выводит значение 3, если два, то следует создать программу, которая выводит значение 2, а если нет соответствующих чисел, то следует создать программу, которая выводит значение 0.

Входные данные: Вводятся три числа.

Выходные данные: Выводится значение в соответствии с условиями задания.

Например:

№	Входные данные	Выходные данные
1	1 2 3	0
2	1 1 1	3
3	1 1 5	2

Задание 5. Вы посетили город X Казахстана. Стоимость билета продается по разным ценам:

- стоимость одной поездки 40 тенге;
- стоимость 20 поездок составляет 600 тенге;
- стоимость 60 поездок составляет 1500 тенге.

Вы должны путешествовать n раз по вашему плану. Сколько билетов нужно купить по разным ценам, чтобы потратить немного денег? Общее количество билетов должно быть не менее n .

Входные данные: вводится число n .

Выходные данные: программа должна вывести 3 необходимых числа проездных билетов за 1, 20, 60 раз.

§ 24. Программирование составных условий

Вспомните!

- Каков синтаксис структуры `if-else`?
- Что из себя представляет структура `if-elif-else`?
- Как использовать вложенные условия на языке Python?

Вы узнаете:

- о составных условиях;
- о логическом сложении;
- о логическом умножении;
- о логическом отрицании;
- о логических операциях;
- как использовать составные условия в Python.

Составные условия – *Күрделі шарттар* – Compound conditions

Логическое сложение – *Логикалық қосу* – Logical addition

Логическое умножение – *Логикалық көбейту* – Logical multiplication

Логическое отрицание – *Логикалық терістеу* – Logical negation

Логические операции – *Логикалық амалдар* – Logical operations

Составные условия

Простейшие условия состоят из одного отношения, например, больше, меньше, больше или равно, меньше или равно, равно и так далее. Но иногда необходимо объединение простых условий в более сложные, например: на улице холодно, и идет снег. Здесь два простых условия (на улице холодно), (на улице идет снег) связаны связкой **И**.

Составные условия состоят из двух или нескольких простых отношений (условий), которые объединяются с помощью логических операций. Логические операции, используемые при написании составных условий на языке программирования: **И** – логическое умножение – на языке Python записывается, как **and**; **ИЛИ** – логическое сложение – на языке Python записывается, как **or**; **НЕ** – логическое отрицание – на языке Python записывается, как **not**.

Логическая операция **И** – требует одновременного выполнения двух условий.

Условие 1 and условие 2 – будет принимать истинное значение, только если оба простых условия истинны одновременно, причем, если условие 1 ложно, то условие 2 проверяться не будет.

Логическая операция **ИЛИ** – требует выполнения хотя бы одного из условий.

Условие 1 or условие 2 – будет принимать ложное значение, только если оба простых условия ложны одновременно, причем, если условие 1 истинно, то условие 2 проверяться не будет.

Логическая операция **НЕ** (отрицание) меняет значение высказывания на противоположное: истинно на ложно, а ложно на истинно.

Not условие 1 – будет принимать ложное значение, если условие 1 истинно и наоборот.

Например, следующие два условия равносильны: **A < B** и **not (A >= B)**.

Порядок выполнения логических операций и отношений:

- 1) операции в скобках
- 2) операция НЕ
- 3) логические отношения >, <, >=, <=, ==, !=
- 4) операция И
- 5) операция ИЛИ

Для изменения порядка действий используются круглые скобки.

Логические переменные

Во многих языках программирования есть возможность использовать переменные, которые хранят логические значения (истина или ложь). В языке программирования Python такие переменные могут принимать значения True (истина) или False (ложь).

Пример 1. Пусть *x* будет равным значению *истина*, а *y* – *ложь*. Создайте программу, которая проверит логические действия И, ИЛИ, НЕ.

Программный код	Результат программы	Программный код	Результат программы
<pre>x=true y=false print (x and y)</pre>	false	<pre>x=true y=false print (not (x))</pre>	false
<pre>x=true y=false print (x or y)</pre>	true	<pre>x=true y=false print (not (y))</pre>	true

Пример 2. Число вводится с клавиатуры. Создайте программу, которая проверяет, что введенное число больше или равно 10 и меньше или равно 100.

Программный код	Результат программы		
	n<10	10<=n <=100	n>100
<pre>n=int (input ()) if n>=10 and n<=100: print ("YES") else: print ("NO")</pre>	5 NO	25 YES	120 NO

1

Отвечаем на вопросы

1. Что такое составные условия?
2. Какие логические операции применяются при создании составных условий?
3. Что вы знаете о логической операции И?
4. В чем особенность логической операции ИЛИ?
5. Что вы знаете о логической операции НЕ?
6. Каков порядок выполнения логических операций и отношений?
7. Что такое логические переменные?

2

Думаем и обсуждаем

1. Почему возникает необходимость усложнения простых условий?
2. В чем важность соблюдения последовательности выполнения логических операций и отношений?

3

Анализируем и сравниваем

Сравните между собой логические операции, используемые при написании составных условий на языке программирования.

Логическая операция И	Логическая операция ИЛИ	Логическая операция НЕ

4

Выполняем в тетради

Напишите результат выполнения логических действий, приведенных ниже:

- $(12 > 25) \text{ and } (30 > 25) - \dots$
 $(27 < 50) \text{ and } (15 > 7) - \dots$
 $(36 < 75) \text{ and } (5 < 43) - \dots$
 $(30 > 20) \text{ or } (30 < 10) - \dots$
 $(30 > 25) \text{ or } (25 > 10) - \dots$
 $(10 > 20) \text{ or } (10 < 50) - \dots$
 $\text{not } (50 > 16) - \dots$
 $\text{not } (24 < 12) - \dots$

1. Проверьте примеры 1 и 2, приведенные в параграфе, используя программу Python.
2. Даны три числа. Составьте программу, которая проверит, что все эти три числа являются положительными. При построении программы используйте логические операции.
3. Дано число. Составьте программу, определяющую, что введенное число является двухзначным числом. Например, 42 – двухзначное число.
4. Даны числа x , y . Если x и y имеют отрицательное значение, то значение каждого необходимо заменить их модулями. А если одно из значений x и y отрицательные, то оба значения должны быть увеличены на 5. Если значения x и y положительны, то составьте программу, которая увеличивает их значения в 10 раз.
5. Даны три числа. Составьте программу, которая находит минимальное и максимальное значения этих трех чисел.

При решении задач обязательно принять во внимание приведенный шаблон:

Постановка задачи	
Блок-схема	Программный код
Результат программы	

Поделитесь своими мыслями, написав эссе на тему «Я в будущем хочу/не хочу быть ...». Для ответа на вопросы *почему «да»? почему «нет»? приведите не менее 3 причин по профессии, которую вы выбрали или не выбрали.*

Создайте программу, которая проверяет, является ли хотя бы одно из трех чисел, введенных с клавиатуры, нечетным. Например, для чисел 10, 15, 20 – «да», а для чисел 10, 20, 30 – «нет».

§ 25. Практикум. Программирование составных условий

Цель работы: Решение задач с использованием составных условий на языке программирования Python.

Уровень А

Задание. Напишите программу на языке программирования Python по приведенным блок-схемам.

№	Блок-схемы	Программный код
1.	<pre>graph TD; A[x and y] --> B{x=true}; B -- False --> C[False Output]; B -- True --> D{y=true}; D -- False --> E[False Output]; D -- True --> F[True Output];</pre>	
2.	<pre>graph TD; A[x or y] --> B{x=true}; B -- True --> C[True Output]; B -- False --> D{y=true}; D -- False --> E[False Output]; D -- True --> F[True Output];</pre>	
3.	<pre>graph TD; A[Not x] --> B{x=true}; B -- True --> C[False Output]; B -- False --> D[True Output];</pre>	

Уровень В

Задание 1. Реализуйте проверку попадания переменной x в диапазон $[1;100]$, то есть условие должно быть истинным, когда x принимает значения в этом диапазоне чисел. Очевидно, что здесь должно быть две проверки: первая – мы проверяем, что $x \geq 1$ и вторая – проверяем, что $x \leq 100$. Если оба этих условия выполняются одновременно, то x попадает в диапазон. Завершите следующий программный код операциями сравнения и логическими операциями. Проверьте результат.

```
x=50
if x ? 1 _____ x ? 100:
    print("x попадает в диапазон [1; 100]")
else:
    print("x не попадает в диапазон [1; 100]")
```

Задание 2. Реализуйте противоположное условие, что x не принадлежит диапазону $[1; 100]$. Завершите следующий программный код операциями сравнения и логическими операциями. Проверьте результат.

```
x=50
if x ? 1 _____ x ? 100:
    print("x не попадает в диапазон [1; 100]")
else:
    print("x попадает в диапазон [1; 100]")
```

Уровень С

Задание 1. Запишите условие, которое является истинным, когда: 1) каждое из чисел X и Y нечетное; 2) только одно из чисел X и Y меньше 10; 3) хотя бы одно из чисел X и Y равно 0.

