

С. Т. Мухамбетжанова, А. С. Тен, Л. Г. Демидова

ИНФОРМАТИКА

Учебник для 7 класса общеобразовательной школы

7

Рекомендовано Министерством образования и науки Республики Казахстан



Алматы «Атамұра» 2021

УДК 373.167.1
ББК 32.973 я 72
М 92

Учебник подготовлен в соответствии с Типовой учебной программой по предмету «Информатика» для 5–9 классов уровня основного среднего образования по обновленному содержанию, утвержденной Министерством образования и науки РК

Условные обозначения

	Самое главное по изучаемой теме		Для тех, кто хочет знать больше
	Шаг за шагом		Индивидуальная работа
	Знание. Понимание		Парная работа
	Применение		Групповая работа
	Анализ		Работа класса
	Синтез. Оценивание		Работа с диском
	Домашнее задание		

Мухамбетжанова С. Т. и др.

М 92 Информатика: Учебник для 7 кл. общеобразоват. шк. / С. Т. Мухамбетжанова, А. С. Тен, Л. Г. Демидова. – Алматы: Атамұра, 2021. – 208 с.

ISBN 978-601-331-942-1

УДК 373.167.1
ББК 32.973 я 72

ISBN 978-601-331-942-1

© Мухамбетжанова С. Т.,
Тен А. С., Демидова Л. Г., 2021
© «Атамұра», 2021

ОГЛАВЛЕНИЕ

Введение	5
----------------	---

РАЗДЕЛ I. ИЗМЕРЕНИЕ ИНФОРМАЦИИ И КОМПЬЮТЕРНАЯ ПАМЯТЬ

1.1. Единицы измерения информации	7
1.2. Перевод из одних единиц измерения информации в другие.....	14
1.3. Компьютерная память.....	18
1.4. Размеры файлов.....	26
1.5. Создание и распаковка архивов различных форматов.....	33
Тестовые задания к разделу I.....	39

РАЗДЕЛ II. СЕТИ И БЕЗОПАСНОСТЬ

2.1. Компьютерные сети и их классификация.....	41
2.2. Антивирусная безопасность.....	51
2.3. Пользовательский интерфейс	58
Тестовые задания к разделу II.....	67

РАЗДЕЛ III. РЕШЕНИЕ ЗАДАЧ С ПОМОЩЬЮ ЭЛЕКТРОННЫХ ТАБЛИЦ

3.1. Таблицы в текстовом процессоре.....	70
3.2. Форматирование элементов электронных таблиц	79
3.3. Типы данных.....	90
3.4. Условное форматирование в электронной таблице.....	99
3.5. Графическое представление табличных данных.....	106
3.6. Моделирование процессов в электронных таблицах	114
Тестовые задания к разделу III.....	122

РАЗДЕЛ IV. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON (ПАЙТОН)

4.1. Работа с файлами	125
-----------------------------	-----

4.2. Программирование алгоритмов ветвления.....	133
4.3. Программирование вложенных условий	139
4.4. Программирование составных условий	147
4.5. Организуем выбор	155
Тестовые задания к разделу IV.....	160

РАЗДЕЛ V. ПРАКТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

5.1. Постановка проблемы	165
5.2. Разработка алгоритма	171
5.3. Программирование алгоритма	178
5.4. Тестирование программы	187
Задания к разделу V	195
Глоссарий	197
Приложение	201
Список использованной литературы	206
Интернет-ресурсы	207

ВВЕДЕНИЕ

Дорогие семиклассники!

Сейчас, когда компьютеры, мобильные технологии, Интернет стали неотъемлемой частью современной жизни, школьный предмет «Информатика» становится проводником в мир информационных технологий, информационной культуры.

Материал, изучаемый на уроках информатики, поможет вам расширить кругозор и лучше понимать те процессы, которые связаны с передачей, приемом, преобразованием и хранением информации.

Для вас разработаны учебник и электронное приложение. Учебник написан в соответствии с Государственным общеобязательным стандартом образования. Основными целями изучения предмета информатики по обновленному содержанию образования являются:

- формирование базовых ИКТ-навыков для решения прикладных задач различных предметных областей, позволяющих применять полученные знания, умения и навыки в окружающей среде, в практике;
- развитие компетенций (креативности, коммуникабельности и умения решать проблемы при выполнении заданий исследовательского, экспериментального, проектного характера).

Электронное приложение – это учебный материал к учебнику и интерактивные задания, выполняя которые нужно перемещать объекты, вводить числа, выбирать ответы, устанавливать соответствия, вводить строки и т.д. Выполняя эти задания, вы сможете закрепить изученный материал, проверить знания, полученные на уроках информатики.

Условные обозначения, данные в начале учебника, позволят вам сориентироваться в формах и видах (индивидуальная, парная, групповая работа и работа в классе) работы. Они помогают структурировать содержание учебника, сделать процесс обучения (поиск ответов на поставленные вопросы, решение практических задач, взаимодействие учащихся друг с другом и учителем и т.д.) более познавательным и интересным.

Учебник состоит из пяти разделов. Изучив первый раздел, вы будете знать, в каких единицах измеряется информация, как определять размер файла и какие виды компьютерной памяти существуют.

Второй раздел научит вас классифицировать компьютерные сети и защищать компьютер от вредоносных программ.

Мощные возможности электронных таблиц, изученные в третьем разделе, помогут вам при обработке результатов экспериментов, самостоятельных ис-

следований, при моделировании процессов и математическом обосновании разрабатываемых проектов.

Четвертый раздел – программирование, самый сложный и самый интересный. Программирование – это часть цифровой грамотности (*digital literacy*), которой можно обучить любого человека, так же как читать и писать. Основы программирования будут формировать у вас особый структурный стиль мышления, полезный во всех областях жизни.

Пятый раздел – «Практическое программирование». Этот раздел познакомит вас с особенностями программирования. Вы научитесь ставить проблемы, разрабатывать и программировать алгоритмы, тестировать программы. Сведения и навыки, полученные в ходе изучения практического программирования, могут быть востребованы в вашей будущей практической деятельности.

В конце тем предложены вопросы, разноуровневые задания, составленные по шести уровням учебных целей таксономии Б. Блума. Уровень 1 (репродуктивный) – знание и понимание (т.е. восприятие, осмысление, запоминание) для всех учащихся. Уровень 2 (продуктивный) – применение знаний по образцу, решение типовых задач, анализ данных для большинства учащихся. Уровень 3 (творческий) – синтез и оценивание, т.е. применение знаний в новой ситуации, создание творческого продукта для некоторых учащихся.

Большая часть тем учебника имеет практико ориентированную направленность. Благодаря этому вы можете быть уверены в востребованности полученных вами знаний в будущем, а также быть готовыми для продолжения образования в современном высокотехнологичном мире.

Желаем успехов!

Ваши авторы

ИЗМЕРЕНИЕ ИНФОРМАЦИИ И КОМПЬЮТЕРНАЯ ПАМЯТЬ

1.1. ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ

ВЫ НАУЧИТЕСЬ

Осуществлять перевод из одних единиц измерения информации в другие.

КЛЮЧЕВЫЕ СЛОВА

Единицы измерения информации	Ақпараттың өлшем бірліктері	Units of information
Информативная информация	Ақпараттық ақпарат	Informative information
Неинформативная информация	Ақпараттық емес ақпарат	Uninformative information

С понятием «информация» вы уже знакомы. Знания, которые человек получает из различных источников, являются для него информацией. Информация может быть **информативной** и **неинформативной**. Информативная информация содержит ненулевую информацию и дополняет знания человека. Например, прогноз погоды на завтра будет информативным сообщением, а вчерашний прогноз погоды – неинформативным, так как он уже известен. В данном примере две ситуации: «есть информация» или «нет информации». Иными словами, количество информации может быть **равно нулю** или **не равно нулю**.

Для количественного определения любой величины необходимо определить единицу измерения. Например, для измерения длины в качестве единицы измерения выбран метр, массы – килограмм и т.д.

За единицу количества информации принят **1 бит** – количество информации, содержащееся в сообщении, уменьшающем неопределенность знаний в два раза.



Неопределенность знаний некоторого события – это количество возможных результатов события.

Ситуация неопределенности возникает при предсказании будущих событий или намерений других людей. Например, при бросании монеты реализуется одно из двух возможных событий: «орел» или «решка». Перед ее подбрасыванием неопределенность знания о результате события равна двум. После подбрасывания будет определен один из двух возможных результатов.

То есть неопределенность знания уменьшится в два раза – из двух вариантов останется один.

Сообщение об одном из двух равновероятных результатов некоторого события равно **1 биту** информации.

Таким образом, если в некотором сообщении содержатся сведения о том, что произошло одно из **N** равновероятных событий, тогда количество информации **i**, содержащееся в сообщении, можно определить по формуле Хартли: $N=2^i$.



Шахматная доска состоит из 8 столбцов и 8 строк, которые составляют 64 клетки. Какое количество бит содержит сообщение о выборе одного шахматного поля?

В 1947 году американский математик Джон Тьюки для обозначения единицы измерения информации ввел термин **бит** (*bit*). Сокращенное словосочетание *binari digit* означает двоичную цифру – **0** или **1**. Бит является наименьшей единицей измерения информации. На практике применяется единица измерения информации – **байт**.



Байт – наименьшая единица обработки и передачи информации.

$$1 \text{ байт} = 8 \text{ бит} = 2^3 \text{ бит}$$

Например, если объем съемного носителя (флешки) равен 8 Гигабайт, то это составляет 68 719 476 736 бит. Такое большое число неудобно использовать. Поэтому биты необходимо переводить в более крупные единицы измерения информации, такие как **килобайт**, **мегабайт**, **гигабайт** и т.д.

Принята следующая система единиц измерения количества информации (табл. 1.1).

Таблица 1.1. Единицы измерения информации

1 байт (байт)	8 бит		
1 килобайт (Кбайт)	2^{10} байт	1024 байт	
1 мегабайт (Мбайт)	2^{10} Кбайт	1024 Кбайт	2^{20} байт
1 гигабайт (Гбайт)	2^{10} Мбайт	1024 Мбайт	2^{30} байт
1 терабайт (Тбайт)	2^{10} Гбайт	1024 Гбайт	2^{40} байт
1 петабайт (Пбайт)	2^{10} Тбайт	1024 Тбайт	2^{50} байт
1 эксабайт (Эбайт)	2^{10} Пбайт	1024 Пбайт	2^{60} байт
1 зеттабайт (Збайт)	2^{10} Эбайт	1024 Эбайт	2^{70} байт
1 йоттабайт (Йбайт)	2^{10} Збайт	1024 Збайт	2^{80} байт

На схеме (рис. 1.1) перевода системы единиц измерения количества информации показано, что каждое следующее число больше предыдущего в 1024 (2^{10}) раза.



Рис. 1.1. Единицы измерения информации

Переведите 1 петабайт, 1 эксабайт, 1 зеттабайт, 1 йоттабайт в байты.



Известно, что 1 байт информации может содержать 256 различных кодов, что вполне достаточно для кодирования одного знака текста. При кодировании числовой, звуковой или графической информации может использоваться большое количество байтов.

С помощью одного байта можно записать двоичные коды 256 (2^8) чисел от 0 до 255.

Рассмотрим решение задач перевода из одних единиц измерения в другие.

Шаг за шагом



1. Сколько бит информации содержится в сообщении «информация»?

Дано:

«Информация» – 10 символов

Найти:

Количество бит в сообщении – ?

1 символ = 1 байт

Решение: 1 байт = 8 бит.

$10 \cdot 8 = 80$ бит

Ответ: В сообщении содержится 80 бит информации.

2. Измерим количество информации в следующем предложении: «Сабит Муканов – классик казахской литературы, поэт, общественный деятель, академик, председатель Союза писателей Казахстана.». Любой символ текста занимает 1 байт. Текст содержит 105 букв, 6 знаков препинания, 13 пробелов.

Дано:

1 символ = 1 байт

Сообщение – *Сабит Муканов* – классик казахской литературы, поэт, общественный деятель, академик, председатель Союза писателей Казахстана.

Найти:

Количество информации в сообщении – ?

Решение:

Количество букв = 105

Количество знаков препинания = 6

Количество пробелов = 13

$105 + 6 + 13 = 124$ байта

Ответ: В сообщении содержится 124 байта.

**Знание. Понимание**

1. Для чего нужны единицы измерения информации?
2. Что принято за единицу измерения количества информации?
3. Что такое *байт*, *килобайт*, *мегабайт* и *гигабайт*?
4. Как они связаны между собой?
5. Расположите единицы измерения информации в порядке возрастания: Тбайт; Кбайт; байт; Мбайт; Гбайт.

**Применение**

6. Сколько бит содержится в слове «Казахстан»?
7. Аскар задает Артему вопросы для определения месяца рождения. Какое количество вопросов достаточно задать, чтобы наверняка определить месяц, в котором родился Артем?

**Анализ**

8. Проанализируйте информацию, поставив вместо вопросительных знаков числа.
 - а) 1 байт = ? бит = 2^3 бит.
 - б) 4096 бит = ? Кбайт.
 - в) 1 Мбайт = ? Кбайт = ? байт = ? бит.
 - г) ? Кбайт = ? байт = 12 288 бит.
 - д) 1 Мбайт = ? Кбайт = ? байт.
 - е) 1 Гбайт = ? Мбайт = ? Кбайт.
 - ж) 2 Тбайт = ? Гбайт = ? Мбайт.
 - з) ? Гбайт = 1536 Мбайт = ? Кбайт.

Синтез



9. Заполните пустые ячейки в таблице.

Таблица 1.2. Перевод информации из одной единицы измерения в другую

Биты	Байты	Килобайты	Мегабайты	Гигабайты
				0,5
			256	
2^{23}				
		512		
	2^{20}			

10. В тетради по казахскому языку содержится 524 288 символов. Посчитайте, сколько информации содержится в тетради в мегабайтах.

Оценивание



11. Расшифруйте фамилию известного ученого, именем которого был назван язык программирования (табл. 1.3). Для раскодировки слова решите примеры, а затем используйте фрагмент дополнительной части кодовой таблицы ASCII (табл. 1.4). Прочитав фамилию, определите, сколько байтов содержит информация.

Таблица 1.3

Пример	Число	Буква
$(200 \cdot 2 + 7) - 200 =$		
$448 : 4 \cdot 2 =$		
$27 \cdot 5 + 106 =$		
$72 : 9 + 10^2 + 52 : 2 =$		
$112 : 2 + 224 : 2 + 56 =$		
$15 \cdot 8 + 9 \cdot 9 + 34 =$		
$12 \cdot 12 + 108 =$		
Фамилия ученого: _____	Количество байтов: _____	

Таблица 1.4. Фрагмент дополнительной части кодовой таблицы ASCII

Символ	10-й код	2-й код	Символ	10-й код	2-й код
А	192	11000000	а	224	11100000
Б	193	11000001	б	225	11100001
В	194	11000010	в	226	11100010
Г	195	11000011	г	227	11100011
Д	196	11000100	д	228	11100100
Е	197	11000101	е	229	11100101
Ж	198	11000110	ж	230	11100110
З	199	11000111	з	231	11100111
И	200	11001000	и	232	11101000
Й	201	11001001	й	233	11101001
К	202	11001010	к	234	11101010
Л	203	11001011	л	235	11101011
М	204	11001100	м	236	11101100
Н	205	11001101	н	237	11101101
О	206	11001110	о	238	11101110
П	207	11001111	п	239	11101111
Р	208	11010000	р	240	11110000
С	209	11010001	с	241	11110001
Т	210	11010010	т	242	11110010
У	211	11010011	у	243	11110011
Ф	212	11010100	ф	244	11110100
Х	213	11010101	х	245	11110101
Ц	214	11010110	ц	246	11110110
Ч	215	11010111	ч	247	11110111
Ш	216	11011000	ш	248	11111000
Щ	217	11011001	щ	249	11111001
Ъ	218	11011010	ъ	250	11111010
Ы	219	11011011	ы	251	11111011
Ь	220	11011100	ь	252	11111100
Э	221	11011101	э	253	11111101
Ю	222	11011110	ю	254	11111110
Я	223	11011111	я	255	11111111

Жизненный цикл информации



Стремительное развитие технических и информационных технологий порождает много проблем. Например, если данные хранятся на магнитных лентах, то срок жизни информации составляет около 25 лет, на компакт-дисках – около 100 лет, на флеш-носителях информация хранится не дольше 100 лет. Способы считывания информации с жестких дисков меняются каждые 15 лет. Информация, представленная на видеолентах, подчиняется закону размагничивания (подобно дискетам и жестким дискам).