Задание 2. Запишите условие, которое является истинным, когда: 1) целое A кратно 2 или 3; 2) целое A не кратно 3.

Задание 3. Запишите условие, которое является истинным, когда: 1) целое N кратно 5 или 7; 2) целое N не кратно 4 и не оканчивается нулем.

§ 26–27. Практикум. Организуем выбор

Цель работы: записать разветвляющиеся алгоритмы на языке программирования Python.

Инструкция по выполнению задания:

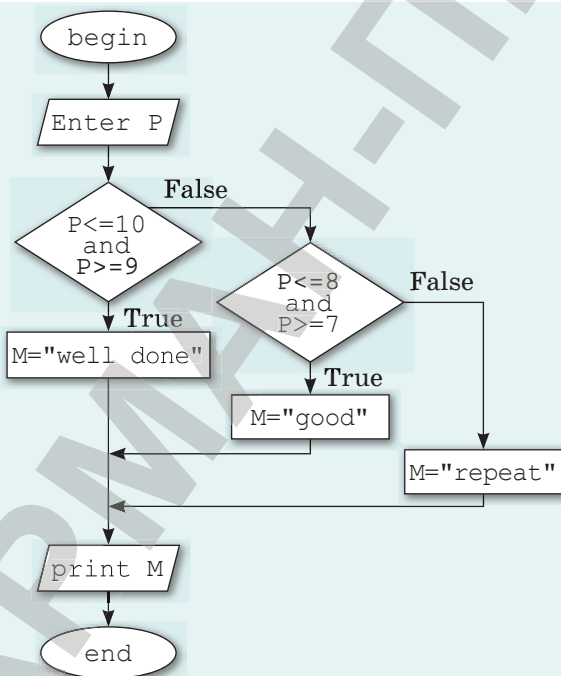
1. При решении задач используйте шаблон оформления.
2. Сначала составьте блок-схему для решения задачи.
3. Составьте программу на языке программирования Python в соответствии с блок-схемой.
4. Проверьте все условия, представленные в описании задач, и поместите результат в шаблон.

Задание 1. Вводя свои оценки по предмету «Информатика», вам необходимо запрограммировать устный текст, который вы получите от учителя, в соответствии с баллами.

Постановка задачи

Создайте программу, в которой учитель скажет «Молодец», если вы каждый урок получаете 9–10 баллов по предмету «Информатика», если же только 7–8 баллов, то скажет «Хорошо», а если меньше 4–6 баллов, то скажет «Подготовьтесь еще».

Блок-схема



Программный код

```
P=int (input ())
if P<=10 and P>=9:
    M="well done"
elif P<=8 and P>=7:
    M="good"
else:
    M="repeat"
print (M)
```

Результат программы

```
10
well done

8
good

6
repeat
```

Задание 2. Дан номер места в плацкартном вагоне. Определите, какое это место: верхнее или нижнее, спальное или боковое.

Задание 3. Создайте программу для решения следующих уравнений.

- 1) Дано целое число x . Вычислите значение y . Если $x > 2$, то $y = \frac{1}{x^2 + 4x + 5}$, в остальных случаях $y = x^2 + 4x + 5$.
- 2) Дано целое число a . Вычислите значение y . Если $a > 1$, то $y = a^2 - 5$, в остальных случаях $y = a^4$.
- 3) Дано целое число z . Вычислите значение y . Если $z > 5$, то $y = z^2 - 15$, в остальных случаях $y = z + 15$.
- 4) Дано целое число b . Вычислите значение y . Если $b > 10$, то $y = b + \frac{b}{3}$, в остальных случаях $y = b^2 - 4b$.

Задание 4. Создайте алгоритм ветвления на языке программирования Python и напишите программу для следующих задач.

- 1) Дано натуральное число. Определите, будет ли это число четным и (или) кратным 2.
- 2) Дано натуральное число. Определите, будет ли это число нечетным и (или) кратным 5.
- 3) Дано натуральное число. Определите, будет ли это число положительным и (или) кратным 7.
- 4) Дано натуральное число. Определите, будет ли это число отрицательным и (или) кратным 4.

Задание 5. Создайте алгоритм ветвления на языке программирования Python и напишите программу для следующих задач.

1) На каждом этаже девятиэтажного дома по 5 квартир. Квартиры расположены в порядке возрастания, нумерация начинается слева. На каждом этаже справа – 1 квартира, по середине – 3 квартиры, слева – 1 квартира. Создайте программу, которая определяет, какой это этаж по номеру квартиры. Номер квартиры ввести с клавиатуры.

А) Сколько квартир в девятиэтажном доме?

Б) На каком этаже и где находится квартира (справа, слева, по центру)?

2) Мама отправила Алинура в магазин купить: 2 литра молока, 10 яиц и 3 лепешки. Стоимость 1 литра молока – 320 тенге, 1 яйца – 36 тенге, 1 лепешки – 100 тенге. Деньги, которые мама дала Алинуру, введите с клавиатуры.

А) Достаточно ли у Алинура денег, чтобы принести из магазина все, что заказала мама?

Б) Если недостаточно денег, как мы можем убедиться, что у Алинура хватит денег, чтобы купить по одному из этих продуктов?

ЗАДАНИЯ ПО СУММАТИВНОМУ ОЦЕНИВАНИЮ К РАЗДЕЛУ 4

1. Набор данных, сохраненный в виде последовательности битов на компьютере.
 - A. Файл.
 - B. Папка.
 - C. Документ.
 - D. Программа.
 - E. Каталог.
2. Типы файлов на языке программирования Python.
 - A. Текстовые.
 - B. Бинарные.
 - C. Составные.
 - D. Сложные.
 - E. Вложенные.

3. Установите соответствие.

1.	r	A)	Для чтения и записи
2.	r+	B)	Для чтения и записи (бинарный)
3.	a+	C)	Только для записи
4.	wb+	D)	Только для чтения
5.	w	E)	Откроет для добавления нового содержимого

4. Заполните пропуски:

В блок-схеме выполнение действия – ..., логическое выражение образуется фигурой

5. Заполните пропуски:

Структура `if` логическое_выражение является ... условного оператора. Строки, на которых расположены выражения, – ... условного оператора.

6. Если логическое выражение является истинным, то выполняется одно указание, а если логическое выражение является ложным, то выполняется другое указание. Какая это структура условного оператора?

- A. Структура if – else.
- B. Структура if.
- C. Структура if-elif-else.

7. Данные условия состоят из двух или нескольких простых отношений, которые объединяются с помощью логических операций.

- A. Составные условия.
- B. Логическое сложение.
- C. Логическое умножение.
- D. Логическое отрицание.
- E. True, False.

8. Установите соответствие:

1.	И	A)	Логическое сложение
2.	ИЛИ	B)	Логическое отрицание
3.	НЕ	C)	Логическое умножение

9. Логическая операция, которая требует одновременного выполнения двух условий:

- A. И
- B. ИЛИ
- C. НЕ

10. Логическая операция, которая требует выполнения хотя бы одного из условий:

- A. ИЛИ
- B. И
- C. НЕ

ИТОГИ 4 РАЗДЕЛА

В разделе «Программирование алгоритмов на языке Python» вы познакомились с понятиями файл, чтение файла, запись файла, текстовый файл, двоичный файл, открытие файла, закрытие файла, разветвленные алгоритмы, условный оператор, логическое выражение, условие, полное ветвление, неполное ветвление, вложенные условия, сложные условия, логические операции, логическое сложение, логическое умножение, логическое отрицание, логические переменные.

В этом разделе вы рассмотрели функции чтения и записи файлов, переименования файлов, методы работы с файлом в Python, структуру if разветвленных алгоритмов, блок-схему алгоритма ветвления, структуру if-else, структуру if – elif – else, встроенную структуру условного оператора, логические операции, используемые при написании сложных условий, порядок выполнения логических операций и отношений.

Были представлены практикумы по работе с файлами, программированию разветвленных алгоритмов, программированию вложенных условий, решению практических задач по организации выбора и программирования сложных условий, чтению и написанию файлов на языке программирования Python, написанию разветвленных алгоритмов на языке программирования Python, применению вложенных условий на языке программирования Python.

Раздел 5

ПРАКТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Изучив раздел, вы научитесь:

- записывать разветвляющиеся алгоритмы на языке программирования Python (пайтон).

§ 28. Постановка задачи

Вспомните!

- Что такое файл?
- Какова функция файловой системы?
- Что вы знаете о программировании вложенных условий?
- Что вы знаете о программировании составных условий?

Вы узнаете:

- об этапах решения задач;
- о постановке задачи;
- о составе постановки задачи.

Задача – Мәселе – Problem

Этап – Кезең – Stages

Цель – Мақсат – Goal

Результат – Нәтиже – Result

Постановка задачи

Процесс решения задач на компьютере – это совместная деятельность человека и вычислительной машины. Этот процесс можно представить в виде нескольких последовательных этапов (схема 10). На долю человека приходятся этапы, связанные с творческой деятельностью – постановкой задачи, алгоритмизацией, программированием задач и анализом результатов, а на долю компьютера – этапы обработки информации в соответствии с разработанным алгоритмом.

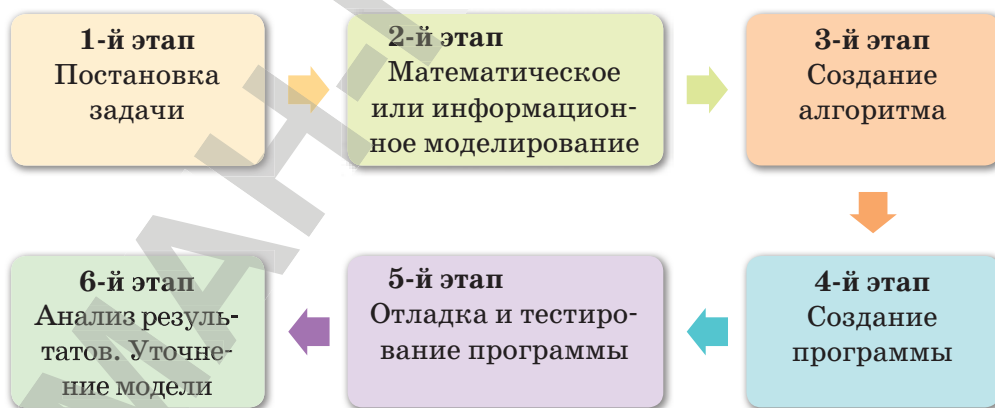


Схема 10. Этапы решения задач с помощью компьютера

Создание любой программы начинается с *постановки задачи*. На этом этапе работу выполняет человек, хорошо представляющий предметную область задачи. Он должен четко определить

цель задачи, дать словесное описание содержания задачи и предложить подход к ее решению. Иначе говоря, на этом этапе формируется условие задачи (схема 11). Этот этап очень важен, так как от правильного понимания целей и требований зависит в основном успех реализации создаваемой программы.

Постановка задачи – точное и четкое определение требуемых результатов и исходных условий в задачах.

Результаты правильные – если они отвечают требованиям поставленных задач.

Результаты неправильные – если они противоречат поставленным требованиям.

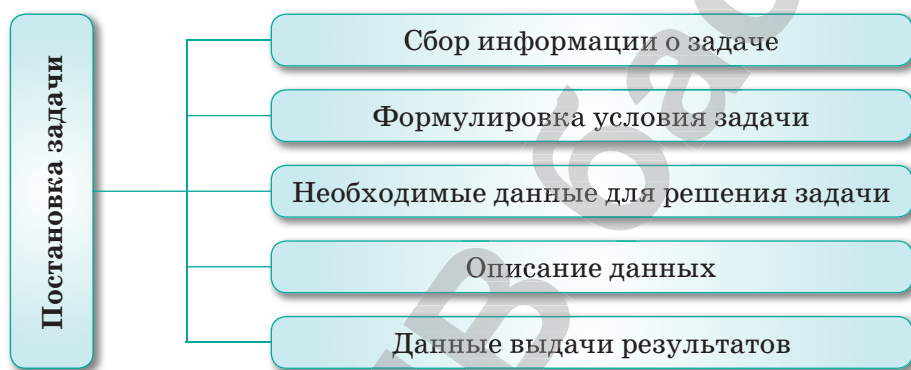


Схема 11. Постановка задачи

Данные в алгоритме ветвления влияют на порядок выполнения шагов алгоритма. Алгоритм предлагает разные пути, в зависимости от действия выбирается один из возможных вариантов. В зависимости от результатов условной проверки выполняется только «да» или «нет». Здесь происходит полное ветвление. В некоторых случаях, в зависимости от результатов проверки, действие «да» выполняется или не выполняется. В этом случае происходит неполное ветвление.

Рассмотрим несколько задач, результаты которых получены с использованием алгоритма ветвления.

Задание 1. Даны два числа. Найдите наибольшее из двух указанных чисел.

Постановка задачи: исходными данными являются переменные a и b . Их значения будут задаваться вводом. Значение большего из чисел должно быть присвоено переменной c

и выведено на результат. Например, если $a = 5$, $b = 10$, то должно получиться: $c = 10$.

Задание 2. Упорядочите значения двух переменных a и b по возрастанию.

Постановка задачи: если для исходных значений переменных справедливо отношение $a \leq b$ (например, $a = 10$, $b = 20$), то оставить их значения без изменения; если же $a > b$ (например, $a = 20$, $b = 10$), то выполнить обмен значениями. Для обмена значений необходимо использование третьей, вспомогательной переменной c . В алгоритме решения данной задачи используется неполное ветвление.

1

Отвечаем на вопросы

1. Из скольких этапов состоит процесс решения проблем на компьютере?
2. Какие действия человека при решении задач на компьютере вы знаете?
3. Какие действия, связанные с компьютером при решении задач на компьютере, вы можете назвать?
4. Что такое постановка задачи?
5. Что относится к составу постановки задачи?

2

Думаем и обсуждаем

1. В чем важность четкого определения заданных условий и требуемого результата в задаче?
2. Почему важно соблюдение этапов решения задач на компьютере?

3

Анализируем и сравниваем

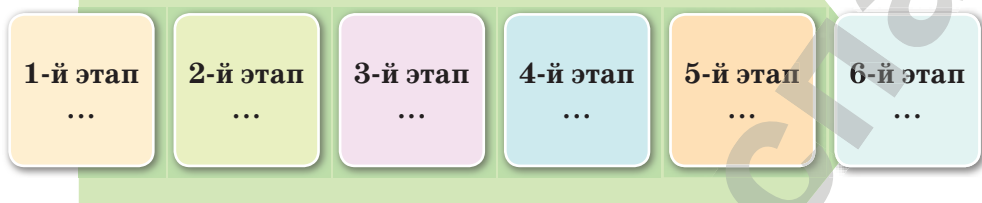
Проанализируйте и сравните каждую из задач решения в отдельности.

Сбор информации о задаче		Необходимые данные для решения задачи
	Формулировка условия задачи	
Данные выдачи результатов		Описание данных

4

Выполняем в тетради

Напишите в схеме этапы решения задач с помощью компьютера:



5

Выполняем на компьютере

В текстовом редакторе создайте таблицу и заполните этапы постановки задачи. Каждую задачу сохраните как отдельный файл.

Постановка задачи	
О задаче (сбор информации о задаче)	
Условия задачи (формулировка условия задачи)	
Входные данные (необходимые данные для решения задачи)	
Описание данных	
Выходные данные (данные выдачи результатов)	

1. Администрация школы объявила конкурс по отбору лучших учащихся среди 7–9 классов. По условиям конкурса, успеваемость учащихся по десятибалльной системе оценки знаний не должна быть меньше 1–5 баллов. Количество баллов 6–7 должно быть не более двух.

- Необходимо создать программу, которая проверит, относится ли ученик к числу лучших по количеству баллов.
2. Дан треугольник. Длины сторон АВ, АС и ВС известны. В зависимости от сторон треугольника определите, является ли он равнобедренным, равносторонним или разносторонним.
 3. Проведите опрос среди одноклассников по теме «Мой любимый город». Добавьте в анкету не менее 10 городов. По результатам этого опроса необходимо создать программу, которая определит города, входящие в первую тройку. Подготовьте интересную информацию о самом любимом городе.
 4. Составьте программу, определяющую средний вес и рост учащихся в классе. Определите наибольший вес и наименьшую длину роста. Подготовьте дополнительную информацию по результатам. Учитывайте нормальный вес и рост в возрасте 13–14 лет.
 5. В городе, где проживает X жителей, 40% жителей интересуются боксом. Если число, введенное с клавиатуры, составляет до 50% поклонников бокса, создайте программу, которая выводит сообщения «смотрит 1 раз в месяц», если до 70% – «смотрит бокс только в выходные дни», а если до 99% – «смотрит бокс ежедневно».

6

Делимся мыслями

Каковы этапы решения задач по математике, физике и химии? Как проводится формирование проблем, предоставление методов решения по данным предметам?

7

Домашнее задание 

Идет подбор персонала на крупное предприятие. Главным условием приема на работу является наличие стажа работника не менее 10 лет и возраст не менее 35 лет. Составьте условия заданной задачи. В чем состоит цель? Каков ожидаемый результат? Стаж и возраст работника вводится с клавиатуры.