Не меньше проблем и с хранением информации в печатной форме. Распечатки с помощью струйного принтера существуют не более 10 лет. Дольше хранятся распечатки, сделанные на лазерных принтерах, – до 15 лет. Дольше всего сохраняется офсетная печать. Но и в этом случае утрата информации произойдет через 50–100 лет.

Можно ли будет найти информацию через 200 лет? Специалистами введен термин «жизненный цикл информации». Он имеет отношение к изменению информации во времени и состоит из следующих частей:

- **рождение информации** – создание самого содержания информации;
- **развитие (накопление) информации** – запись информации на определенные информационные носители;
- **старение информации** – уменьшение объема, потеря определенных связей внутри информационного содержания;
- **гибель информации** – уменьшение информационного объема до нуля бит.

Проблемы, связанные с жизненным циклом информации, пока еще до конца не изучены.

Неопределенность

В мире, в котором всё заранее предсказуемо, не возникает неопределенностей. Такой мир стал бы однозначно определенным, и в нем не было бы ничего нового и всё сводилось бы к вечному повторению. Первые исследования проблемы неопределенности были предприняты в середине XVII века выдающимися французскими математиками Б. Паскалем и П. Ферма. Ученые по просьбе игроков в кости провели анализ их шансов победить. В дальнейшем для анализа ситуаций неопределенности и прогнозирования событий будущего теория вероятности стала важнейшим теоретическим инструментом.

1.2.

ПЕРЕВОД ИЗ ОДНИХ ЕДИНИЦ ИЗМЕРЕНИЯ ИНФОРМАЦИИ В ДРУГИЕ

ВЫ НАУЧИТЕСЬ

Осуществлять перевод одних единиц измерения информации в другие.

КЛЮЧЕВЫЕ СЛОВА

Единицы измерения информации	Ақпараттың өлшем бірліктері	Units of information
Перевод	Аудару	Translation

Рассмотрим несколько примеров перевода из одних единиц измерения информации в другие.



Подсчитайте информационный объем сообщения «ИНФОРМАЦИЯ» в битах. Если 1 символ равен одному байту, то чему равно количество байтов в сообщении «ИНФОРМАЦИЯ»?

Вы уже знаете, что **1 байт = 8 бит**. А сколько бит в двух, трех, пяти, десяти байтах? Как перевести байты в биты?

$$2 \text{ байта} = 2 \times 8 = 16 \text{ бит}$$

$$3 \text{ байта} = 3 \times 8 = 24 \text{ бита}$$

$$5 \text{ байт} = 5 \times 8 = 40 \text{ бит}$$

Следовательно, для того чтобы узнать, сколько бит в **N** байт, нужно применить формулу:

$$\mathbf{N \text{ байт} = N \cdot 8 \text{ бит}}$$

Как выполнить обратную задачу, то есть перевести биты в байты? Например, как узнать, сколько байт в 80 битах? Для этого нужно 80 разделить на 8. Для того, чтобы перевести биты в байты, нужно количество бит разделить на 8.

$$\mathbf{N \text{ бит} = N : 8 \text{ байт}}$$

По аналогии с рассмотренными нами примерами можно составить схему перевода их одних единиц измерения в другие.



Рис. 1.2. Схема перевода из одних единиц измерения в другие

Сформулируйте правило перехода от крупных единиц к более мелким (от байтов к битам) и наоборот.



Информационный объем сообщения или **информационная емкость сообщения** – это количество информации в сообщении, измеряемое в битах, байтах, Кбайтах, Мбайтах и т.д.



Для определения информационного объема сообщения применяется формула:

$$V = K \cdot i, \text{ где}$$

V – объем информации

K – количество символов в сообщении

i – количество бит, необходимых для кодирования одного символа.

Шаг за шагом



1. Рассчитайте объем информации (в Кбайтах), требуемый для хранения рисунка, который состоит из 1200 цветных точек по горизонтали и 600 точек по вертикали, а каждая цветная точка кодируется одним байтом, т.е. может иметь один из 256 различных цветов.

Решение:

Сначала выясним, сколько всего точек содержит рисунок. Для этого выполним:

$$1200 \cdot 600 = 720\,000 \text{ точек} = 720\,000 \text{ байт.}$$

Переведем байты в килобайты:

$$720\,000 : 1024 = 703,125 \text{ Кбайт.}$$

Ответ: Рисунок требует для хранения 703,125 Кбайт.

2. Выясните, сколько килобайт информации займет книга, которая имеет 200 страниц по 60 строк на странице и 80 символов в строке.

Решение:

Определим, сколько символов размещается на одной странице:

$$60 \cdot 80 = 4800 \text{ символов.}$$

Определим, сколько символов (байт) в книге:

$$4800 \cdot 200 = 960\,000 \text{ символов} = 960\,000 \text{ байт.}$$

Переведем байты в килобайты:

$$960\,000 : 1024 = 937,5 \text{ Кбайт.}$$

Ответ: Книга займет 937,5 Кбайт информации.

**Знание. Понимание**

1. Как происходит перевод крупных единиц измерения информации в мелкие? Приведите пример.
2. Как происходит перевод мелких единиц в крупные? Приведите пример.
3. Определите информационный объем сообщения «компьютер» в битах.
4. Сколько мегабайт в одном гигабайте?
5. Объясните схему перевода одних единиц измерения в другие (рис. 1.2).

**Применение**

6. Переведите из одной единицы измерения в другую и заполните ячейки табл. 1.5:

Таблица 1.5

1 байт	=		бит
128 бит	=		байт
1 Кбайт	=		бит
1,5 Кбайт	=		байт
2048 байт	=		Кбайт
2 Мбайт	=		Кбайт
81 920 бит	=		Кбайт
1,5 Мбайт	=		байт

7. Каждая страница учебника содержит 40 строк, в каждой строке – 60 символов. Учебник состоит из 176 страниц. Определите информационный объем учебника.

**Анализ**

8. Определите объем информации для хранения имен и телефонов всех своих друзей в смартфоне.

Синтез

9. Сколько останется места на флеш-накопителе емкостью 2 Гбайт, если на него записать книгу, в которой 256 страниц по 95 строк и 80 символов в строке?

Оценивание

10. Размер фотографии 4000 · 1500 точек. Каждая точка имеет столько цветов, что для ее кодирования требуется 2 байта. Сколько места займет данная фотография?

11. Рассчитайте информационный объем предложения (в битах): «Казахстан – многонациональное государство». Каждый символ кодируется двумя байтами в коде **Unicode**.

Феномен информации

Передача данных – уникальное явление, которое дает возможность всему в мире развиваться и приспособливаться. В этом аспекте человек вышел на самый высокий уровень в известном нам мире.

Способы передачи:

- наскальные рисунки (отпечатки рук) (около 40 тысяч лет назад) – самые древние рукотворные знаки;
- шумерский язык (около 3400 лет до н.э.) – самый древний письменный язык;
- человечество придумало 7000 языков. Самый распространенный язык в мире – «ломаный английский» (около 3 млрд носителей);
- сигнал гамма-излучения от вспышки блазара QSO B0218+357. Это самый древний сигнал, «услышанный» человеком (<https://naked-science.ru/article/sci/astrofiziki-nashli-samyy-dalekiy>).

1.3. КОМПЬЮТЕРНАЯ ПАМЯТЬ

ВЫ НАУЧИТЕСЬ

Описывать назначение видов памяти компьютера.

КЛЮЧЕВЫЕ СЛОВА

Компьютерная память	Компьютерлік жад	Computer memory
Оперативная память	Жедел жад	Random Access Memory
Кэш-память	Кэш жады	Cache memory
CMOS-память	CMOS жады	CMOS memory
Постоянная память	Тұрақты жад	Permanent memory
Видеопамять	Бейне жады	Video memory

Современные компьютеры имеют много разнообразных запоминающих устройств. Эти устройства отличаются между собой по назначению, временным характеристикам, объему хранимой информации и др.



Компьютерная память – это устройство хранения информации, или запоминающее устройство. Память является обязательной частью компьютера.

Компьютерную память можно разделить на **внутреннюю** и **внешнюю**.

Внешняя память предназначена для долговременного хранения информации пользователя, а внутренняя используется компьютерной системой.



Оперативная память (*Random Access Memory – RAM*) – это быстрое запоминающее устройство не очень большого объема, связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами (рис. 1.3, 1.4).

После загрузки компьютера в оперативную память загружается информация. Информация хранится до тех пор, пока работает компьютер. После выключения компьютера вся информация из памяти стирается.



Рис. 1.3. Оперативная память для настольного компьютера



Рис. 1.4. Оперативная память для ноутбука

Кэш-память (или **сверхоперативная память**) – очень быстрое запоминающее устройство небольшого объема. Она используется при обмене данными между микропроцессором и оперативной памятью.



Кэш-память бывает двух видов: **первого уровня**, которая располагается на процессоре, и **второго уровня**, которая является отдельной высокоскоростной микросхемой статической памяти (SRAM) (рис. 1.5, 1.6).



Рис. 1.5. Процессор и кэш-память



Рис. 1.6. Кэш-память в виде отдельной высокоскоростной микросхемы

Обмен между данными происходит по следующей схеме: когда процессору требуются данные, он сначала ищет их в кэш-памяти первого уровня. Если там нет данных, обращение идет к кэш-памяти второго уровня. Если процессор не находит данных в кэш-памяти, он обращается к оперативной памяти, поиск в которой занимает намного больше времени, чем в кэш-памяти.

Кэш-память используется для увеличения быстродействия компьютера в целом.

CMOS-память (рис. 1.7) предназначена для длительного хранения данных о конфигурации и настройке компьютера (дата, время, пароль). Для этого используют специальные электронные схемы со средним быстродействием, установленные на материнской плате.



Рис. 1.7. CMOS-память



Постоянная память (ПЗУ, англ. ROM, *Read Only Memory* – память только для чтения) – энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения.

Прежде всего в постоянную память (рис. 1.8) записывают программу управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.



Рис. 1.8. Постоянная память



Видеопамять (VRAM) – это разновидность оперативного запоминающего устройства, в котором хранятся закодированные изображения.

Для хранения графической информации используется **видеопамять** (рис. 1.9).

Это запоминающее устройство организовано так, что его содержимое доступно сразу двум устройствам – процессору и дисплею. Поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.



Рис. 1.9. Видеопамять



Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных. Целостность ее содержимого не зависит от того, включен или выключен компьютер, и не имеет прямой связи с процессором.

Информация от ВЗУ к процессору, и наоборот, циркулирует примерно по следующей цепочке:



К устройствам внешней памяти относятся: накопитель на жестких магнитных дисках (НЖМД или HDD), гибких магнитных дисках (НГМД), магнитных лентах (стримеры), оптических дисках (CD-ROM), магнитооптических дисках, USB-флеш (рис. 1.10).



Рис. 1.10. Накопители

Устройства, относящиеся к внешней памяти компьютера, постоянно модифицируются. Их производительность, объем, размер и внешний вид улучшаются в соответствии с современными требованиями.

Знание. Понимание



1. Объясните, как вы понимаете, что такое «компьютерная память».
2. Для чего необходима внутренняя, внешняя память?
3. Сравните внутреннюю и внешнюю память.
4. Что такое кэш-память?
5. Где хранятся закодированные изображения?
6. Для чего предназначена постоянная память?



Применение

7. Установите соответствие между назначением и видами памяти в табл. 1.6.

Таблица 1.6

№	Назначение памяти	№	Вид памяти
I	Хранит информацию о начальной загрузке и настройках компьютера	A	Оперативное запоминающее устройство
II	Промежуточное запоминающее устройство для обмена информацией между ОЗУ и процессором	B	Внешний жесткий диск
III	Видеопамять	C	Постоянное запоминающее устройство
IV	Хранит информацию в текущий момент времени, пока включен компьютер	D	Внешний лазерный (оптический) диск
V	Хранит большие объемы информации после выключения компьютера	E	Внутренняя кэш-память
VI	Хранит большие объемы информации, служит для переноса ее на другие компьютеры	F	Оперативное запоминающее устройство, в котором хранятся закодированные изображения

I	II	III	IV	V	VI

8. Выполните на компьютере:

- Откройте **Диспетчер устройств**.
- Отметьте устройства, подключенные к компьютеру.
- Вставьте флеш-накопитель.
- Что изменилось в **Диспетчере устройств**?

9. Заполните табл. 1.7 недостающей информацией.

Таблица 1.7

Функция	Человек	Компьютер
Хранение информации		
Обработка информации		
Прием информации		
Передача информации	Речь, двигательная система	

Анализ



10. Составьте кластер по видам компьютерной памяти.

Примечание: кластер – это графическая форма организации информации, когда выделяются основные смысловые единицы, которые фиксируются в виде схемы с обозначением всех связей между ними. Он представляет собой изображение, способствующее систематизации и обобщению учебного материала (рис. 1.11).

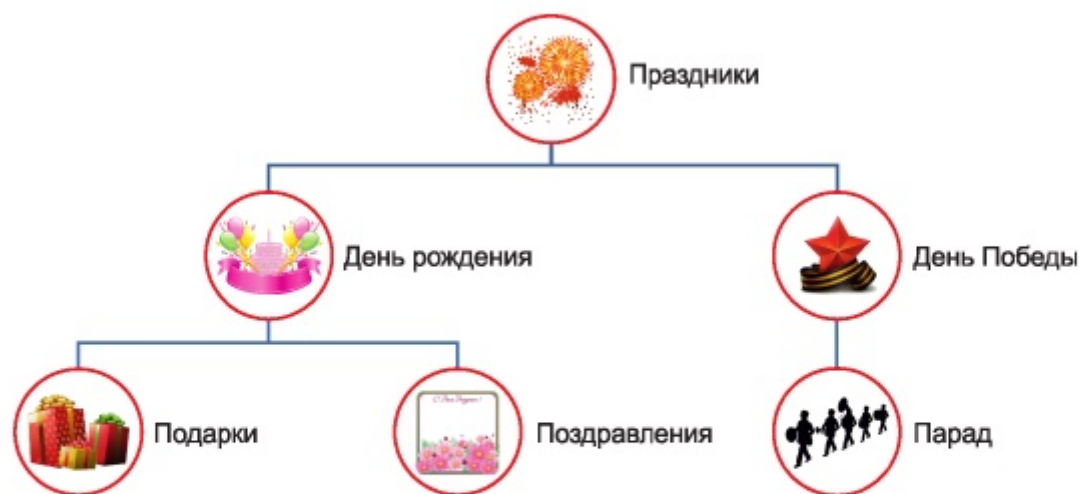


Рис. 1.11. Кластер



Синтез. Оценивание

11. Разгадайте кроссворд (рис. 1.12).