§ 29. Разработка алгоритма

Вспомните!

- Какие этапы решения задач вы знаете?
- Что такое постановка задачи?
- Что такое состав постановки задачи?

Вы узнаете:

- об алгоритме;
- о свойствах алгоритма;
- о способах разработки алгоритма.

Алгоритм – Алгоритм – Algorithms

Дискретность – Дискреттілік – Discreteness

Определенность – Анықтылық – Definiteness

Результативность – Нәтижелік – Effectiveness

Массовость – Жалпылық – Generality

Понятность – Түсініктілік – Certainty

Что такое алгоритм?

Алгоритм – полное и точное предписание исполнителю выполнить определенную последовательность действий, направленных на решение поставленной задачи или достижение указанной цели.

Исполнители алгоритма

Выполнение алгоритма может быть поручено человеку, вычислительной машине (компьютеру) или автоматическому устройству. Разработка алгоритма – процесс творческий.

При создании алгоритма необходимо следить, чтобы он удовлетворял свойствам алгоритмов, позволяющих отличать их от других инструкций (*схема 12*).

Дискретность

Алгоритм должен быть разбит на конечное число шагов (действий)

Определенность

Каждый шаг алгоритма должен быть четким, однозначным и задавать выполнение определенного действия

Результативность

Алгоритм за конечное число шагов должен приводить исполнителя к решению задачи

Массовость

Алгоритм решения задачи разрабатывается в общем виде

Понятность

Каждый шаг алгоритма должен быть понятен исполнителю

Схема 12. Свойства алгоритма

Существует несколько способов записи алгоритма (схема 13).

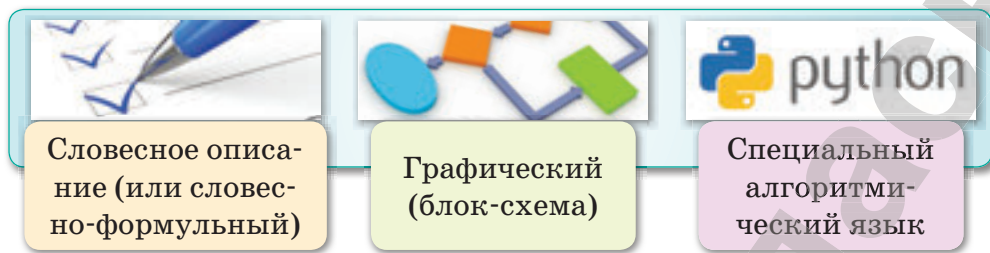


Схема 13. Способы записи алгоритма

Перед составлением программ чаще всего используются словесно-формульный или графический способы.

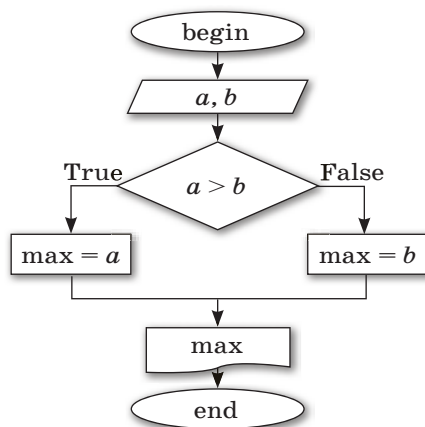
Графический способ записи алгоритма в виде блок-схемы имеет более компактный и наглядный вид. Алгоритм изображается геометрическими фигурами (блоками), соединенными линиями со стрелками. В блоках записывается последовательность действий. Линии определяют порядок их выполнения.

В пределах одной схемы рекомендуется изображать блоки одинаковых размеров. С информацией об основных видах блоков и их функциях вы знакомились в 6 классе.

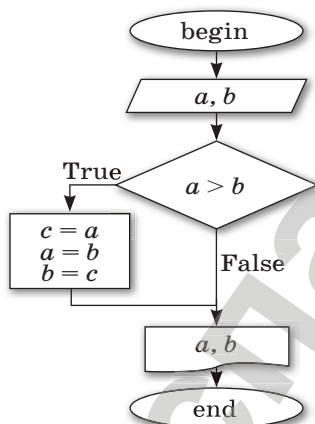
В блок может входить несколько линий, то есть блок может являться приемником любого числа блоков. Из блока (кроме логического) может выходить только одна линия. Схему алгоритма следует выполнять как единое целое, однако в случае необходимости допускается обрывать линии, соединяющие блоки.

Для задач, предусмотренных в предыдущем параграфе, используем графический (блок-схема) метод при разработке алгоритма.

Задача 1. Даны два числа. Найдите наибольшее из двух указанных чисел.



Задача 2. Упорядочите значения двух переменных a и b по возрастанию.



1

Отвечаем на вопросы

1. Что такое алгоритм?
2. Что относится к исполнителям алгоритма?
3. Какие свойства алгоритма вы знаете?
4. Сколько способов записи алгоритма вы знаете?
5. Как представляется графический метод разработки алгоритма?
6. Что такое основные геометрические блоки? Каковы их назначения?

2

Думаем и обсуждаем

1. Почему разработка алгоритма считается творческой работой?
2. В чем заключается важность сохранения его свойств при разработке алгоритма?

3

Анализируем и сравниваем

Проанализируйте и сравните особенности применения одного из способов разработки алгоритма решения физических, математических и химических задач.

Виды задач	Разработка алгоритма решения задач (словесно-формульный, графический, специальный алгоритмический языки)
Физические задачи	
Математические задачи	
Химические задачи	

4

Выполняем в тетради

Сопоставьте свойства алгоритма и их характеристики:

1	Дискретность	А	Алгоритм решения задачи разрабатывается в общем виде
2	Массовость	В	Алгоритм должен быть разбит на конечное число шагов
3	Определенность	С	Алгоритм за конечное число шагов должен приводить исполнителя к решению задачи
4	Понятность	Д	Каждый шаг алгоритма должен быть четким
5	Результативность	Е	Каждый шаг алгоритма должен быть понятен исполнителю

5

Выполняем на компьютере

Продолжим работу над 5 задачами, которые приводятся в параграфе 28 в разделе «Выполняем на компьютере». Откройте файлы постановки задач, написанные для каждой задачи ниже. Создайте блок-схемы по графическому методу разработки алгоритма для каждой задачи с использованием текстового редактора или другого любого графического редактора и сохраните файл. Это нужно будет для того, чтобы создавать программу по разработанным алгоритмам в следующем параграфе.

Задачи	Блок-схемы
1.	

6

Делимся мыслями

Составьте алгоритм «Как построить скворечник для птиц?» и поделитесь мыслями со своими одноклассниками.

Для начала необходимо выбрать материал, из которого будет сделан скворечник (картон, пластик, дерево). Определите преимущества и недостатки каждого материала. При

создании скворечника учитывайте разные явления природы (снег, дождь, ветер, жара). Какие свои секреты бывают при выборе цвета скворечника?

В блок-схеме должны быть отражены различные варианты, такие как

Если скворечник сделан из картона,

то ...

иначе ...

если скворечник сделан из дерева,

то ...

иначе ...

если скворечник сделан из пластика,

то ...

Рассмотрите в блок-схеме различные ситуации, касающиеся изменений природы и цвета скворечника.

7

Домашнее задание

1. В кухне и гостиной необходимо сделать ремонт и заменить обои. Высота и ширина комнат вводятся с клавиатуры. Вычислите площадь двух комнат и определите, для какой комнаты вам нужно больше обоев.
2. При покупке билета на канатную дорогу Медеу – Шымбулак предусмотрены особые льготы для некоторой категории людей. Например, для детей до 5 лет проезд бесплатный как в будние, так и в выходные дни. Пенсионеры могут пользоваться подъемником бесплатно только в будние дни. Для отдыхающих в возрасте от 11 до 23 лет предусмотрена скидка 50%, а старше 24 лет тариф оплачивается полностью.
Создайте программу, которая рассчитывает стоимость канатной дороги Медеу – Шымбулак с учетом скидок путем ввода номера рабочего дня на клавиатуре (например, понедельник 1, вторник 2, ..., воскресенье 7) и возраст отдыхающего.

§ 30. Программирование алгоритма

Вспомните!

- Что такое алгоритм?
- Какие свойства алгоритма вы знаете?
- Назовите способы записи алгоритма.

Вы узнаете:

- о программе;
- о языках программирования;
- о программировании алгоритма.

Программа – *Программа* – Program

Языки программирования – *Программалау тілдері* – Program language

Команда – *Команда* – Command

Оператор – *Оператор* – Operator

Что такое программа?

В основу работы компьютеров положен программный принцип управления, состоящий в том, что компьютер выполняет действия по заранее заданной программе. Этот принцип обеспечивает универсальность использования компьютера: в определенный момент времени решается задача в соответствии выбранной программой. После ее завершения в память загружается другая программа и т.д.