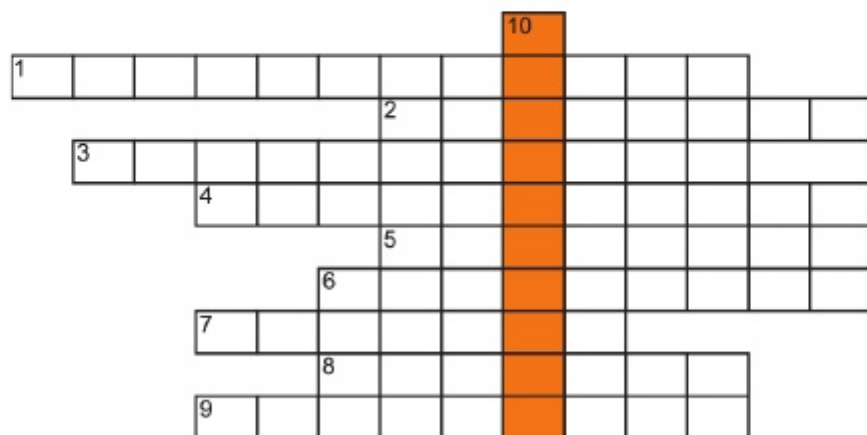


Рис. 1.12. Кроссворд «Память компьютера»

По горизонтали:

1. Внутренняя память, которая хранит информацию временно, пока компьютер включен.
2. Гибкий магнитный диск.
3. Внутренняя память, которая хранит информацию для начальной загрузки компьютера.
4. Вид памяти, которая может хранить большие объемы информации. Принцип чтения/записи информации – луч лазера.
5. Один из магнитных дисков, хранит все программы и данные на вашем компьютере.
6. Устройство для чтения и записи информации на диск.
7. Устройство для хранения информации на компьютере.
8. Один из видов внешней памяти, предназначенный для длительного хранения и переноса на другой компьютер больших объемов информации.
9. Другое название компакт-диска (оптического диска).

По вертикали:

10. Другое название жесткого магнитного диска.



Общий объем информации в организме человека составляет 60 зеттабайт. В одной клетке содержится 1,5 Гб информации. В среднем человек состоит из 40 триллионов клеток.

Эволюция носителей информации показана на рис. 1.13.



Рис. 1.13. Эволюция носителей информации



SSD (англ. *Solid-State Drive*) – твердотельный накопитель, который применяется для хранения информации и представляет собой набор микросхем флеш-памяти, размещенных на одной плате. Кроме микросхем памяти SSD содержит управляющий контроллер. В настоящее время твердотельные накопители используются как в ноутбуках, нетбуках, планшетах, так и в стационарных компьютерах для повышения производительности. По сравнению с традиционными жесткими дисками HDD, твердотельные SSD-накопители имеют меньший размер и вес, не шумят, менее подвержены повреждениям (у SSD нет механических частей), работают значительно быстрее, чем классические жесткие диски, файлы и программы открываются в четыре раза быстрее. Максимальная ёмкость SSD – до 72 терабайт.

1.4. РАЗМЕРЫ ФАЙЛОВ

ВЫ НАУЧИТЕСЬ

Сравнивать файлы разных размеров, хранящих одинаковую информацию.

КЛЮЧЕВЫЕ СЛОВА

Файлы разных размеров	Түрлі өлшемдегі файлдар	Files of different sizes
Текстовый файл	Мәтіндік файл	Text file
Графический файл	Графикалық файл	Graphic file
Видеофайл	Бейне файлы	Video file

Часто при работе с разными файлами с одинаковым содержанием необходимо сравнить их и выбрать тот файл, который будет удовлетворять вашим требованиям. Например, документ, созданный в текстовом процессоре Word 97–2003, имеет расширение **.doc**, а документ, созданный в текстовом процессоре Word 2007 и выше версий, имеет расширение **.docx** (рис. 1.14).



Рис. 1.14. Расширения **.doc** и **.docx**

Сегодня актуальны файлы с именем и расширением **.docx**. Количество файлов с расширением имени **.doc** ввиду постепенного обновления программного обеспечения первых постепенно уменьшается.



Шаг за шагом

Чтобы узнать разницу между двумя текстовыми файлами с расширением **.docx** и **.doc**, проведем мини-исследование. Для этого сравним их, применяя команду **Свойства** из контекстного меню, нажав правой кнопкой мыши на выбранный файл (рис. 1.15).

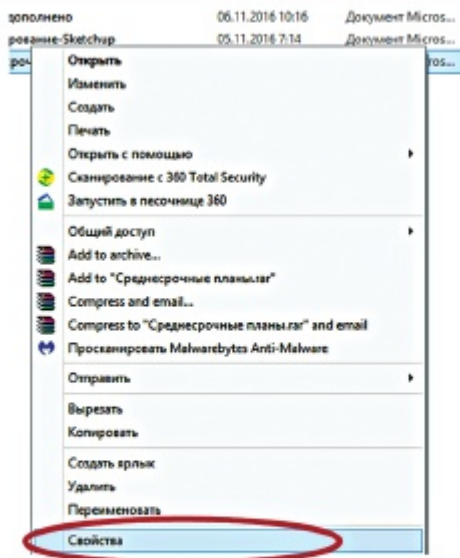


Рис. 1.15. Свойства файла

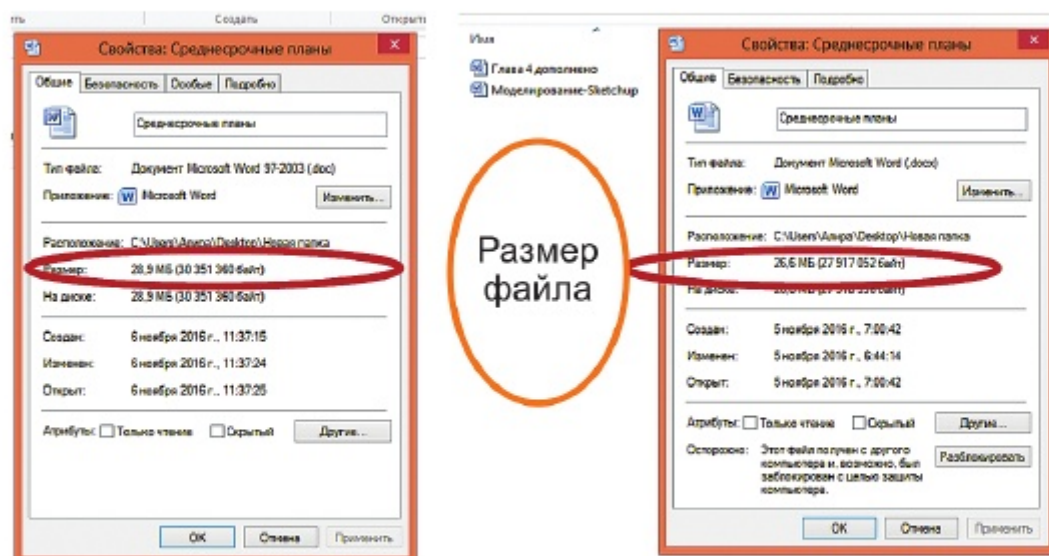


Рис. 1.16. Сравнение текстовых файлов .doc и .docx

Из рис. 1.16 мы видим, что размеры файлов немного отличаются. Файл, созданный с расширением **.doc**, имеет размер 28,9 Мбайт, а документ с расширением **.docx** занимает 26,6 Мбайт. Можно сделать вывод, что документы, имеющие одинаковое содержание, отличаются размером файла. И разница между ними составляет 2,6 Мбайт.

Данное сравнение хорошо иллюстрирует, сколько места на диске занимает одна и та же информация в зависимости от того, в каком формате она записана.

Рассмотрим следующий пример, сравнив два графических файла с одинаковым изображением, но с разным расширением, одинакового содержания, но разных форматов: **TIFF** и **JPEG** (рис. 1.17).

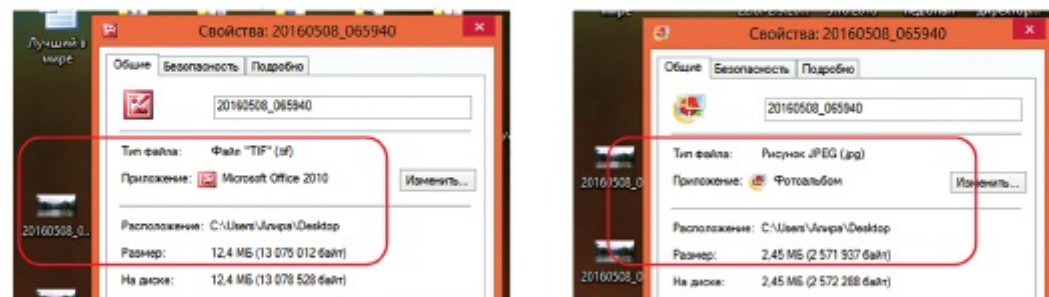


Рис. 1.17. Сравнение графических файлов

Разберем первый файл в формате TIFF. Он часто используется для хранения изображений. Данный формат позволяет сохранять фотографии в различных цветовых пространствах и с большой глубиной цвета (8, 16, 32 и 64 бит). TIFF широко поддерживается графическими приложениями и используется в полиграфии. Изображение в TIFF не будет терять качество после каждого сохранения файла. Но, к сожалению, именно из-за этого TIFF-файлы имеют большой размер. В данном примере он составляет 12,5 Мбайт.

Второй файл с расширением JPG – это самый распространенный и популярный формат графических файлов. На практике сохранение фотографии с минимальной степенью сжатия не дает видимого ухудшения качества изображения. На рис. 1.17 это же изображение имеет размер файла 2,45 Мбайт. Разница при сравнении размеров двух файлов составляет 10,5 Мбайт.

С самыми разными **видеоформатами** мы сталкиваемся постоянно. Сегодня существует множество форматов **видеофайлов**. Существующие видеоформаты обладают уникальными особенностями. Возникают вопросы: «Какой формат видео лучше? Какому видеоформату отдать предпочтение?» Рассмотрим классификацию некоторых форматов видеофайлов, представленную на рис. 1.18.



Формат 3GP, расширение файла .3gp – формат видео из мобильного контента, которое понимается не всеми плеерами и предназначено для мобильных экранов. Качество очень низкое, а размер очень маленький.



Формат MOV, расширение файла .mov – видео, предназначенное для плеера Quick Time от Apple. При выборе файла нужно обращать внимание на другие характеристики: размер, разрешение и т.д.



Формат AVI, расширение файла .avi – самый популярный формат. Он может содержать как видео, так и аудиотекст. Изображение и звук воспроизводятся синхронно.



Формат MPEG-4, расширение файла .mp4 – самый распространенный формат для большинства современных видеоустройств. В этот контейнер входят все популярные видеокодеки. Главное – качество изображений теряется минимально в процессе сжатия.

Рис. 1.18. Классификация некоторых форматов видеофайлов

Существует множество различных **аудиоформатов** (звуковой файл). Наиболее часто используются такие форматы, как **MP3** и **WAV**. Тип формата обычно определяется расширением файла .mp3, .wav, .ogg, .wma.

WAV – один из первых аудиоформатов. Обычно используется для хранения несжатых аудиозаписей. В среднем одна минута звука в формате WAV занимает около 10 Мбайт.

MP3 (MPEG Layer-3) – наиболее распространенный в мире звуковой формат. На текущий момент **MP3** не является лучшим форматом по соотношению размера файла с качеством звучания, но поддерживается большинством устройств.

Для сжатия данных в аудиоформат применяется определенный алгоритм кодирования – **кодек**. Например, в формате **MP3** всегда используется кодек MPEG Layer-3, а в формате **MP4** могут быть использованы разные кодеки.

При сжатии MP3 звук теряет качество, тем самым уменьшается размер файла.

При хранении и отправке файлов разных форматов, но с одинаковым содержанием предпочтение чаще отдается файлу с меньшим размером.

Знание. Понимание



1. Каким образом можно сравнить файлы с одинаковым содержанием, но разных форматов?
2. В чем преимущество расширений текстовых документов **.doc** перед **.docx**, **.txt** перед **.rtf**?
3. Чем отличаются графические файлы с расширениями **.tiff** от **.jpg**, **.bmp** от **.jpg**?
4. Как можно поменять расширение файла для уменьшения его размера?
5. Какие форматы видеофайлов наиболее распространены? Приведите примеры.
6. В чем разница между аудиоформатами **MP3** и **WAV**?

Применение



7. Создайте два текстовых документа: в **Word** с расширением **.doc** и **Блокноте** с расширением **.txt**.
 - Сохраните файлы, назвав их одинаково: **Алгоритм сравнения**.
 - Сравните два файла, открыв документы, созданные в текстовом процессоре **Word** и в **Блокноте**.



Анализ

8. В пустые ячейки табл. 1.8 напротив форматов (расширений) информации впишите их форму представления: текстовая, звуковая, графическая, видеоинформация или аудиоинформация.

Таблица 1.8

										
1	2	3	4	5	6	7	8	9	10	11



Синтез

9. Проведите исследование: какие текстовые, графические, аудио- и видеоформаты вы используете чаще всего?

Какие звуковые и видеоформаты поддерживает программное обеспечение вашего телефона?



Оценивание

10. Художник нарисовал одну тысячу картин для выставки. Одна картина с расширением .jpg имеет приблизительный размер 5 Мбайт. Сколько потребуется места на жестком диске для хранения всех картин?

11. Почему компьютерную графику почти всегда используют в сжатом виде?



Феномен информации

Воспользуемся самым простым графическим редактором **Paint** (рис. 1.19).

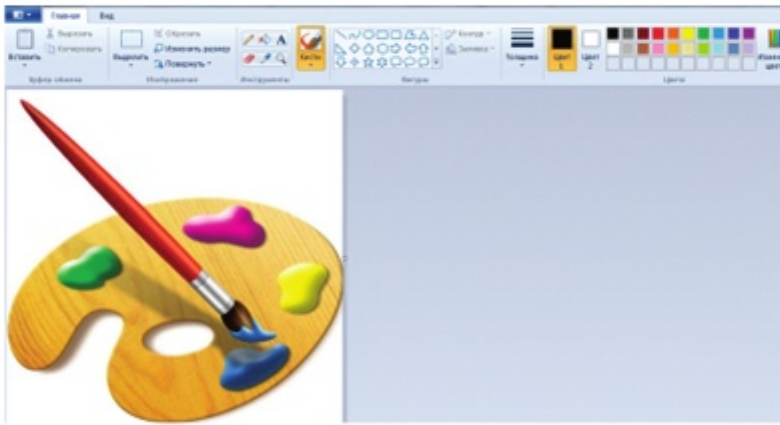


Рис. 1.19. Окно графического редактора **Paint**

Уменьшим размер изображения, выполнив следующие действия:

1. Откройте изображение в **Paint**.
2. Нажмите на кнопку **Изменить размер**, откроется окно (рис. 1.20).

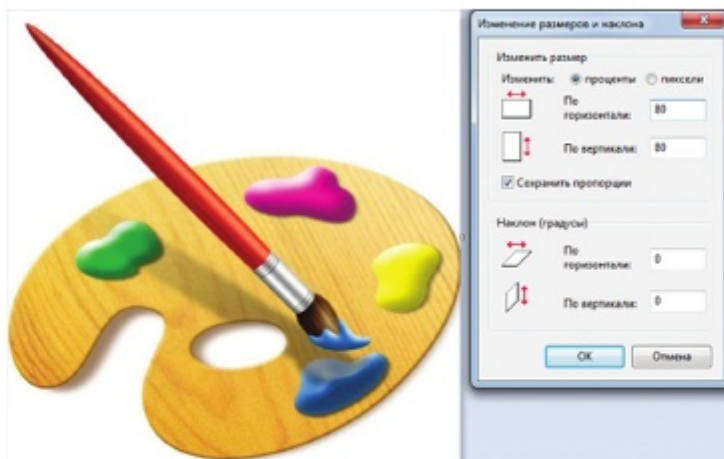


Рис. 1.20. Окно **Изменение размеров и наклона**

3. В поле **Изменить:** активируйте «проценты», установите галочку **Сохранить пропорции**, введите число для уменьшения изображения от исходного, например, 50%.

4. Нажмите на **ОК** (рис. 1.21).

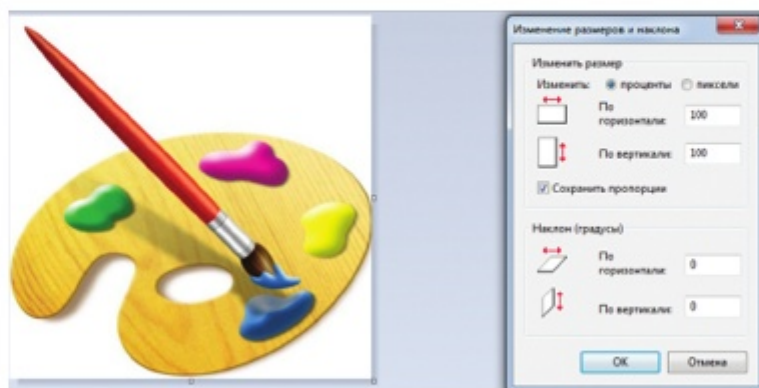


Рис. 1.21. Сохранение изменения размеров

5. Сохраните результат, нажав на **Файл – Сохранить как**.

6. Выберите расширение и введите имя файла.

1.5.

СОЗДАНИЕ И РАСПАКОВКА АРХИВОВ РАЗЛИЧНЫХ ФОРМАТОВ

ВЫ НАУЧИТЕСЬ

Создавать и распаковывать архивы различных форматов.

КЛЮЧЕВЫЕ СЛОВА

Архивные файлы	Сығылған файлдар	Archive files
Распаковка файлов	Файлдарды ашу	Unzipping files
Сжатие	Қысу	Compression
Программы-архиваторы	Мұрағатшы бағдарламалар	Archiver programs



Проблема сжатия данных существует давно, с момента появления компьютеров. **Сжатием** (уменьшением размера) данных называется преобразование данных с целью уменьшения занимаемого ими объема. Сжатие данных используется при работе с файлами больших форматов. Степень сжатия в основном зависит от исходных размеров файлов. Хорошо сжимаются почти все предварительно несжатые данные, например, исполняемые файлы, тексты, простые несжатые изображения. В настоящее время сжимать данные необходимо в двух случаях:

- 1) для длительного хранения данных на различных носителях информации;
- 2) для передачи данных по каналам связи.

При сжатии данных получаются файлы особого типа, называемые **архивами**.

Архив – это файл, содержащий в себе один или несколько файлов в сжатом виде.



Расширения файлов, являющихся архивами, могут быть **zip, arj, rar** и др. Тип расширения зависит от программы-архиватора, в которой они были созданы.

Программы-архиваторы – это специальные программы, с помощью которых можно сжимать отдельные файлы или группы файлов.

Процесс сжатия файлов называется **созданием архива**. Процесс восстановления сжатых файлов – **распаковкой архива**.



С помощью программ-архиваторов **WinRar, WinZip, WinArj** можно не только сжимать файлы, но и просматривать содержание архивов, распаковывать, устанавливать защиту.

Наиболее популярным архиватором является программа WinRar, которая имеет три главных преимущества:

- поддержка разных платформ операционной системы;
- работа с разными типами архивов, в том числе запакованными другими архиваторами;
- высокая степень сжатия архива.



Шаг за шагом

Существуют два способа создания архива (архивации) данных в программе архиватора WinRar: использование окна программы или контекстного меню. Используем простейший способ создания (архивации) файлов через контекстное меню (рис. 1.22).

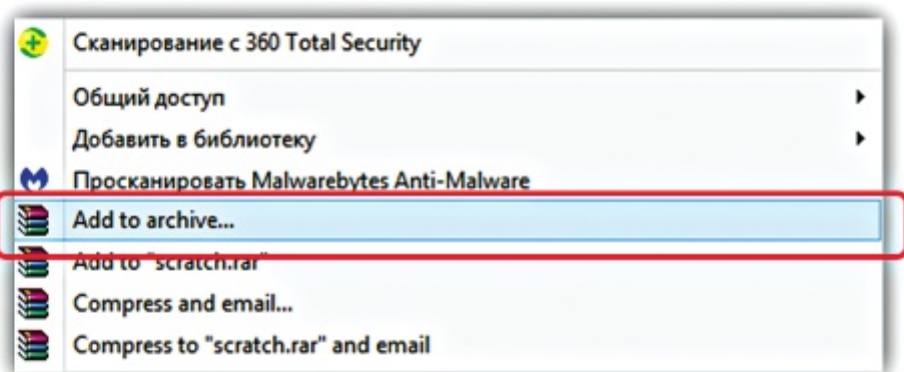


Рис. 1.22. Добавить в архив

Создадим архив по алгоритму:

1. Выберите файл (для выделения группы файлов нажмите клавишу **Ctrl** и щелкайте левой кнопкой мыши – ЛКМ) или папку.
2. Нажмите правой кнопкой мыши (ПКМ) на выделенные файлы, появится контекстное меню (или нажмите на клавиши **Shift + F10**).
3. Выберите в контекстном меню команду **Add to archive... (Добавить в архив...)**.
4. В диалоговом окне **Archive name and parameters (Имя и параметры архива)** нажмите на кнопку **Browse... (Обзор...)** (рис. 1.23).
5. В поле **Archive name (Имя архива)** задайте имя архива или оставьте то имя, которое будет отображено в поле по умолчанию (по названию файла или папки).

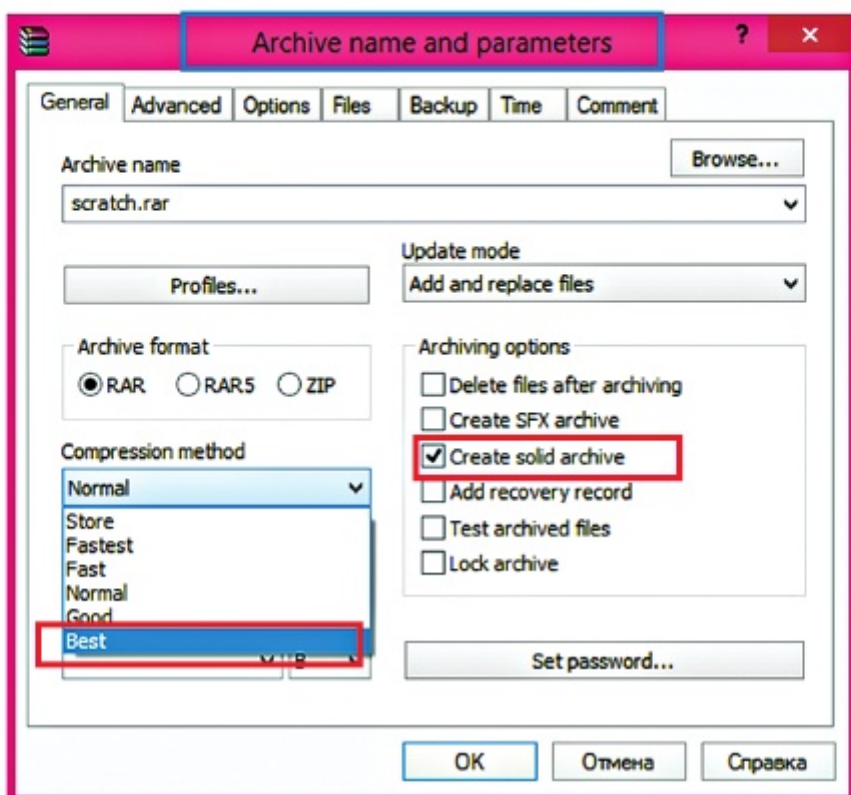


Рис. 1.23. Имя и параметры архива

6. Выберите параметр архивации **Create solid archive (Создать непрерывный архив)** из группы **Archiving options (Параметры архивации)** (см. рис. 1.23).

Выберите из списка **Compression method (Метод сжатия)** **Best – Лучший**.

7. Нажмите кнопку **OK**.

8. Будет создан один непрерывный файл с расширением **.rar**.

9. Попробуйте самостоятельно распаковать на любом компьютере файлы из архива (рис. 1.24).

10. Обсудите алгоритм распаковки архива.

11. Создайте самораспаковывающийся архив. В чем преимущество такого архива?

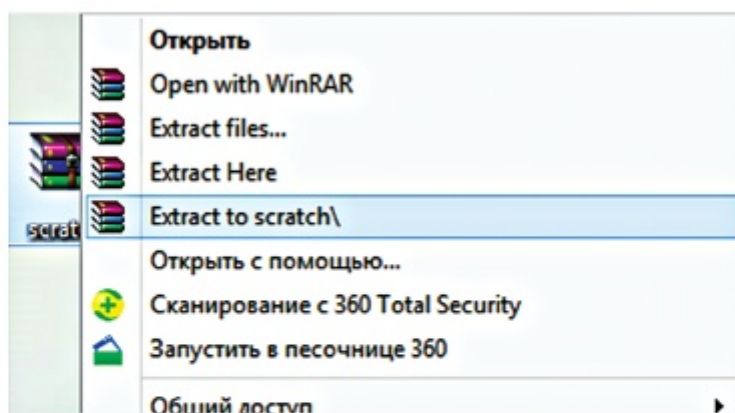


Рис. 1.24. Распаковка архива



Знание. Понимание

1. Объясните своими словами значение терминов «создание архива» и «распаковка».
2. Какие программы-архиваторы вы еще знаете?
3. Как вы думаете, всегда ли необходимо создавать архив?
4. В чем преимущество архива?
5. Какие файлы будут максимально сжаты: текстовые или графические?



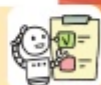
Применение

6. Создайте папку **Заготовки** с текстовыми, графическими, звуковыми и видеофайлами.
7. Создайте рабочую папку под именем **Архивы**.
8. В папке **Архивы** создайте четыре подпапки с именами: **Документы**, **Графика**, **Музыка** и **Видео**.
9. В текстовом процессоре **Word** создайте таблицу.
10. Впишите в нее названия файлов и их размер из папки **Заготовки** (табл. 1.9).

Таблица 1.9

Текстовые файлы		Графические файлы		Звуковые файлы		Видеофайлы	
Название	Размер	Название	Размер	Название	Размер	Название	Размер
Итого		Итого		Итого		Итого	
Документы.rar		Графика.rar		Музыка.rar		Видео.rar	
Размер		Размер		Размер		Размер	

Анализ



- С помощью архиватора **WinRar** создайте архивы соответствующих файлов из папки **Заготовки** в архивы с именами: **Документы.rar**, **Графика.rar**, **Музыка.rar**, **Видео.rar**. Создаваемые архивы поместите соответственно в папки **Документы**, **Графика**, **Музыка** и **Видео**.
- Выпишите размер созданных архивов в таблицу.
- Сделайте вывод: какие файлы лучше всего сжимаются.
- Распакуйте архивы **Документы.rar**, **Графика.rar**, **Музыка.rar**, **Видео.rar** в папки **Документы**, **Графика**, **Музыка**, **Видео**.
- Создайте в рабочей папке самораспаковывающийся архив (SFX-архив) с именем **Самораспаковка.exe**, в который запакуйте все файлы из папки **Заготовки**.
- Извлеките архив **Самораспаковка.exe**, дважды нажав левой кнопкой мыши.

Синтез. Оценивание



- Проверка архивов на вирусы:
 - Откройте программу **WinRar**.
 - С помощью комбинации клавиш **Alt + D** выполните проверку архивных файлов на вирусы.
- Создайте многотомный архив, выбрав размер тома из выпадающего списка, связанного с этим полем ввода. Создайте архив файлов, разделенных на тома, емкостью 700 Кбайт.
- Создайте папки **Большие файлы** и **Маленькие файлы**, соответственно записав туда файлы большого и маленького размеров.

20. Выполните архивацию файлов из папки, используя разные способы архивации:

- с помощью контекстного меню;
- с помощью кнопки **Добавить (Add...)**;
- создайте архив с паролем (**Ctrl + P**).

21. Убедитесь, что уже сжатые файлы в формате **.mp3** и **.jpg** совсем или почти не сжимаются.

22. Заполните таблицу «Архивация файлов» (табл. 1.10) в текстовом процессоре **Word**. Определите, во сколько раз уменьшился размер файла.

Таблица 1.10

Архивация файлов

Имя файла	Тип файла	Истинный размер файла	Размер файла после создания архива	Во сколько раз уменьшился размер файла

23. Убедитесь, что размер файла после создания архива почти равен исходному файлу.



Архив данных – это копия данных компьютера, хранящаяся, как правило, на внешних носителях информации. С помощью внешнего носителя с архивом данных можно впоследствии восстановить данные в том виде, в котором они были на момент архивации.

Архиваторы

Кроме WinRAR существуют и другие виды архиваторов: WinZip, WinAce, 7-Zip, IZArc и PowerArchiver. У каждого из них свой набор поддерживаемых форматов, свои плюсы и минусы. При выборе архиватора необходимо оценить его функциональность, то есть базовый набор возможностей:

- поддержка архиваторами архивных форматов сжатия: zip, zipx, rar, ace и 7z;
- создание, помимо обычных архивов, многотомных, самораспаковывающихся в формате exe (Self-Extracting – SFX) и непрерывных (Solid) архивов, а также архивирование больших файлов (свыше 4 Гбайт);
- наличие функционала для восстановления поврежденных архивов и шифрования данных;
- реализация поддержки drag-and-drop (с англ. «бери-и-брось»);
- пакетная распаковка архивов и приостановка процесса архивирования/распаковки при обработке больших объемов данных.

ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ I

Из предложенных альтернативных ответов выберите только один верный. Каждый правильный ответ оценивается в 1 балл.

1. Перевод текста с английского языка на русский является процессом:

- а) хранения информации;
- б) передачи информации;
- в) поиска информации;
- г) обработки информации;
- д) ни одним из перечисленных выше процессов.

2. За единицу измерения информации принимается:

- а) 1 бод;
- б) 1 бар;
- в) 1 бит;
- г) 1 кг;
- д) 1 фут.

3. Как 1 бит информации изменяет неопределенность?

- а) Не изменяет.
- б) Уменьшает в два раза.
- в) Увеличивает в два раза.
- г) Уменьшает в восемь раз.

4. Что такое 1 байт?

- а) 1024 Кбайт.
- б) 10 Мбайт.
- в) 8 бит.
- г) 4 бит.

5. Чему равен 1 Гбайт?

- а) 1024 Кбайт.
- б) 1000 Кбайт.
- в) 1024 Мбайт.
- г) 1000 Мбайт.

6. Чему равны 4 килобайта?

- а) 5120 байт.
- б) 4000 байт.
- в) 4096 байт.
- г) 4000 бит.

7. Программой-архиватором называют:

- а) программу для уменьшения информационного объема (сжатия) файлов;
- б) программу резервного копирования файлов;
- в) интерпретатор;
- г) транслятор;
- д) систему управления базами данных.

8. Архивный файл представляет собой:

- а) файл, которым долго не пользовались;
- б) файл, защищенный от копирования;
- в) файл, сжатый с помощью архиватора;
- г) файл, защищенный от несанкционированного доступа;
- д) файл, зараженный компьютерным вирусом.

9. Какое из названных действий можно произвести с архивным файлом?

- а) Переформатировать.
- б) Распаковать.
- в) Просмотреть.
- г) Запустить на выполнение.
- д) Отредактировать.

10. Степень сжатия файла зависит:

- а) только от типа файла;
- б) только от программы-архиватора;
- в) от типа файла и программы-архиватора;
- г) от производительности компьютера;
- д) от объема оперативной памяти персонального компьютера, на котором производится архивация файла.

11. Компьютерная память – это:

- а) устройство хранения информации, или запоминающее устройство;
- б) процессор;
- в) клавиатура;
- г) внешние устройства компьютера;
- д) микропроцессор.

2.1. КОМПЬЮТЕРНЫЕ СЕТИ И ИХ КЛАССИФИКАЦИЯ

ВЫ НАУЧИТЕСЬ

Классифицировать компьютерные сети.

КЛЮЧЕВЫЕ СЛОВА

Компьютерные сети	Компьютерлік желілер	Computer networks
Сервер	Сервер	Server
Топология	Топология	Topology
Клиент	Клиент	Client
Абонент	Абонент	Subscriber

Компьютеры могут быть связаны в сети для коллективного использования данных и ресурсов. Сеть может быть очень простой и представлять собой два компьютера, соединенных одним кабелем (рис. 2.1). В состав сложной сети входят сотни компьютеров.



Рис. 2.1. Схема сети из двух компьютеров

Компьютерные сети – это объединение компьютеров для обмена информацией и совместного использования ресурсов.



Компьютерные сети разделены:

- по масштабу;
- по типу организации работы компьютеров в сети;
- по топологии;
- по типу среды передачи.



По масштабу компьютерные сети бывают: локальные, региональные, глобальные (рис. 2.2). Какая территория будет охвачена в составе перечисленных сетей?