Программа – это запись алгоритма решения задачи в виде последовательности команд или операторов языком, который понимает компьютер.

Исполнитель программы – компьютер или другое автоматическое устройство.

На этапе программирования алгоритма алгоритм решения задачи переводится на язык программирования. Для программирования обычно используются языки высокого уровня, поэтому составленная программа требует перевода ее на машинный язык. После такого перевода получается уже соответствующая машинная программа.

Язык программирования – формальная знаковая система, предназначенная для записи программ.

Ряд действий, выполняемых в программировании алгоритма (схема 14):

- 1 Выбор языка программирования
- 2 Уточнение методов организации данных
- 3 Запись алгоритма на выбранном языке программирования

Схема 14. Действия в программировании алгоритма

Язык программирования Python относится к числу языков программирования высокого уровня.

Рассмотрим примеры программирования разветвленных алгоритмов на языке программирования Python. Для этого мы будем программировать разработанные алгоритмы задач, приведенные в предыдущем параграфе.

Задача 1. Даны два числа. Найдите наибольшее из двух указанных чисел.

Программный код	Результат программы
<pre>a=int(input()) b=int(input()) if a>b: max=a else: max=b print('max=', max)</pre>	Если $a > b$: 10 4 max = 10 иначе, если $a \leq b$: 4 10 max = 10

Нетрудно понять смысл этого алгоритма. Если значение переменной a больше, чем b , то переменной max присвоится значение a . В противном случае, когда $a \leq b$, переменной max присвоится значение b . Условием, по которому разветвляется

алгоритм, является отношение $a > b$. Такое отношение является **логическим выражением**. Если оно справедливо, то результатом будет логическая величина «истина» и выполнение алгоритма продолжится по ветви «да»; в противном случае логическое выражение примет значение «ложь» и выполнение алгоритма пойдет по ветви «нет».

Задача 2. Упорядочите значения двух переменных a и b по возрастанию.

Программный код	Результат программы
<pre>a=int(input()) b=int(input()) if a>b: c=a a=b b=c print(a,b)</pre>	<p>Если $a > b$:</p> <p>6 4 4 6</p> <p>Иначе значения останутся без изменения:</p> <p>4 6 4 6</p>

Для обмена значений необходимо использование третьей, вспомогательной переменной c . Кроме того, как вы сами видите в алгоритме решения данной задачи используется неполное ветвление.

1

Отвечаем на вопросы

1. Что такое программа?
2. Что относится к исполнителям программы?
3. Что такое язык программирования?
4. Какие действия, выполняемые в программировании алгоритма, вы знаете?

2

Думаем и обсуждаем

1. В чем заключается необходимость перевода алгоритма решения задачи на язык программирования?
2. В чем заключается важность действий, выполняемых в программировании алгоритма?

3

Анализируем и сравниваем

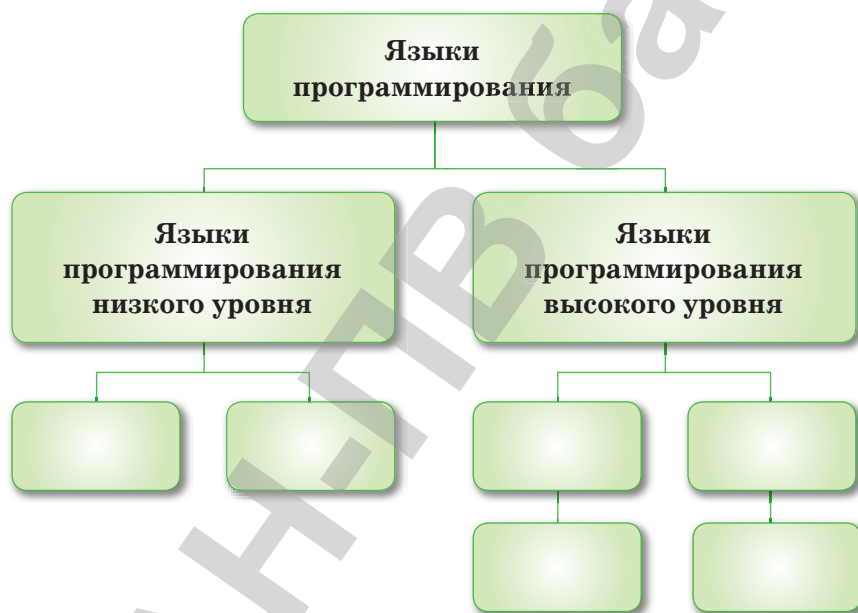
Используя дополнительные материалы и источники информации, сравните между собой языки программирования, приведенные ниже:

Языки программирования		
Pascal	Lazarus	Python

4

Выполняем в тетради

Используя дополнительные материалы и источники информации, заполните схему и запишите в тетрадь.



5

Выполняем на компьютере

Продолжим работу над 5 задачами, которые приводятся в параграфе 28 в разделе «Выполняем на компьютере».

Откройте файлы постановки задач, разработанные алгоритмы в виде блок-схем с использованием графического метода разработки алгоритма для каждой из этих задач, приведенных ниже.

Напишите программу разветвляющихся алгоритмов на языке программирования Python.

Постановка задачи	
Блок-схема	Программный код
Результат программы	

6

Делимся мыслями

Какие пословицы вы знаете о природе?

- 1 – про животных,
- 2 – про землю и воду,
- 3 – про горы и камни,
- 4 – про лес,
- 5 – про растения.

Придумайте блок-схему и код программы, которая выдает не менее 3 пословиц в соответствии с введенным с клавиатуры числом от 1 до 5.

7

Домашнее задание

Подготовьте презентацию на тему «Язык программирования Python».

Включите в презентацию информацию об особенностях языка программирования Python, его отличиях от других языков программирования, его эффективности, недостатках и возможностях.

Презентация должна состоять из 8–10 слайдов.

§ 31–32. Тестирование программы

Вспомните!

- Что такое программа?
- Какие языки программирования вы знаете?
- Что такое программирование алгоритма?

Вы узнаете:

- о тестировании программы;
- об отладке программы;
- о тестировании разветвляющихся алгоритмов.

Тестирование – *Тестилеу* – Testing

Отладка – *Түзету* – Debugging

Ошибка – *Қате* – Error

Сообщение – *Хабарлама* – Message

В этом параграфе мы рассмотрим следующий этап создания программы для выполнения ее на компьютере – отладка и тестирование программы.

Тестирование – это процесс выполнения программы на некотором наборе данных, для которого заранее известен результат или известны правила поведения этих программ.

Отладка – это действие, направленное на обнаружение и исправление ошибок в программе.

Отладка и тестирование программы заключается в поиске и устранении синтаксических и логических ошибок в программе. В ходе синтаксического контроля программы транслятором выявляются конструкции и сочетания символов, недопустимые с точки зрения правил их построения или написания, принятых в данном языке.

Сообщения об ошибках компьютер выдает программисту, при этом вид и форма выдачи подобных сообщений зависят от вида языка программирования и версии используемого транслятора. После устранения синтаксических ошибок проверяется логика работы программы в процессе ее выполнения с конкретными исходными данными.

Несмотря на то, что создание программы происходит в несколько этапов, наиболее важным и трудоемким является этап отладки и тестирования программы. Именно на этом этапе устраняются все логические и синтаксические ошибки в создаваемой программе.

Синтаксические ошибки – это ошибки в синтаксисе последовательности символов или знаков, которые предназначены для записи на определенном языке программирования.

Логические ошибки – это ошибки, возникающие из-за ошибки в алгоритме или логике программы.

Успех отладки в значительной степени предопределяет рациональная организация тестирования. При отладке отыскиваются и устраняются, в основном, те ошибки, наличие которых устанавливается при тестировании. Как было уже отмечено, тестирование не может доказать правильность программы, в лучшем случае оно может продемонстрировать наличие в нем ошибки. Программа, не содержащая синтаксических ошибок, тем не менее, может содержать логические ошибки, не позволяющие программе выполнять заложенные в ней функции. Логические ошибки могут происходить из-за неправильного понимания алгоритма программы (*схема 15*).