Рис. 2.2. Масштаб компьютерных сетей

Существуют различные способы объединения компьютеров в сеть. Основными типами компьютерных сетей являются **одноранговая сеть** и **сеть на основе сервера** (рис. 2.3).

Сервер – абонент сети, отдающий в сеть свой ресурс и имеющий или не имеющий доступ к ресурсам сети. Он используется для хранения файлов, веб-страниц, для управления почтой, для хранения копий файлов.

Виды серверов: **файл-серверы** и **принт-серверы**, **серверы приложений**, **почтовые серверы**, **факс-серверы**, **коммуникационные серверы**, **сервер служб каталогов**.

В одноранговой сети нет выделенных серверов, функции управления сетью передаются по очереди от одного компьютера к другому.

В одноранговой сети все компьютеры равноправны. Каждый из них имеет уникальное имя компьютера и обычно пароль для входа в него во время загрузки операционной системы.

Компьютер-сервер – это центральный компьютер, который распределяет общие ресурсы между многими пользователями. Компьютер, который постоянно не включен в сеть, а подключается периодически, называется **клиентом**, или **абонентом**.

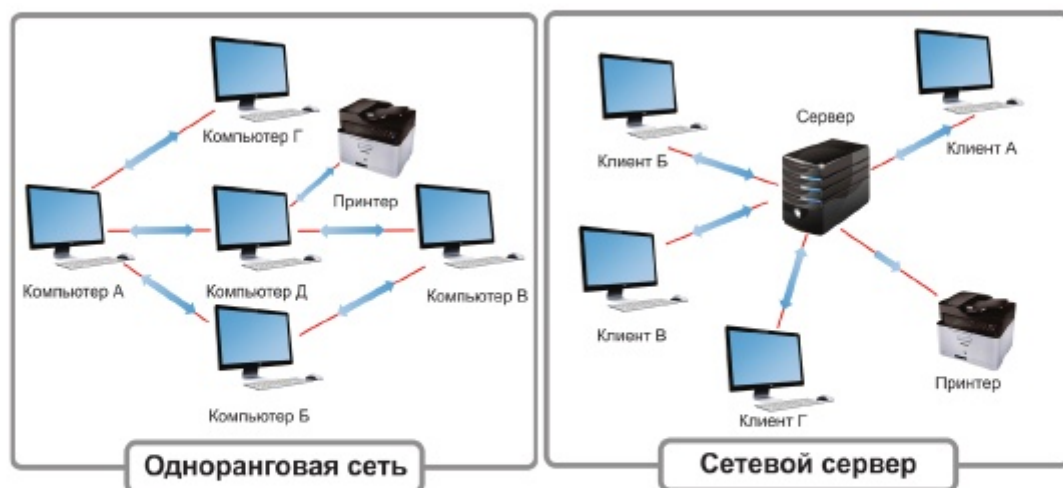


Рис. 2.3. Типы организации работы компьютеров в сети

Для обеспечения возможности подключения к сети можно использовать различные устройства (рис. 2.4): компьютеры, сотовые телефоны, концентраторы, коммутаторы, маршрутизаторы, точки беспроводного доступа.



Рис. 2.4. Концентратор, коммутатор, маршрутизатор

Топология характеризует физическое расположение компьютеров, кабелей и других компонентов сети (рис. 2.5, а). **Физическая топология** определяет способ подключения компьютеров, принтеров и других устройств к сети.

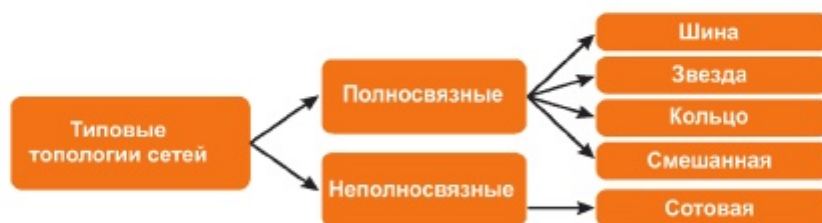


Рис. 2.5, а. Типовые топологии сетей

Наиболее распространенные физические топологии локальных вычислительных сетей (рис. 2.5, б): **шинная, кольцевая, звездообразная, ячеистая, иерархическая звезда.**

Проанализируйте информацию по образцу: «В звездообразной топологии имеется центральная точка подключения, в качестве которой обычно выступает такое устройство, как концентратор, коммутатор или маршрутизатор».

По какому принципу построены топологии остальных компьютерных сетей, представленных на рис. 2.5, б? Согласны ли вы, что наиболее распространенной является шинная топология сети?

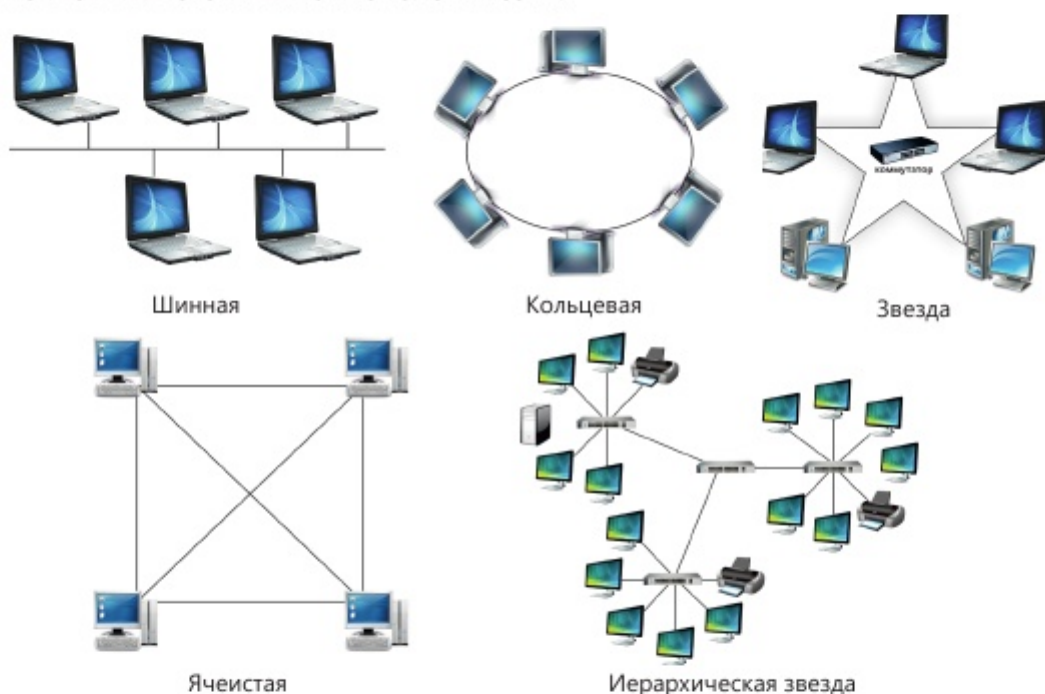


Рис. 2.5, б. Топология сетей

По типу среды передачи сети делятся на:

- **проводные** (витая пара, коаксиальный кабель, оптоволокно);
- **беспроводные** (радиосвязь (*Wi-Fi*, *WiMAX*), инфракрасная связь, СВЧ-связь (*Bluetooth*), *Li-Fi* – передача с помощью света).

Беспроводными локальными сетями называются сети для передачи и приема данных с помощью радиоволн. Область действия сети может ограничиваться лишь одной комнатой, но может охватывать и большее пространство.

Беспроводное подключение очень редко используется в стационарных компьютерах. Это идеальное решение для ноутбуков, «умных» телефонов и других подобных устройств.

Спутниковое подключение имеет самый широкий радиус подключения, но используется только в стационарных компьютерах (рис. 2.6).

В настоящее время мобильные телефоны превратились практически в полноценный компьютер, позволяющий выходить в Интернет, отправлять почту, смотреть онлайн передачи и многое другое. Но некоторые функции не могут работать без качественной связи. Именно по этой причине сейчас мы можем наблюдать быстрый рост и прогресс в области **беспроводных технологий связи**.

Тенденция увеличения передаваемого объема информации в системах сотовой связи создала предпосылки постоянной их эволюции. На рисунке 2.7 представлена эволюция сетей мобильной связи, разделенных на пять основных поколений (G – от англ. *Generation* – «поколение»).



Рис. 2.6. Спутниковое подключение



1-е поколение (Стандарт 1G)

- 1) Сети 1G появились в 1980-х годах.
- 2) Предназначались для беспроводных телефонных технологий



2-е поколение (Стандарт 2G)

- 1) Сеть со стандартом 2G появилась в 1991 году.
- 2) Было доступно шифрование цифрового сигнала, передача голоса, текста, изображения



3-е поколение (Стандарт 3G)

- 1) Стандарт 3G появился приблизительно в 2000 году.
- 2) Появился доступ в Интернет, стал пакетным и более быстрым



4-е поколение (Стандарт 4G)

- 1) 4-е поколение появилось приблизительно в 2010 году.
- 2) Стал доступен мобильный Интернет до гигабайтных скоростей, мобильные игры, онлайн-видео, видеосвязь



5-е поколение (Стандарт 5G)

- 1) Стандарт 5G доступен с 2019 года.
- 2) Отличается высокой скоростью Интернета, подключением большого количества телефонов

Рис. 2.7. Поколения сотовых сетей мобильной связи

5G – это новое поколение мобильной связи. По сравнению с 4G, поколение 5G имеет ряд преимуществ:

- высокая скорость передачи данных;
- низкая задержка сигнала;
- подключение большего числа портативных устройств (телефон, планшет или нетбук и т.д.);
- высокая энергоэффективность;
- высокая пропускная способность;
- высокая мобильность пользователей;
- масштабная виртуализация.

В технологии 5G многие функции реализованы не на уровне физической инфраструктуры, а программным способом. Основными параметрами **4G** и **5G** являются:

- скорость загрузки для пользователей (4G – 10 Мбит/с, 5G – 100 Мбит/с);
- плотность подключения (4G – 100 тыс. устройств/км², 5G – 1 млн устройств/км²).

Перечислим основные сферы применения 5G.

Интернет вещей (IoT) – это концепция «умного» дома, производственные и вычислительные мощности, инфраструктура «умного» города. Устройства и системы объединяются в общую сеть с дистанционным управлением. Например, бытовые приборы, климат-контроль, системы экстренного оповещения, быстрая и качественная связь между станками, измерительными приборами и т.д. Жители «умных» городов смогут пользоваться оперативным доступом к различным сервисам: центрам государственных услуг, городскому транспорту и др.

Беспилотный транспорт – это автономные грузоперевозки, городское такси, сельскохозяйственная техника. Транспорт может быть переведен на беспилотный режим для обеспечения большей точности, надежности и безопасности процессов.

Облачные технологии – хранение данных, моментальная загрузка и вычисления, требующие высокой аппаратной мощности только с помощью мобильного Интернета.

Здравоохранение – качественная связь с удаленными регионами позволит оказывать поддержку в случае экстренных ситуаций, во время сложных операций или диагностики с использованием видеопотока 5G.

Виртуальная и дополненная реальность (VR/AR) – системы интерактивного обучения, навигационные системы, сложные инженерные процессы и даже тактильный Интернет.

С распространением 5G будут активно развиваться «умные» системы, такие как дом, город, производство, использоваться голосовые помощники. В мире уже есть примеры внедрения городских систем видеонаблюдения с интеллектуальным распознаванием лиц. От камер видеонаблюдения требуется лишь передать на сервер видео необходимого разрешения для распознавания лица. Искусственный интеллект станет доступен большому числу пользователей, обеспечивая безопасность зданий, быстрый поиск людей, передачу биометрических параметров для хранения и обработки и т.д.

Знание. Понимание



1. Как классифицируются сети по своим масштабам?
2. В чем отличие одноранговой сети от сети с выделенным сервером?
3. Какие топологии сетей вы знаете?
4. Где применяется беспроводное подключение, спутниковое подключение?
5. Какое подключение имеют сотовые телефоны?
6. С чем связана эволюция стандартов сотовой системы?
7. Почему поколения сотовой связи называют «G»?
8. Какое подключение применяется в сотовой системе 5G?

Применение



9. Необходимо объединить компьютеры в многоквартирном доме с несколькими подъездами. Какую топологию сети выберете? Какой тип кабеля? С какими средствами передачи? Обоснуйте свой выбор.
10. Какой тип сети вы бы выбрали для объединения трех смартфонов среди друзей? Обоснуйте свой выбор.
11. Составьте кластер эволюции стандартов сотовой системы.
12. Приведите примеры сфер применения 5G.
13. Каков ваш прогноз по появлению шестого поколения – сотовой системы 6G?

Анализ



14. Перечислите наиболее распространенные физические топологии локальных вычислительных сетей. Придумайте различные способы соединения 10 компьютеров в компьютерном классе в сеть.

15. Нарисуйте схемы топологии сетей.

16. Найдите способ, обеспечивающий самый короткий маршрут передачи информации между любыми двумя абонентами.

17. Рассмотрите достоинства и недостатки таких объединений. Заполните табл. 2.1.

Таблица 2.1

Критерии оценки сети	Шинная	Кольцевая	Звезда	Иерархическая	Ячеистая
Экономические затраты на кабель					
Возможность нелегального подключения					
Возможность подключения абонента без остановки работы сети					
Возможность обмена информацией без сервера					
Влияет ли поломка компьютера абонента на работу сети?					



Синтез

18. Определите топологию сетей, изображенных на рис. 2.8.

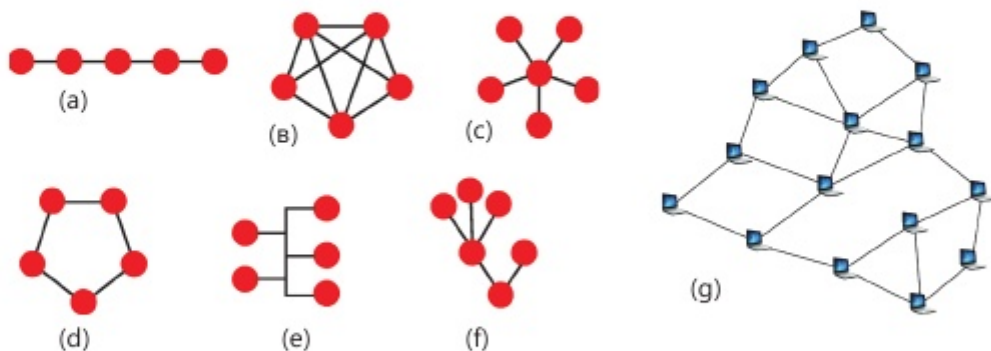


Рис. 2.8. Топология разнообразных сетей

Оценивание

19. Проведите мини-исследование. Оцените, какая топология сетей в классе, дома, в крупных супермаркетах, аэропорту.

Интернет-технологии в современном мире

Современные интернет-технологии изменили жизнь общества, образование, медицину, бизнес, чтение книг, просмотр видео, запуск игр и др. Сегодня без мобильной связи, гаджетов (англ. *gadget* – специализированное устройство или приложение небольшого размера), компьютеров и Интернета невозможно обучение в школе, вузе, ведение бизнеса, получение информации, общение в социальных сетях.

Из поколения в поколение портативные устройства, такие как планшеты и смартфоны, становятся всё более функциональными. Они оснащены мощными процессорами, которые заметно расширяют функциональность устройств. Им доступно большое разнообразие сервисов и приложений. В настоящее время с помощью планшета или смартфона можно сделать качественную фотосъемку, снять фильм, отредактировать его и предоставить для просмотра пользователям Интернета.

Современный человек ежедневно пользуется интернет-технологиями и услугами, которые помогают ему в самых различных сферах. Мы видим, как виртуальная реальность тесно переплетается с материальной, помогает пользователю при ежедневном получении, обработке, отправке всевозможных данных, создавая архивы, управляя целыми производственными процессами.

Приведем топ-10 технологий в исследованиях Gartner, которые стали популярными в цифровом мире:

1. Интернет вещей (IoT);
2. Большие данные и предиктивная аналитика;
3. Искусственный интеллект;
4. Цифровые двойники;
5. Периферийные вычисления;
6. Виртуальная реальность (VR, AR);
7. Блокчейн;
8. Смарт-пространство;

9. Этика и приватность в цифровом мире;

10. Квантовая вычислительная техника.