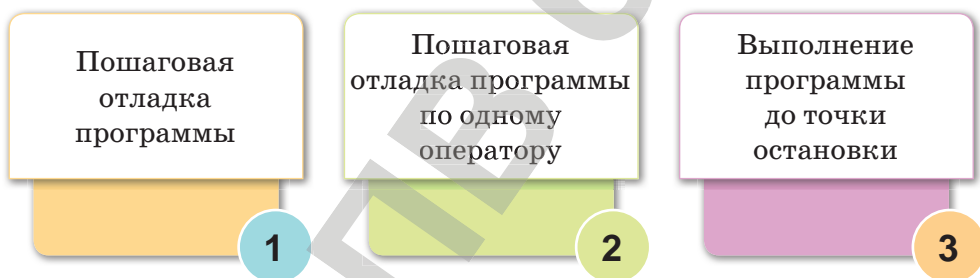


Схема 15. Способы отладки программы

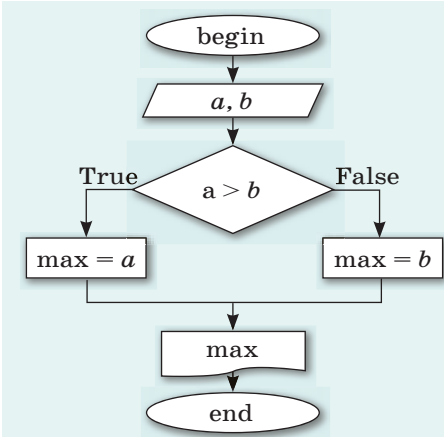
Отладка программы заключается в проверке правильности работы программы и аппаратуры.

Рассмотрим несколько задач, решение которых получается с помощью разветвляющихся алгоритмов.

Задача 1. Даны два числа. Найдите наибольшее из двух указанных чисел.

1. Постановка задачи: исходными данными являются переменные a и b . Их значения будут задаваться вводом. Значение большего из чисел должно быть присвоено переменной \max и выведено. Например, если $a = 4$, $b = 10$, то должно получиться: $\max = 10$.

2. Разработка алгоритма



3. Программирование алгоритма

```

a=int(input())
b=int(input())
if a>b:
    max=a
else:
    max=b
print('max=',max)
  
```

4. Тестирование программы:

Тестирование и отладку программы проверим путем заполнения таблицы пошагового выполнения алгоритма.

1) Результаты тестирования пошагового выполнения алгоритма для значений $a = 4$, $b = 10$:

Шаг	Операция	a	b	max	Проверка условия
1	Ввод a, b	4	10	–	
2	Сравнение $a > b$	4	10	–	ложь
3	$\text{max} = b$	4	10	10	
4	Вывод max	4	10	10	

2) Результат тестирования пошагового выполнения алгоритма для значений $a = 10$, $b = 4$:

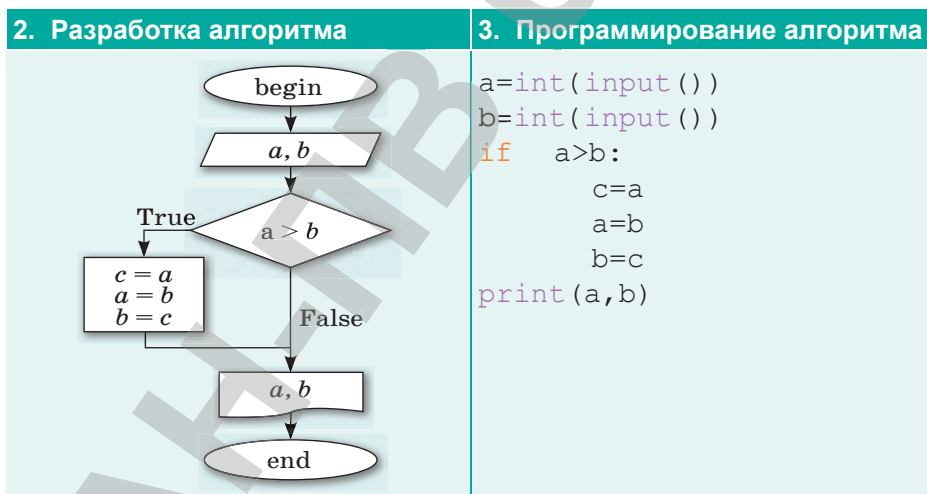
Шаг	Операция	a	b	max	Проверка условия
1	Ввод a, b	10	4	–	
2	Сравнение $a > b$	10	4	–	истина
3	$\text{max} = a$	10	4	10	
4	Вывод max	10	4	10	

3) Результат тестирования пошагового выполнения алгоритма для значений $a = 10$, $b = 10$:

Шаг	Операция	a	b	max	Проверка условия
1	Ввод a, b	10	10	–	
2	Сравнение a>b	10	10	–	ложь
3	max=b	10	10	10	
4	Вывод max	10	10	10	

Задача 2. Упорядочить значения двух переменных a и b по возрастанию.

- Постановка задачи:** если для исходных значений переменных справедливо отношение $a \leq b$ (например, $a = 10$, $b = 20$), то оставить их значения без изменения; если же $a > b$ (например, $a = 20$, $b = 10$), то выполнить обмен значениями. Для обмена значений необходимо использование третьей, вспомогательной переменной c . В алгоритме решения данной задачи используется неполное ветвление.



4. Тестирование программы:

- Результат тестирования пошагового выполнения алгоритма для значений $a = 4$, $b = 10$:

Шаг	Операция	a	b	c	Проверка условия
1	Ввод a, b	4	10	–	
2	Сравнение a>b	4	10	–	ложь
3	Вывод a, b	4	10	4, 10	

- 2) Результат тестирования пошагового выполнения алгоритма для значений $a = 10, b = 4$:

Шаг	Операция	a	b	c	Проверка условия
1	Ввод a, b	10	4	–	
2	Сравнение $a > b$	10	4	–	истина
3	$c = a$ $a = b$ $b = c$	10	4	$a = 4$ $b = 10$ $c = 10$	
4	Вывод a, b	10	4	4, 10	

- 3) Результат тестирования пошагового выполнения алгоритма для значений $a = 4, b = 4$:

Шаг	Операция	a	b	c	Проверка условия
1	Ввод a, b	4	4	–	
2	Сравнение $a > b$	4	4	–	ложь
3	Вывод a, b	4	4	4, 4	

1

Отвечаем на вопросы

1. Что такое тестирование?
2. Какие действия проводятся в отладке?
3. Какие основы отладки и тестирования программы вы знаете?
4. Какие бывают ошибки, возникающие в результате тестирования?
5. Что такое синтаксические ошибки?
6. Что такое логические ошибки?
7. Какие способы отладки программы вы знаете?

2

Думаем и обсуждаем

1. Почему этап отладки и тестирования программы считается важным?
2. В чем важность правильного тестирования программы?

3

Анализируем и сравниваем

Сравните синтаксические и логические ошибки, возникшие при отладке и тестировании программы.

Синтаксические ошибки	Логические ошибки

4

Выполняем в тетради

Заполните в тетрадях способы отладки программы.

- 1) ...
- 2) ...
- 3) ...

5

Выполняем на компьютере

1. Проверьте в программе Python результаты тестирования задач 1 и 2, приведенных в параграфе.
2. Продолжим работу над 5 задачами, которые приводятся в параграфе 28 в разделе «Выполняем на компьютере». Откройте файлы постановки задач, разработанные алгоритмы в виде блок-схем с использованием графического метода разработки алгоритма и программы этих задач на языке программирования Python. Заполните таблицу для 5 задач. Затем проведите отладку и тестирование для каждой программы.

Задача	
1. Постановка задачи	
2. Разработка алгоритма	3. Программирование алгоритма
4. Тестирование программы	
1)	
2)	
... приведите возможные значения и проверьте результаты тестирования программы на языке Python	

6

Делимся мыслями

С какими конкретными примерами тестирования вы сталкиваетесь в повседневной жизни? Можно ли использовать это во время урока?

7

Домашнее задание

Подготовьте отчет о том, как вы исправили синтаксические и логические ошибки, обнаруженные при тестировании программы задач.

§ 33–34. Практикум. Практическое программирование

Цель работы: записывать разветвляющиеся алгоритмы на языке программирования Python.

Инструкция по выполнению задания:

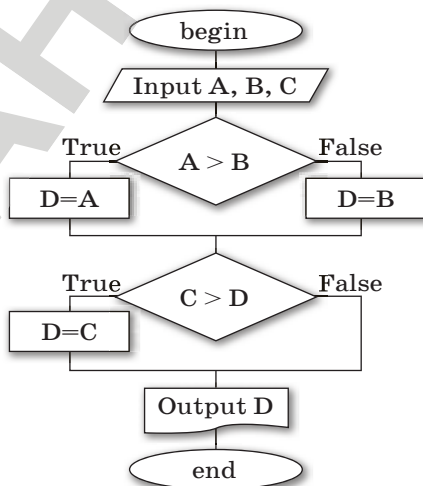
1. Используйте шаблон оформления при решении задачи.
2. Сначала сформулируйте постановку задачи.
3. Создайте блок-схему для решения задачи.
4. Напишите программу на языке программирования Python в соответствии с блок-схемой.
5. Проверьте все условия, указанные в задаче, проведите работы по отладке и тестированию программы и исправьте все ошибки.
6. Поместите результат задачи на образец.

Уровень А

1. **Задача.** Составить алгоритм определения наибольшего значения среди трех величин: A , B , C .

Постановка задачи: Сначала нужно найти большее из значений A и B и присвоить его вспомогательной переменной, например D ; затем найти большее значение среди D и C . Найденное большее значение можно присвоить все той же переменной D , которая в результате получит наибольшее значение среди всех трех величин.

2. **Разработка алгоритма:**



3. Программирование алгоритма:

```

A = int (input ())
B = int (input ())
C = int (input ())
if A>B:
    D=A
else:
    D=B
if C>D:
    D=C
print ('D=', D)

```

4. Тестирование программы:

1) Результат тестирования пошагового выполнения алгоритма для значений A=5, B=6, C=7:

Шаг	Операция	A	B	C	D	Проверка условия	Результат программы
1	Ввод A, B, C	5	6	7	–		5
2	Сравнение A>B	5	6	7	–	ложь	6
3	D=B	5	6	7	6		7
4	Сравнение C>D	5	6	7	6	истина	D=7
5	D=C	5	6	7	7		
6	Вывод D	5	6	7	7		

2) Результат тестирования пошагового выполнения алгоритма для значений A=7, B=6, C=5:

Шаг	Операция	A	B	C	D	Проверка условия	Результат программы
1	Ввод A, B, C	7	6	5	–		7
2	Сравнение A>B	7	6	5	–	истина	6
3	D=A	7	6	5	7		5
4	Сравнение C>D	7	6	5	7	ложь	D=7
5	Вывод D	7	6	5	7		

- 3) Результат тестирования пошагового выполнения алгоритма для значений $A=6, B=7, C=5$:

Шаг	Операция	A	B	C	D	Проверка условия	Результат программы
1	Ввод A, B, C	6	7	5	–		6
2	Сравнение $A > B$	6	7	5	–	ложь	7
3	$D=B$	6	7	5	7		5
4	Сравнение $C > D$	6	7	5	7	ложь	$D=7$
5	Вывод D	6	7	5	7		

- 4) Результат тестирования пошагового выполнения алгоритма для значений $A=5, B=7, C=6$:

Шаг	Операция	A	B	C	D	Проверка условия	Результат программы
1	Ввод A, B, C	5	7	6	–		5
2	Сравнение $A > B$	5	7	6	–	ложь	7
3	$D=B$	5	7	6	7		6
4	Сравнение $C > D$	5	7	6	7	ложь	$D=7$
5	Вывод D	5	7	6	7		

- 5) Результат тестирования пошагового выполнения алгоритма для значений $A=5, B=5, C=5$:

Шаг	Операция	A	B	C	D	Проверка условия	Результат программы
1	Ввод A, B, C	5	5	5	–		5
2	Сравнение $A > B$	5	5	5	–	ложь	5
3	$D=B$	5	5	5	5		5
4	Сравнение $C > D$	5	5	5	5	ложь	$D=5$
5	Вывод D	5	5	5	5		

Уровень В

Определите значение следующих логических выражений (True, False) для разных значений переменных x, y .

Следуйте инструкциям для задания.

- A. $(x \geq 0) \text{ and } (y * y \leq 4)$
- B. $(x \geq 0) \text{ or } (y * y \leq 4)$
- C. $(x * y \leq 0) \text{ and } (x > y)$
- D. $(x * y \leq 0) \text{ or } (x < y)$
- E. $(\text{not}(x * y > 0)) \text{ and } (y > x)$
- F. $(\text{not}(x * y < 0)) \text{ or } (y < x)$

Уровень С

1. Даны координаты точки x и y . Необходимо определить, в какой четверти системы координат находится точка. В результате выполнения программы число должно быть «1», если точка находится в первой четверти «2», если оно во второй четверти, «3», если оно находится в третьей четверти, и «4», если оно находится в четвертой четверти. Следуйте инструкциям для задания.
2. Дано трехзначное число. Необходимо определить, является ли сумма чисел трехзначного числа четной или нечетной. В результате выполнения программы, если сумма нечетная, то появляется сообщение «нечетное», если четное, сообщение «четное». Следуйте инструкциям для задания.

ЗАДАНИЯ ПО СУММАТИВНОМУ ОЦЕНИВАНИЮ К РАЗДЕЛУ 5

- 1. Заполните последовательность этапов решения задач на компьютере:**
 - А. Анализ результатов. Уточнение модели ____
 - В. Постановка задачи ____
 - С. Создание программы ____
 - Д. Математическое или информационное моделирование ____
 - Е. Отладка и тестирование программы ____
 - Ф. Создание алгоритма ____
- 2. Полное и точное предписание исполнителю выполнить определенную последовательность действий – это ...**
 - А. Алгоритм.
 - В. Программа.
 - С. Команда.
 - Д. Операнд.
 - Е. Код.
- 3. Исполнители алгоритма.**
 - А. Человек.
 - В. Машина.
 - С. Автоматическое устройство.
 - Д. Блок-схема.
 - Е. Фигура.
- 4. Сопоставьте свойства алгоритма:**

1.	Алгоритм должен быть разбит на конечное число шагов	А	Результативность
2.	Каждый шаг алгоритма должен быть четким	В	Массовость
3.	Алгоритм за конечное число шагов должен приводить исполнителя к решению задачи	С	Понятность
4.	Алгоритм решения задачи разрабатывается в общем виде	Д	Определенность
5.	Каждый шаг алгоритма должен быть понятен исполнителю	Е	Дискретность

5. Способы разработки алгоритма.

- А. Словесное описание.
- В. Графический.
- С. Специальный алгоритмический язык.
- Д. Табличный.
- Е. Цифровой.

6. Заполните пропуски:

При графическом способе алгоритм изображается ... ,

7. Запись алгоритма решения задачи в виде последовательности команд или операторов языком, который понимает компьютер, – это ...

- А. Программа.
- В. Алгоритм.
- С. Команда.
- Д. Код.
- Е. Блок-схема.

8. Формальная знаковая система, предназначенная для записи программ

- А. Язык программирования.
- В. Алгоритм.
- С. Команда.
- Д. Программный код.
- Е. Блок-схема.

9. Сопоставьте:

1 Тестирование	А	ошибки, возникающие из-за ошибки в алгоритме или логике программы
2 Отладка	В	ошибки в синтаксисе последовательности символов или знаков
3 Синтаксические ошибки	С	деятельность, направленная на обнаружение и исправление ошибок в программе
4 Логические ошибки	Д	процесс выполнения программы на некотором наборе данных

10. Заполните пропуски:

Отладка и тестирование программы заключается в поиске ... и ... синтаксических и ... ошибок в программе.

ИТОГИ 5 РАЗДЕЛА

В разделе «Практическое программирование» вы познакомились с такими понятиями, как формирование задачи, цель, результат задачи, алгоритм, исполнители алгоритма, свойства алгоритма, программа, исполнители программы, язык программирования, тестирование, исправление, ошибка, сообщение. Также вы рассмотрели процесс создания отчета на компьютере, действия, связанные с человеком и компьютером при создании отчета на компьютере, способы разработки алгоритма и графический метод разработки алгоритма, основные геометрические блоки и их функции, действия, выполняемые при программировании алгоритма, программирование разветвленных алгоритмов на языке программирования Python, синтаксические и логические ошибки.

Были предложены практикумы по формулированию проблемы, разработке алгоритма, программированию алгоритма, решению практических задач по темам тестирования программы, написанию разветвленных алгоритмов на языке программирования Python с приведением различных примеров.

Глоссарий

Microsoft Excel – прикладная программа, которая предназначена для создания электронных таблиц.

Автозаполнение – автоматический ввод некоторых данных в определенные диапазоны.

Алгоритм – полное и точное предписание исполнителю выполнить определенную последовательность действий, направленных на решение поставленной задачи или достижение указанной цели.

Антивирусная программа (или антивирусное средство защиты, средство обнаружения вредоносных программ, антивирус) – специальная программа, предназначенная для выявления компьютерных вирусов, а также нежелательных (считающихся вредоносными) программ, восстановления зараженных (модифицированных) файлов и предотвращения заражения файлов или операционной системы вредоносным кодом.

Бинарный файл – файл, в котором данные отображаются в закодированной форме.

Бит – наименьшая единица измерения информации.

Внешняя память – устройство для длительного хранения программ и информации (магнитные, оптические диски и флеш-памяти) с помощью носителей информации.

Внутренняя память – электронное устройство, расположенное внутри компьютера. Оно предназначено для хранения информации и программ в процессе работы.

Гибкий диск – носитель информации небольшой емкости, представляющий собой гибкий пластиковый диск в защитной оболочке.

Диаграмма – графическое изображение, наглядно демонстрирующее соотношение каких-либо величин.

Жесткий диск – запоминающее устройство большой емкости.