Интеграция реальных вещей в сеть Интернет развивается большими темпами, так как использование удаленного доступа разными вещами с помощью Интернета становится все более востребованным. Первый образец Интернет вещей (*IoT, Internet of Things*) был создан в 1990 году Джоном Ромки.

Активное проникновение сетевых технологий и Интернета привели к изменениям не только в бизнесе, но и в сфере образования, превратившись в глобальный и ключевой инструмент коммуникации. Ежегодно увеличивается число людей, получающих образование в дистанционном формате.

В результате инноваций и развития новых технологий на замену профессиям, которые могут быть легко автоматизированы, приходят новые профессии (рис. 2.9).



Рис. 2.9. Примеры новых профессий

2.2. АНТИВИРУСНАЯ БЕЗОПАСНОСТЬ

ВЫ НАУЧИТЕСЬ

Защищать компьютер от вредоносных программ.

КЛЮЧЕВЫЕ СЛОВА

Антивирусная программа	Вирустан қорғау программасы	Antivirus program
Вредоносная программа	Зиянды бағдарлама	Malware program
Вирус	Вирус	Virus



Человек как живой организм подвергается воздействию различных вирусов. А может ли «заболеть», «заразиться» вирусом компьютер? Как это происходит, как этого избежать и как «вылечить» компьютер? Причиной заражения компьютера является вирус, только компьютерный.

Вирус – это программа, которая активизируется, копируя саму себя в исполняемые объекты. Вирусы могут проникнуть в ваш компьютер из других зараженных компьютеров, через носители данных или через сеть (локальную или Интернет).

Вредоносная программа (*Malware* – сокращение от «malicious software») – любое программное обеспечение, специально созданное для того, чтобы причинять ущерб компьютеру, серверу или компьютерной сети и хранящимся на них данным.



К вредоносным программам относятся:

- 1) **черви** – это программы, которые распространяют свои копии по сети;
- 2) **программы-шпионы** устанавливаются скрытным путем на компьютеры для сбора информации о пользователе без его ведома;
- 3) **троянские программы** – это вредоносные программы, выполняющие несанкционированные пользователем действия. Такие действия могут включать:
 - удаление данных;
 - блокирование данных;
 - изменение данных;
 - копирование данных;
 - замедление работы компьютеров и компьютерных сетей.

Познакомьтесь с некоторыми типами вирусов и их описанием.

1. Загрузочные вирусы атакуют загрузочные секторы (гибкого или жесткого диска) и устанавливают свои подпрограммы, загружаемые при запуске компьютера.

2. Файловые вирусы атакуют исполняемые программы, в частности, все файлы с расширениями **exe** и **com**.

3. Вирусы скриптов являются разновидностью файловых вирусов. Вирусы скриптов способны заразить файлы других форматов, позволяющих выполнение скриптов, например **html**.

4. Макровирусы атакуют документы, в которые могут вставляться макроканды (макросы). Эти вирусы часто внедряются в приложения обработки текстов или электронных таблиц, так как в эти типы файлов легко вставляются макросы.

5. Сетевые вирусы распространяются по компьютерной сети. Именно они представляют главную угрозу для пользователей глобальной сети. К ним относятся: интернет-черви, троянские программы (*Trojan* и *Backdoor*).



Признаки, которые показывают, что компьютер может быть заражен вирусом:

- прекращение работы или неправильная работа ранее успешно функционировавших программ;
- медленная работа компьютера;
- невозможность загрузки операционной системы;
- исчезновение файлов и каталогов или искажение их содержимого;
- изменение даты и времени модификации файлов;
- изменение размеров файлов;
- неожиданное значительное увеличение количества файлов на диске;
- существенное уменьшение размера свободной оперативной памяти;
- вывод на экран непредусмотренных сообщений или изображений;
- подача непредусмотренных звуковых сигналов;
- частые зависания и сбои в работе компьютера.



Проанализируйте

На рис. 2.10 приведен алгоритм работы вирусов на компьютере. Проанализировав схему, объясните, каким образом будет осуществляться проникновение вируса на компьютер.

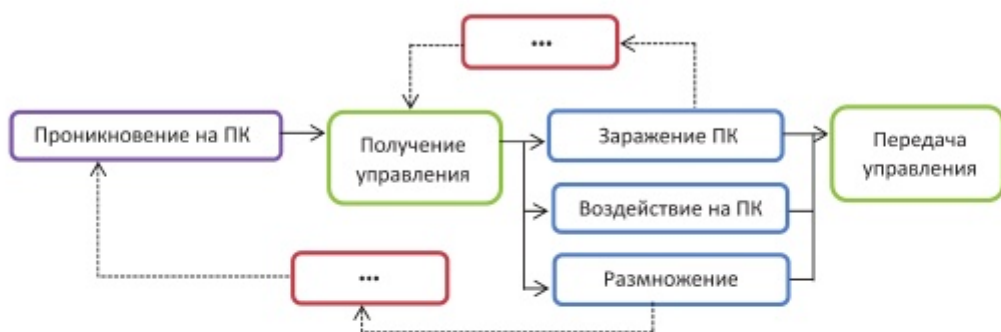


Рис. 2.10. Алгоритм работы вирусов на компьютере

Различают следующие виды антивирусных программ (рис. 2.11):



Рис. 2.11. Виды антивирусных программ

Программы-детекторы осуществляют поиск конкретного вируса в оперативной памяти и в файлах. Недостатком программ-детекторов является то, что они могут находить только те вирусы, которые известны разработчикам таких программ.

Программы-доктора и **программы-вакцины** находят зараженные вирусами файлы и «лечат» их. При «лечении» они удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние.

Программы-ревизоры относятся к самым надежным средствам защиты от вирусов. Ревизоры запоминают исходное состояние программ, каталогов и системных областей диска компьютера, не зараженного вирусом, а затем по желанию пользователя сравнивают текущее состояние с исходным.

Программы-фильтры представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов: попытки коррекции

файлов с расширениями **com** и **exe**, загрузка резидентной программы и др. Программы-фильтры способны только обнаруживать вирус на самой ранней стадии его существования до размножения. Для уничтожения вирусов требуется применение других программ.

К профилактическим мерам защиты компьютера от вируса относятся:

- регулярная проверка компьютера с помощью антивирусных программ;
- контроль информации, поступающей из Интернета, через электронную почту, съемные носители;
- общие средства защиты информации – своевременное создание резервных копий данных, использование лицензионного программного обеспечения.

Применение совместно всех видов защиты информации позволит достигнуть высокой безопасности.

На сегодняшний день существуют различные антивирусные программы. Наиболее известные из них **Kaspersky Internet Security, Bitdefender Internet Security, 360 Total Security, Avira Internet Security Suite, ESET Smart Security, Avast, Trend Micro, F-Secure, Panda, AVG** и др.

Антивирусные программы быстро устаревают в связи с появлением новых вирусов, поэтому необходимо регулярное обновление их версий.

Рассмотрим подробнее работу антивирусной программы **360 Total Security**. Программа основана на пяти модулях: эффективных антивирусных ядрах **Avira** и **Bitdefender**, «облачной» системе анализа **360 Cloud**, проактивной технологии выявления угроз **QVMII** и восстанавливающем наборе инструментов **System Repair**.

Антивирусная программа **360 Total Security** проверяет все подозрительные файлы в реальном времени, может анализировать объекты в защищенной среде и содержит ежедневно пополняемую библиотеку антивирусных сигнатур.



Проанализируйте

Проанализируйте известные антивирусные программы. Найдите в Интернете бесплатную версию антивирусной программы **360 Total Security**. Установите ее на ваш компьютер. Проведите с ее помощью комплексную проверку вашего компьютера на наличие вирусов.

<p>1. Запустите любой браузер, например, Google Chrome.</p> <p>2. В поле поиска введите название 360 Total Security.</p> <p>3. Выберите ссылку на программу с высоким рейтингом</p>	<p>ru.wikipedia.org › wiki › 360_Total_Security</p> <p>360 Total Security — Википедия</p> <p>360 Total Security — комплексная антивирусная защита с 5 антивирусными движками: Kaspersky, Avira, QVM3; облачный движок 360 Cloud, ...</p> <p>Первый выпуск: 17 июля 2006 Разработчик: Qihoo</p> <p>Операционная система: Microsoft Window... Последняя версия: 10.8.0.1200 (03 дека...</p> <p>360-total-security.ru uptime.com › ... › Антивирусы</p> <p>360 Total Security 10.8.0.1060 для Windows - Скачать</p> <p>Скачать последнюю версию 360 Total Security для Windows. Обеспечьте полную защиту вашему ПК с пятью антивирусными движками. Защита ...</p> <p>★★★★★ Рейтинг: 4.3 - 11 голосов - Бесплатно - Windows</p> <p>bbiproг.org.ua › Безопасность › Антивирусы</p> <p>360 Total Security 10.8.0.1269 скачать бесплатно ...</p> <p>5 дней назад — Скачать 360 Total Security. 360 Total Security — мощная антивирусная программа с дополнительными возможностями по оптимизации и ...</p> <p>★★★★★ Рейтинг: 4.7 - 95 голосов - Бесплатно - Windows</p>
<p>4. Скачайте и установите 360 Total Security.</p> <p>5. Установите программу 360 Total Security для Windows.</p> <p>6. Если после установки программа предложит проверку, выполните ее</p>	
<p>7. Если на вашем компьютере антивирусная программа обнаружит вирусы или ошибки, то вы сможете их исправить, нажав на команду Исправить</p>	
<p>8. После исправления ошибок и вирусов, ответьте на вопросы:</p> <ol style="list-style-type: none"> 1) С помощью какого модуля можно очистить систему от ненужных объектов? 2) Каким образом можно улучшить быстродействие устройств? 	



Знание. Понимание

1. Какие виды вирусов вам известны?
2. Кто и для чего создает вредоносные программы?
3. Какие признаки проявления вирусов на компьютере вы знаете?
4. Какие антивирусные программы используются наиболее часто?
5. Какие антивирусные программы вы применяете на своем компьютере?
6. Существует ли такая вероятность, что та или иная антивирусная программа не может обезвредить какие-либо вирусы? Каковы будут ваши действия в таких случаях?



Применение

7. Что вы будете делать, если на ваш компьютер проникнет вирус? Как избежать заражения компьютерным вирусом?



Анализ

8. Классифицируйте вредоносные программы по принципу действия и их «вредности». Используйте различные источники информации.



Синтез

9. Почему даже чистая отформатированная флеш-карта может стать источником заражения вирусом?
10. Изучите вредоносную троянскую программу, используя различные источники информации.



Оценивание

11. Заполните диаграмму Венна. Запишите в нее то, что знаете об отличиях между компьютерными вирусами и антивирусными программами. Что объединяет их? Объясните.



Антивирусные программы основаны на следующих принципах: проверка файлов, проверка загрузочных секторов дисков и оперативной памяти, поиск в них известных и новых вирусов.



Большинство антивирусных программ сочетают в себе, во-первых, функцию постоянной защиты – **антивирусный монитор**. Антивирусный монитор запускается автоматически при старте операционной системы. Он работает в качестве фонового системного процесса, проверяя на вредоносность совершаемые другими программами действия.

Основная задача антивирусного монитора – обеспечение максимальной защиты от вредоносных программ.

Во-вторых, антивирусные программы содержат функцию защиты по требованию пользователя – **антивирусный сканер**. Такой сканер запускается по заранее выбранному расписанию или в произвольный момент пользователем. Он производит поиск вредоносных программ в оперативной памяти, а также на жестких и сетевых дисках компьютера.

2.3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

ВЫ НАУЧИТЕСЬ

Оценивать эргономичность пользовательских интерфейсов.

КЛЮЧЕВЫЕ СЛОВА

Интерфейс	Интерфейсі	Interface
Эргономика	Эргономика	Ergonomics
Юзабилити	Юзабилити	Usability
Пользовательский интерфейс	Пайдаланушы интерфейсі	User interface
Интерфейс командной строки	Командалық жол интерфейсі	Command line interface

Информационное пространство пользователя – это информационные ресурсы, доступные ему при работе за компьютером. Эффективная работа пользователя зависит от того, насколько понятен и удобен интерфейс работы с программным обеспечением или устройством. С различными видами интерфейсов вы сталкиваетесь ежедневно, например, когда заходите на сайт или пользуетесь сотовым телефоном.



Что такое интерфейс? Какие требования должны предъявляться к пользовательскому интерфейсу?



Интерфейс (от англ. *interface* – «место соприкосновения») – совокупность средств и методов взаимодействия между элементами системы.

Интерфейс делится на следующие виды (рис. 2.12).



Рис. 2.12. Виды интерфейсов

Термин «интерфейс» используется во многих областях науки и техники. Его значение относится к различному взаимодействию и управлению программным обеспечением или аппаратным оснащением.

Пользовательский интерфейс (на англ. *user interface* или сокращенно *UI*) – система правил и средств, обеспечивающая взаимодействие программы с пользователем.



Пользовательский интерфейс – это не только внешний вид программы или экран, который видит пользователь. Он объединяет в себе все элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением.

Элементы пользовательского интерфейса показаны на рис. 2.13.

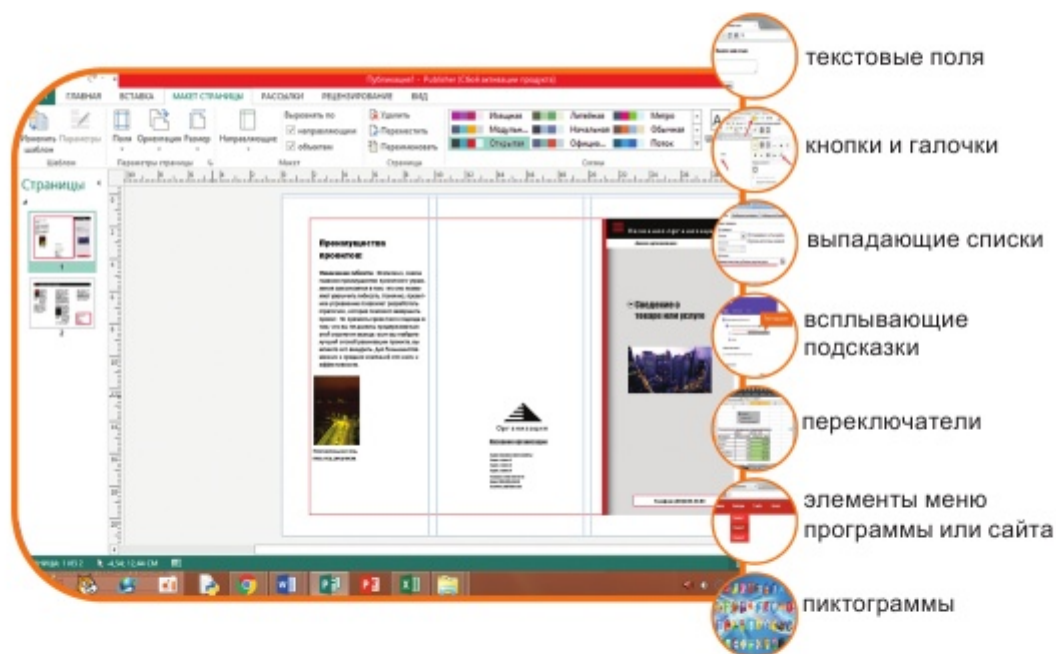


Рис. 2.13. Элементы интерфейса

Интерфейс командной строки (англ. *Command Line Interface, CLI*) – принцип организации пользовательского интерфейса между человеком и компьютером, в котором инструкции компьютеру даются в основном путем ввода с клавиатуры текстовых строк (команд) и текстового вывода результатов на экран.



В современном программном обеспечении имеется большое число команд. Если пользователь знает их, то после набора команд выполняется действие, соответствующее поступившей команде. Необходимость запоминать многочисленные команды отпала с появлением **графических интерфейсов**. Командная строка является наиболее надежным типом интерфейса. Для вызова командной строки (рис. 2.14) щелкните правой кнопкой в меню **Пуск** + командная строка (**Win + X**).