Извлечение из архива – процесс восстановления файлов из архива до первоначального состояния до загрузки в архив.

Исполнители программы – компьютер или другое автоматическое устройство.

Компьютерные вирусы – специальная подпрограмма, которая обеспечивает скрытый доступ к компьютеру (коду программ, памяти системы, секторам загрузки), блокирует работу компьютера или наносит вред данным и системам работы.

Компьютерная сеть – разветвленная система обработки информации, предназначенная для общения с помощью средств связи двух или более компьютеров, которые могут обмениваться данными друг с другом.

Кэш-память – устройство, которое используется при обмене данными между процессором и оперативной памятью.

Логическая операция И – операция, которая требует одновременного выполнения двух условий.

Логическая операция ИЛИ – операция, которая требует выполнения хотя бы одного из условий.

Логическая операция НЕ – операция, которая меняет значение высказывания на противоположное: истинно на ложно, а ложно на истинно.

Логические ошибки – ошибки, возникающие из-за ошибки в алгоритме или логике программы.

Оперативное запоминающее устройство, или оперативная память – устройство записи, чтения и быстрого временного хранения данных для компьютерных вычислений.

Отладка – деятельность, направленная на обнаружение и исправление ошибок в программе.

Память компьютера (внешняя или внутренняя) – часть вычислительной машины, физическое устройство или среда для хранения данных, используемых в вычислениях, в течение определенного времени.

Пользовательский интерфейс – совокупность средств и правил взаимодействия человека и компьютера.

Постановка задач – точное и четкое определение требуемых результатов и исходных условий в задачах.

Постоянное запоминающее устройство – читающая память для постоянного запоминания информации.

Программа – запись алгоритма решения задачи в виде последовательности команд или операторов языком, который понимает компьютер.

Сжатие информации – процесс, в результате которого в несколько раз уменьшается размер файлов, что позволяет сэкономить память компьютера.

Синтаксические ошибки – ошибки в синтаксисе последовательности символов или знаков, которые предназначены для записи на определенном языке программирования.

Составное условие – состоит из двух или нескольких простых отношений (условий), которые объединяются с помощью логических операций.

Строка формулы – панель, используемая для ввода и изменения текста и формул в ячейках листа.

Таблица – способ представления данных путем размещения на пересечении строк и столбцов.

Текстовый файл – файл, в котором хранятся последовательности символов, которые понимает человек.

Тестирование – процесс выполнения программы на некотором наборе данных, для которого заранее известен результат или известны правила поведения этих программ.

Тип данных – определенная характеристика содержимого в таблице Excel, благодаря которой программа идентифицирует информацию и понимает, каким образом ее следует обрабатывать и отображать.

Тип файла – описание файла, система, описывающая к какому типу относится и в какой программе он открывается.

Условный оператор – особая конструкция языка программирования, в которой присутствует ветвление.

Условное форматирование – инструмент для выделения значений таблицы по своим условным параметрам.

Файл – совокупность данных, хранящихся на компьютере в виде бит-цепочки.

Файловая система – каталог, определяющий способ организации, хранения и именования данных, а также задающий ограничения на формат и доступ к данным.

Язык программирования – формальная знаковая система, предназначенная для записи программ.

Использованная и рекомендуемая литература

1. Бриггс Джейсон. Python для детей. Самоучитель по программированию. – Москва: Огни, 2013. – 177 с.
2. Бэрри Пол. Изучаем программирование на Python. – М.: Эксмо, 2016. – 332 с.
3. Васильев А. Н. Python на примерах. Практический курс по программированию. – М.: Наука и техника, 2016. – 432 с.
4. Васильев А. Н. Python на примерах. Практический курс по программированию. Руководство. – М.: Наука и техника, 2017. – 752 с.
5. Гуриков С.Р. Основы алгоритмизации и программирования на Python. – М.: Форум, 2018. – 991 с.
6. Гуриков С.Р. Основы алгоритмизации и программирования на Python. Учебное пособие. Гриф МО РФ. – М.: Инфра-М, Форум, 2018. – 707 с.
7. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 277 с.
8. МакГрат Майк. Python. Программирование для начинающих. – М.: Эксмо, 2013. – 727 с.
9. Мэтиз Эрик. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – М.: Питер, 2017. – 551 с.
10. Мэтиз Эрик. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – М.: Питер, 2018. – 760 с.

Электронные ресурсы

- | | |
|---|---|
| 1. bilimland.kz | 8. it-uroki.ru/ |
| 2. nauka.kz | 9. programmer-lib.ru |
| 3. nis.edu.kz | 10. pythonworld.ru |
| 4. bilim.kz | 11. pythonicway.com |
| 5. www.nci.kz | 12. python.org |
| 6. codecademy.com | 13. pythonru.com |
| 7. yaklass.ru | 14. learnpython.org |

Содержание

Предисловие	4
Раздел 1. Измерение информации и компьютерная память	5
§ 1. Единицы измерения информации	6
§ 2. Практикум. Единицы измерения информации	14
§ 3. Память компьютера.....	16
§ 4. Размер файлов.	
Сравнение размеров файлов разных форматов	23
§ 5. Размер файлов.	
Создание и открытие архивов различных форматов.....	33
Задания по суммативному оцениванию к разделу 1	42
Итоги 1 раздела	44
Раздел 2. Сети и безопасность	45
§ 6. Компьютерные сети и их классификация	46
§ 7. Антивирусная безопасность	52
§ 8. Пользовательский интерфейс	58
§ 9. Практикум. Пользовательский интерфейс	68
Задания по суммативному оцениванию к разделу 2	70
Итоги 2 раздела	72
Раздел 3. Решение задач с помощью электронных таблиц	73
§ 10. Таблицы в текстовом процессоре	74
§ 11. Практикум. Таблицы в текстовом процессоре	81
§ 12. Форматирование элементов электронных таблиц	83
§ 13. Типы данных. Использование различных типов данных в электронной таблице.....	91
§ 14. Типы данных. Создание формулы для вычислений в электронной таблице.....	99
§ 15. Условное форматирование	107
§ 16. Графическое представление табличных данных	115
§ 17. Практикум.	
Моделирование процессов в электронных таблицах.....	123
Задания по суммативному оцениванию к разделу 3	126
Итоги 3 раздела	128

Раздел 4. Программирование алгоритмов

на языке Python	129
§ 18. Работа с файлами	130
§ 19. Практикум. Работа с файлами	141
§ 20. Программирование алгоритмов ветвления.....	144
§ 21. Практикум. Программирование алгоритмов ветвления.....	150
§ 22. Программирование вложенных условий.....	152
§ 23. Практикум. Программирование вложенных условий.....	159
§ 24. Программирование составных условий	161
§ 25. Практикум. Программирование составных условий	165
§ 26–27. Практикум. Организуем выбор.....	167
Задания по суммативному оцениванию к разделу 4.....	170
Итоги 4 раздела	172

Раздел 5. Практическое программирование..... 173

§ 28. Постановка задачи	174
§ 29. Разработка алгоритма.....	179
§ 30. Программирование алгоритма	184
§ 31–32. Тестирование программы.....	189
§ 33–34. Практикум. Практическое программирование	195
Задания по суммативному оцениванию к разделу 5.....	199
Итоги 5 раздела	201
Глоссарий.....	202
Использованная и рекомендуемая литература.....	205



Внимание

При необходимости вы всегда сможете найти CD с электронным приложением на сайте *arman-pv.kz* и загрузить его на свой компьютер для дальнейшей работы.

Учебное издание

**Гульназ Ибрагимовна Салгараева
Айгуль Сейсенбаевна Маханова
Лаззат Азретовна Реалина**

ИНФОРМАТИКА

Учебник для учащихся 7 класса уровня основного среднего образования по обновленному содержанию

Главный редактор	К. Караева
Редактор	Г. Маликова
Технический редактор	В. Бондарев
Художественный редактор	Е. Мельникова
Бильд-редактор	Ш. Есенкулова
Художник-оформитель	О. Подопригора
Дизайн обложки	В. Бондарев
Верстка	Л. Костина, С. Сулейменова Г. Илишева

По вопросам приобретения обращайтесь по следующим адресам:

г. Нур-Султан, м-н 4, д.2, кв.55

Тел.: 8 (7172) 92-50-50, 92-50-54. E-mail: astana@arman-pv.kz

г. Алматы, м-н Аксай 1А, д.28Б

Тел./факс: +7 (727) 243-04-39, 973-83-83, 973-63-63.

E-mail: info@arman-pv.kz

Сдано в набор 28.07.20. Подписано в печать 16.06.21. Формат 70 x 100¹/₁₆.

Бумага офсетная. Гарнитура «ММ Мектептік» Печать офсетная.

Объем 16,77 усл.печ.л. Тираж 30000 экз.

Артикул 807-002-001р-21