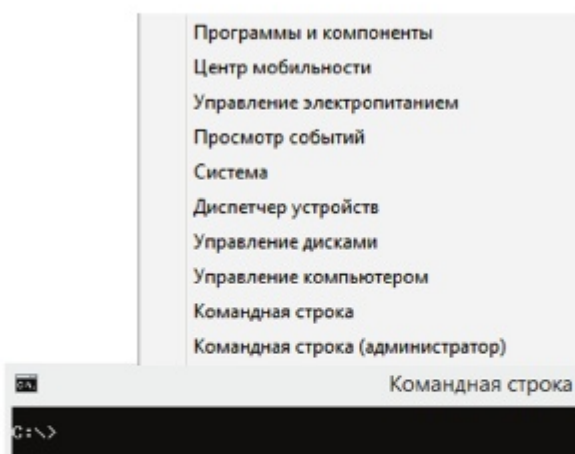


Рис. 2.14. Командная строка



Графический пользовательский интерфейс (англ. *Graphical user interface, GUI*) – система средств для взаимодействия пользователя с компьютером, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана (окон, значков, меню, кнопок, списков и т.п.).

Графический интерфейс основан на представлении всех системных объектов и функций в виде графических компонентов экрана. Графический интерфейс применяется во всех операционных системах, программах, web-страницах. Графический интерфейс дает пользователю возможность с помощью манипуляторов (клавиатура, мышь, джойстик и т.п.) управлять всеми элементами интерфейса.

Основным достоинством графического интерфейса является «дружеский интерфейс», так как он понятен для пользователей любого уровня. Графический интерфейс является частью пользовательского интерфейса и определяет взаимодействие с пользователем на уровне визуализированной информации. На рис. 2.15 приведены основные элементы графического интерфейса.

Основной экран Windows – это **Рабочий стол**, занимающий всю видимую часть монитора. На нем располагаются **Панель задач**, область уведомлений, меню **Пуск**, значки различных систем управления операционной системы, приложений, окна работающих программ, папки и файлы, ярлыки и т.д. (рис. 2.16).



Рис. 2.15. Основные элементы графического интерфейса

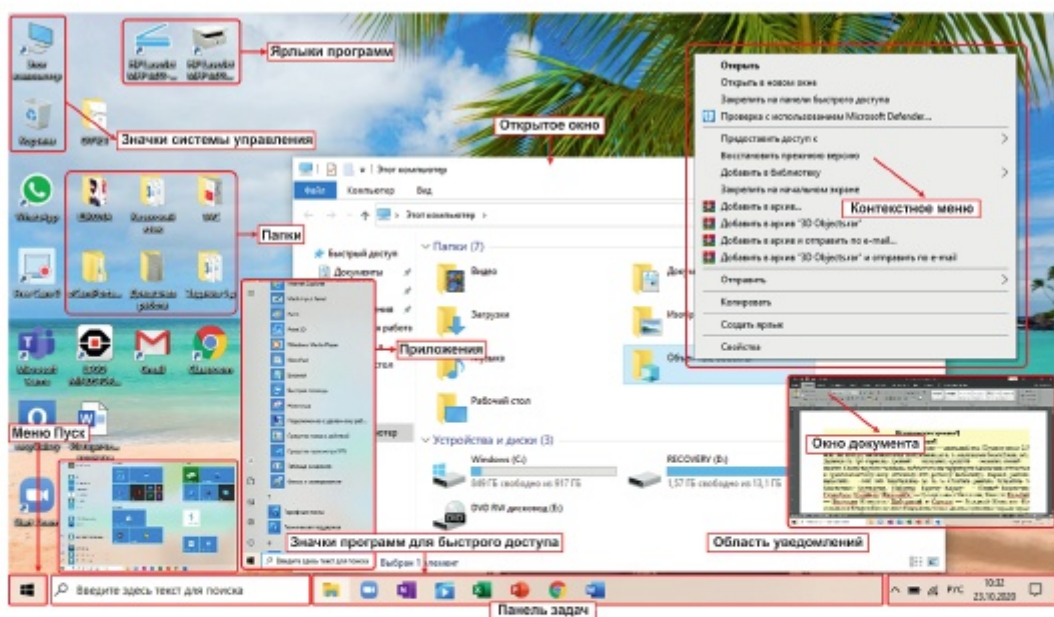


Рис. 2.16. Основные элементы графического интерфейса операционной системы **Windows**

Все эти основные элементы пользовательского интерфейса вы увидите в любой современной версии операционной системы Windows. Со временем их вид и функциональность изменяются и совершенствуются. Но независимо от этого общая концепция построения интерфейса управления системой в целом сохраняется.

Интерфейсы являются основой взаимодействия всех современных информационных систем. Научившись работать с одной программой в операционной системе Windows, вы сможете легко освоить другие программы, имеющие одинаковый интерфейс.

Эффективная работа пользователя зависит от критерия эргономичности (качества) и юзабилити любого интерфейса.



Юзабилити (от англ. *usability*) – это удобный, понимаемый, легкий в освоении и привлекательный для пользователей эргономичный продукт, соответствующий международному стандарту.

Термин «юзабилити» чаще всего применяется по отношению к системам с графическим интерфейсом пользователя: программное обеспечение, веб-сайты, операционные системы, приложения для мобильных устройств и др. А термин «эргономичность» к составляющим продукта – кнопки, переключатели, форма корпуса и т.д. Но такое разграничение не является строгим.



Эргономичность – способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях.

При оценке эргономичности пользовательского интерфейса нужно учитывать критерии, показанные в таблице 2.2.

Таблица 2.2

Критерии эргономичности интерфейса



Простота, понятность, скорость работы пользователей

	<p>Видимость статуса – пользователь получает обратную связь</p>
	<p>Управляемость и свобода для пользователя – поддержка функции отмены (undo) и повтора (redo)</p>
	<p>Согласованность и стандарты – пользователи видят согласованность и стандарты слов, ситуаций или операций</p>
	<p>Предотвращение ошибок – возможность устранять условия возникновения ошибок либо выявлять их и предупреждать пользователя о предстоящей проблеме</p>
	<p>Инструкции по использованию должны быть видимы или легко получаемы</p>
	<p>Гибкость и эффективность использования – возможность настраивать под себя часто используемые операции</p>

	<p>Эстетический дизайн – в интерфейсе не должно быть лишней информации</p>
	<p>Помочь пользователю понять и исправить ошибку – сообщения об ошибках, написанные простым языком</p>
<p>Этапы проектирования</p> 	<p>Справка и документация – информация должна быть простой в поиске, соответствовать задаче пользователя, описывать конкретную последовательность действий. Она не должна быть слишком большой</p>

Существуют еще другие виды интерфейсов, такие как текстовый интерфейс, пользовательский аудиоинтерфейс, тактильный интерфейс, перцептивный пользовательский интерфейс, интерфейс «мозг – компьютер» и др.



Знание. Понимание



1. Назовите ключевые слова на трех языках.
2. Что такое интерфейс? Пользовательский интерфейс?
3. Перечислите элементы пользовательского интерфейса.
4. Какие виды пользовательского интерфейса вам знакомы?
5. В чем особенность командного интерфейса?
6. Охарактеризуйте графический интерфейс.
7. Назовите основные элементы графического интерфейса.



8. Что такое юзабилити и эргономичность?
9. По каким критериям оценивается эргономичность пользовательского интерфейса?
10. Приведите примеры пользовательского интерфейса.

Применение. Анализ



11. Запустите командную строку **Пуск – Программы – Служебные – Командная строка**. Введите в командную строку команду **DIR**. Проанализируйте, для чего предназначена эта команда.



12. Попробуйте запустить из командной строки любую программу с вашего компьютера. Сравните графический и командный интерфейсы.

13. Запустите интегрированную среду разработки IDLE среды Python и текстовый процессор Word. Сравните их графический интерфейс в диаграмме Венна в PowerPoint (2007, 2013, 2016).

Для создания диаграммы Венна нажмите кнопку **SmartArt** на вкладке **Вставка** в группе **Иллюстрации**. В коллекции **Выбор графического элемента SmartArt** нажмите кнопку **Связь**, выберите макет диаграммы Венна – **Простая Венна**, затем нажмите кнопку **ОК** (рис. 2.17).



Рис. 2.17. Простая диаграмма Венна



Синтез. Оценивание



14. Поисковая работа с интернет-источниками. Используя критерии оценки эргономичности пользовательского интерфейса (табл. 2.2), проведите анализ нескольких сайтов. Оцените интерфейсы выбранных сайтов. Выберите лучший. Обоснуйте свой выбор.

15. Нарисуйте интерфейс своей будущей обучающей программы, обсудите, как она будет работать. Оцените ее интерфейс.



Пользовательский графический интерфейс

Чем отличается разработка от проектирования графического интерфейса? **Разработка** графического интерфейса – процесс, включающий в себя программирование, а **проектирование** – создание макетов экранов на основе исследований пользователей.

Проектирование графического интерфейса требует максимальной реализации возможности программы. Главная задача интерфейса – сделать взаимодействие с пользователем простым и понятным. Интерфейс не должен перегружать пользователя большим количеством элементов: кнопок, графических элементов, меню и текста.

Наряду с достоинствами графического интерфейса существуют недостатки:

- большее потребление памяти в сравнении с текстовым интерфейсом;
- сложность организации удаленной работы;
- невозможность автоматизации, если она не была заложена автором программы;
- сложность использования графического интерфейса незрячими людьми.

ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ II

1. Компьютерные сети – это:

- а) компьютеры для сбора информации и их совместного использования;
- б) объединение компьютеров для обмена информацией и совместного использования ресурсов;
- в) объединение компьютеров для обмена информацией;
- г) объединение компьютеров для совместного использования информации.

2. К какой категории разделения относятся компьютерные сети: локальные, региональные, глобальные?

- а) По типу организации работы компьютеров в сети.
- б) По топологии.
- в) По масштабу.
- г) По типу среды передачи.

3. Компьютер-сервер – это:

- а) центральный компьютер, который распределяет общие ресурсы между многими пользователями;
- б) компьютер, который распределяет общие ресурсы между многими пользователями;
- в) центральный компьютер, который передает информацию;
- г) центральный компьютер, который распределяет ресурсы между одним пользователем.

4. На рисунке 2.18 изображена топология сетей. Сеть какого типа отсутствует?

- а) Шинная.
- б) Кольцевая.
- в) Звезда.
- г) Ячеистая.
- д) Иерархическая звезда.

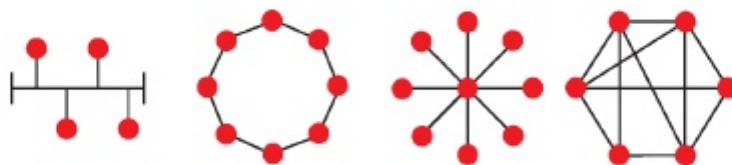


Рис. 2.18. Топология сетей

5. В чем преимущество беспроводной связи стандарта 5G?

- а) Беспроводные телефонные технологии.
- б) Шифрование цифрового сигнала.
- в) Доступ в Интернет.
- г) Высокая скорость Интернета.
- д) Мобильный Интернет.

6. Вредоносная программа – любое программное обеспечение, специально созданное для того, чтобы:

- а) оказывать помощь пользователю, работающему с компьютерной сетью;
- б) причинять ущерб компьютеру, серверу или компьютерной сети и хранящимся на них данным;
- в) причинять ущерб данным;
- г) причинять ущерб персональным данным пользователя.

7. Какие виды антивирусных программ вы изучили?

- а) Детекторы, вирусы, ревизоры, фильтры, вакцины.
- б) Детекторы, доктора, вирусы скриптов, фильтры, вакцины.
- в) Детекторы, доктора, ревизоры, фильтры, макровирусы.
- г) Детекторы, доктора, ревизоры, фильтры, вакцины.

8. Выберите элементы графического интерфейса. (Возможен выбор нескольких вариантов ответа.)

- а) Строка заголовка.
- б) Панель задач.
- в) Ярлыки.
- г) Панель инструментов.
- д) Рабочий стол.
- е) Окно.
- ж) Курсор.

9. Выберите критерии при оценке эргономичности пользовательского интерфейса. (Возможен выбор нескольких вариантов ответа.)

- а) Простота, понятность.
- б) Скорость работы пользователей.
- в) Сложность понимания.
- г) Управляемость и свобода для пользователя.
- д) Несогласованность со стандартами.
- е) Инструкции по использованию.
- ж) Эстетический дизайн.

10. К юзабилити относятся:

- а) неудобство использования, пригодность использования, эргономичность;
- б) удобство использования, пригодность использования, неэргономичность;
- в) удобство использования, непригодность использования, эргономичность;
- г) удобство использования, пригодность использования, эргономичность.

11. Способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях называется:

- а) юзабилити;
- б) интерфейс;
- в) эргономичность;
- г) достаточность.

12. Выберите виды интерфейса. (Возможен выбор нескольких вариантов ответа.)

- а) Аппаратный.
- б) Системный.
- в) Программный.
- г) Аппаратно-программный.
- д) Оконный.
- е) Пользовательский.
- ж) Текстовый.

13. Отличительными особенностями компьютерного вируса являются:

- а) значительный объем программного кода;
- б) необходимость запуска со стороны пользователя;
- в) способность к повышению помехоустойчивости операционной системы;
- г) маленький объем; способность к самостоятельному запуску и многократному копированию кода, к созданию помех корректной работе компьютера;
- д) легкость распознавания.

14. Назначение антивирусных программ под названием «детекторы»:

- а) обнаружение и уничтожение вирусов;
- б) контроль возможных путей распространения компьютерных вирусов;
- в) обнаружение компьютерных вирусов;
- г) «излечение» зараженных файлов;
- д) уничтожение зараженных файлов.

РЕШЕНИЕ ЗАДАЧ С ПОМОЩЬЮ
ЭЛЕКТРОННЫХ ТАБЛИЦ

3.1. ТАБЛИЦЫ В ТЕКСТОВОМ ПРОЦЕССОРЕ

ВЫ НАУЧИТЕСЬ

Форматировать элементы таблицы в текстовом процессоре.

КЛЮЧЕВЫЕ СЛОВА

Текстовый процессор	Мәтіндік процессор	Word processor
Строка	Жол	Line
Столбец	Баған	Column
Ячейка	Ұяшық	Cell

Таблицы используются для упорядочения и представления данных, рисунков, объектов и других элементов. Они позволяют выстроить числа в столбцы, а затем отсортировать их, а также выполнить различные вычисления.



Таблица – это объект, который добавляется в документ для структурирования данных.

Таблица состоит из **столбцов** и **строк**, а их пересечения называются **ячейками**. Таблица имеет границу в виде тонкой сплошной линии черного цвета. Граница сохраняется при печати, а в случае удаления границы линии сетки отображаются на экране. На рис. 3.1 представлены компоненты таблицы: маркер перемещения, интервалы между ячейками, символы конца строки, символы конца ячейки, маркер изменения размера таблицы.

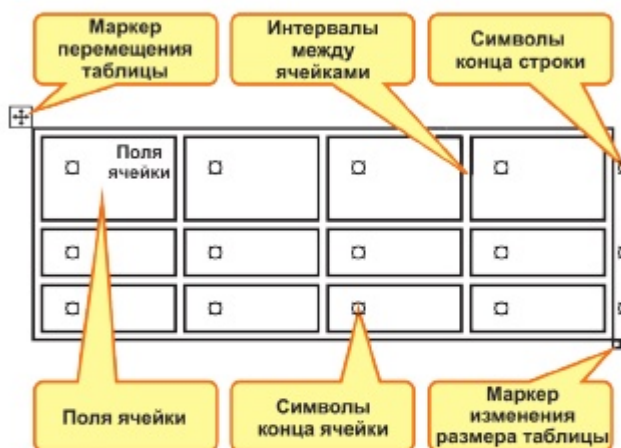


Рис. 3.1. Компоненты таблицы

размера таблицы. **Символы конца строки и конца ячейки** видны в таблице, если включена команда ¶ (скрытые символы форматирования) из группы **Абзац** вкладки **Главная**.

Шаг за шагом



Создадим таблицу разными способами в текстовом процессоре **Word**. Чтобы вставить таблицу в документ, нужно установить курсор туда, куда должна быть добавлена таблица.

Первый способ.

1. Выберите на **Ленте** вкладку **Вставка**, группу **Таблицы**, откройте список **Таблица**.

2. В списке **Вставка таблицы** (рис. 3.2) выделите нужное число строк и столбцов (максимальный размер таблицы 10 · 8).

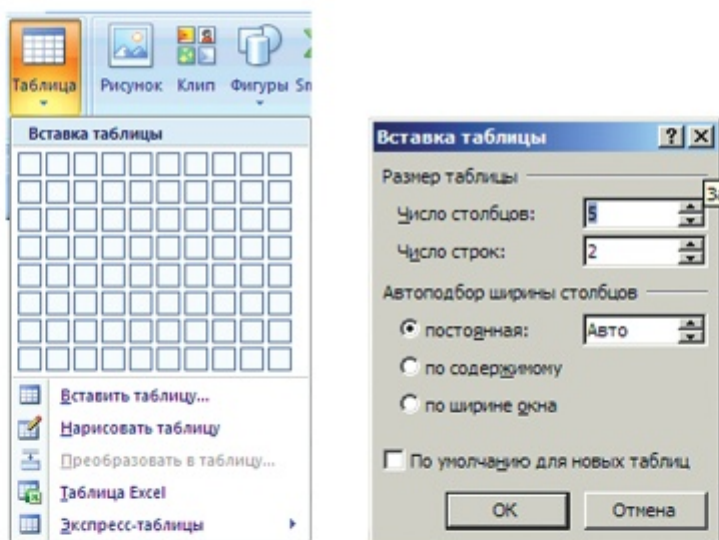


Рис. 3.2. Создание таблицы

Второй способ.

1. Выберите на **Ленте** вкладку **Вставка**, группу **Таблицы** (рис. 3.2), откройте список **Таблица**.

2. В списке **Вставка таблицы** выберите пункт **Вставить таблицу**.

3. В окне **Вставка таблицы** укажите число строк и столбцов.

4. Нажмите на кнопку **ОК**.

Третий способ.

1. Выберите на **Ленте** вкладку **Вставка**, группу **Таблицы**, откройте список **Таблица** (рис. 3.3).

2. Выберите пункт **Нарисовать таблицу**. Карандашом нарисуйте таблицу: рисуйте сначала внешний контур, затем – внутренние разделители. Таблица сначала обозначается пунктиром, показывающим, как будет располагаться линия. Если отпустить кнопку мыши, линия станет разделительной.

Все таблицы, создаваемые в текстовом процессоре, выглядят одинаково: набор белых ячеек с черными границами. С помощью команд текстового процессора Word за несколько секунд таблицу можно сделать яркой и эффектной. Для этого используйте дополнительные вкладки **Конструктор** и **Макет (Работа с таблицами)** (рис. 3.3), которые появляются после создания таблицы или с помощью контекстного меню выделенной части таблицы.




Рис. 3.3. Дополнительные вкладки **Конструктор** и **Макет**

Необходимо помнить, что вкладки **Конструктор** и **Макет** будут активны при размещении курсора в таблице.

С помощью дополнительной вкладки **Конструктор** можно:

– добавить (удалить) разделительные линии (используя карандаш, коман-

да **Нарисовать таблицу**); можно также удалить их ластиком  с вкладки **Конструктор (Работа с таблицами)**;

– к созданной таблице добавить строки (столбцы, ячейки) – поставить курсор в ячейку таблицы и вызвать контекстное меню **Вставить**, **Вставить столбцы** (рис. 3.4) или на вкладке **Макет (Работа с таблицами)**, группа **Строки и столбцы**, команды **Вставить сверху/снизу/слева/справа**;

– добавить следующую строку в конец таблицы – поставить курсор в последнюю ячейку последней строки таблицы и нажать клавишу **Tab**;

– изменить размер строк/столбцов – контекстное меню выделенной таблицы, окно **Свойства таблицы**, вкладка **Столбец/Строка** – указать ширину/высоту или на вкладке **Макет**, группа **Размер ячейки**, команды **Высота строки** таблицы/**Ширина столбца** таблицы;

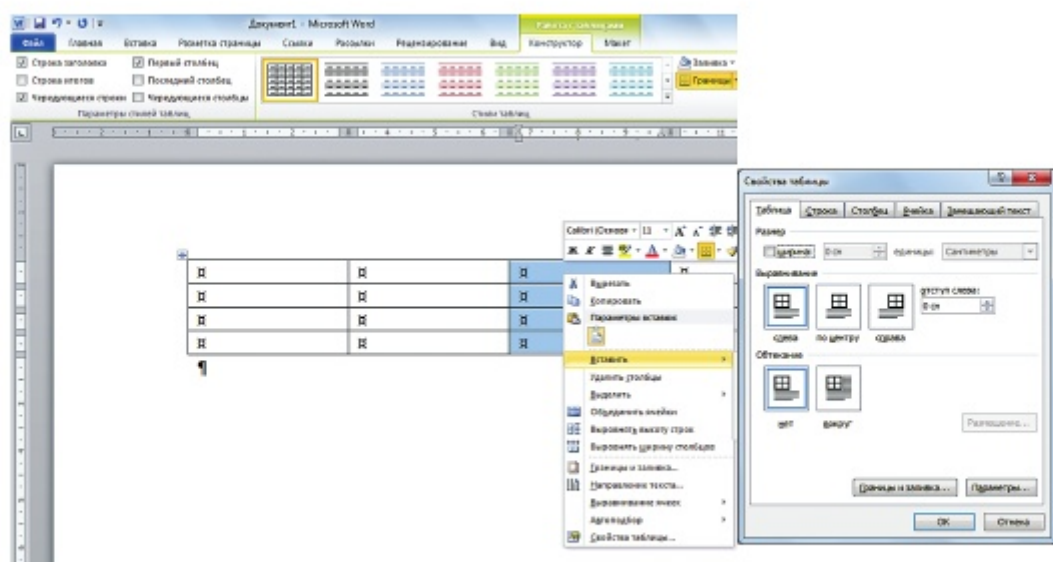


Рис. 3.4. Работа с таблицей

– **столбцы/строки** копировать, удалять, вставлять или перемещать – **Выделить столбец/строку**, вызвать контекстное меню (правая кнопка мыши), выбрать нужный пункт.

Для ввода данных в таблицу нужно придерживаться следующих правил:

– щелкнуть на ячейке и ввести текст с клавиатуры или вставить из буфера обмена при копировании текста. Если текст не помещается в строке, то он переносится на другую строку и увеличивает высоту строки;

– для изменения ориентации текста в ячейке необходимо установить курсор в ячейку и в меню **Формат** выбрать команду **Направление текста**;

– для изменения выравнивания текста в ячейке на панели инструментов **Таблицы и границы** выберите параметр выравнивания по **вертикали** и **горизонтали**;

– для перемещения, копирования и удаления текста в ячейках необходимо выделить этот текст. Выделенный текст можно удалять клавишей **Delete** или **Backspace**.

Для **редактирования элементов** (ячеек, строк, столбцов) необходимо выделить эти элементы, а затем использовать меню **Таблица** или **контекстное меню**. К операциям редактирования таблиц относятся:

– вставить и удалить строки и столбцы;

- объединить и разбить ячейки;
- разбить таблицу.

После того как установлены формат таблицы и ширина полей, можно приступить к **форматированию** элементов таблицы.

Текстовый процессор Word содержит несколько десятков команд для профессионального форматирования данных.



Форматирование – это изменение свойств данных, определяющих внешний вид, без изменения содержимого.

Форматирование данных означает, что можно произвести изменение полей, абзацев и отступов, выравнивание текста (по левому, по правому краям, по центру, по ширине), установку межстрочного интервала, вставку графических объектов и др.

Для форматирования таблицы выделите таблицу, выберите вкладку **Конструктор** группы **Стили таблиц**. Кнопка **Дополнительные параметры** раскрывает окно **Встроенные стили**, откуда можно выбрать любой стиль для быстрого оформления таблицы.



Знание. Понимание



1. Для чего применяется таблица?
2. Перечислите основные компоненты таблицы. Для чего они нужны?
3. Какие способы создания таблиц вы знаете?
4. Что такое форматирование данных?
5. Как форматировать таблицу с применением стилей?
6. Какие дополнительные вкладки появляются при работе с таблицами?



Применение



7. Создайте в текстовом процессоре Word таблицу по химии «Свойства некоторых веществ» из **четырёх строк** и **четырёх столбцов**.
 - 1) Выделите всю таблицу и установите шрифт **Arial** в **12** пунктов.
 - 2) Задайте в таблице заголовок, объединив ячейки первой строки.

3) Заполните созданную таблицу данными о свойствах веществ (см. табл. 3.1).

Таблица 3.1

Свойства некоторых веществ			
Свойства	Кислород	Железо	Вода
Агрегатное состояние	Газ	Твердое	Жидкость
Цвет	–	Серое	–

4) Если вам при заполнении таблицы понадобится изменить ширину столбца, то установите курсор на границу столбца так, чтобы он приобрел вид двусторонней стрелочки. Нажав и удерживая левую кнопку мыши, расширьте или уменьшите ширину столбца.

5) Добавьте еще одну строку в таблице и запишите в ней данные о **запахе** вещества (табл. 3.2).

Таблица 3.2

Свойства некоторых веществ			
Свойства	Кислород	Железо	Вода
Агрегатное состояние	Газ	Твердое	Жидкость
Цвет	–	Серое	–
Запах	–	–	–

6) Отформатируйте созданную вами таблицу (табл. 3.3):

- сначала отцентрируйте заголовок таблицы. Для этого выделите первую строку, нажмите кнопку **По центру**, затем – кнопку **Полужирный**;
- сделайте обрамление таблицы, как на образце.

7) Для обрамления таблицы используйте вкладку **Главная** из группы **Абзац**. В пункте меню выберите команду **Границы и заливка...**, появится окно, которое разделено на три части: **Тип**, **Тип** и **Образец**. Выберите в первой части окна **Тип**: обрамление – **все**. Во второй части окна установите в параметре **Ширина**: 1,5 пт (пунктов), **Цвет**: синий. В третьей части окна **Образец** – **Применить к**: выберите из списка параметр: **таблице**. Нажмите на кнопку с командой **ОК**.

Таблица 3.3

Свойства некоторых веществ			
Свойства	Кислород	Железо	Вода
Агрегатное состояние	Газ	Твердое	Жидкость
Цвет	–	Серое	–
Запах	–	–	–



Анализ



8. Создайте кроссворд в текстовом процессоре Word (рис. 3.5). После ввода ответов вы узнаете, какое слово зашифровано в закрашенном столбце. Для вставки символов используйте вкладку **Вставка** на **Ленте**, из группы **Символы** выберите **Другие символы**, **Шрифт: Wingdings**, **Wingdings1**, **Wingdings2**.

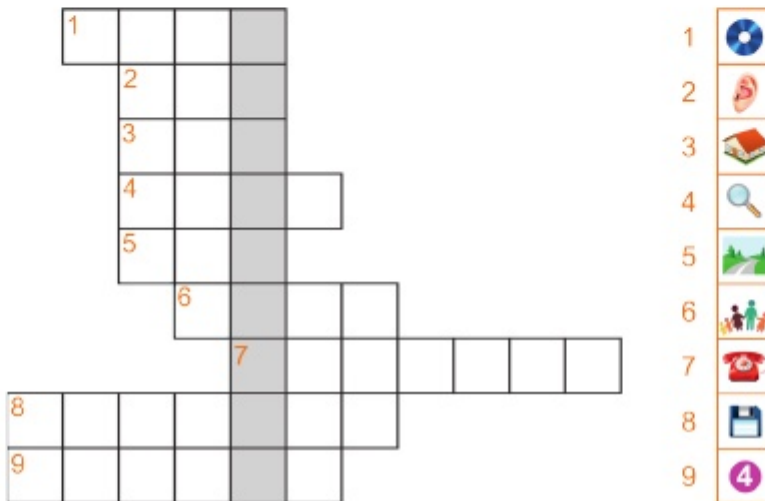


Рис. 3.5. Кроссворд «Устройство, выполняющее заданную последовательность операций»



Синтез



9. Создайте и напечатайте таблицу, в которой можно увидеть сведения о 5–7 книгах, хранящихся в домашней библиотеке. Подумайте, какие столбцы должны быть в таблице, как они должны называться. Добавьте в таблицу еще 3 строки и заполните их.

10. Создайте таблицу «Расписание уроков» в текстовом процессоре Microsoft Word (см. табл. 3.4), выполнив действия по алгоритму.

1) Выберите во вкладке **Вставка** группы **Таблицы** команду **Таблица**, во **Вставка таблицы** укажите **7 столбцов** и **7 строк**. Появится таблица, в которой по умолчанию будет установлена ширина столбцов и высота строк.

- 2) Для форматирования таблицы выделите всю таблицу, в контекстном меню выберите команду **Свойства таблицы...**, в окне установите параметры: ширину столбца – **2,5–3 см** и высоту строки – **1,5 см**. Можно также форматировать таблицу, применив дополнительные вкладки **Конструктор** и **Макет**, которые появляются при выделении таблицы.
- 3) Поменяйте высоту первой строки, установив **0,5 см**, а ширину первого столбца – **1 см**.
- 4) Введите заголовок: **№, понедельник, вторник, среда, четверг, пятница, суббота, воскресенье**.
- 5) Для ввода нумерации строк выделите столбец со второй строки до конца и выполните: вкладка **Главная**, группа **Абзац**, команда **Нумерация** – выберите из **Библиотеки нумерованный список 1., 2., 3., ...**
- 6) Выполните форматирование для заголовка, установив шрифт **Times New Roman**, размер шрифта **11**, выравнивание **по центру, полужирный шрифт**. Проведите аналогичное форматирование для первого столбца.
- 7) В пустые ячейки таблицы введите учебные предметы по расписанию.
- 8) Выполните форматирование для текста, установив шрифт **Times New Roman**, размер шрифта **11**, выравнивание **по левому краю, обычный шрифт**.
- 9) Сделайте заливку фона любым цветом, одновременно выделив строку и столбец, нажав на клавишу **Ctrl**.
- 10) Сохраните таблицу.

Таблица 3.4

Расписание уроков

№	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
1.	Алгебра	Казахский язык	Геометрия	Русский язык	Физическая культура	Физика
2.	Информатика	Русский язык	Русский язык	Литература	Геометрия	Биология
3.	Литература	Физика	Литература	Алгебра	Биология	История Казахстана
4.	Физическая культура	Алгебра	Химия	Казахский язык	Технология	Английский язык
5.	История Казахстана	Казахская литература	Казахский язык	Английский язык	Химия	Классный час
6.	Английский язык	География	Биология	Самопознание	География	



Оценивание

11. Среднесуточная доля казахстанских телеканалов за сентябрь 2020 года показана на рис. 3.6.

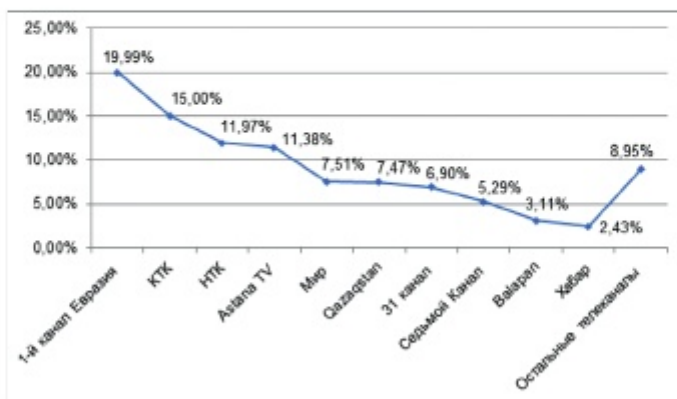


Рис. 3.6. Среднесуточная доля казахстанских телеканалов за сентябрь 2020 года

На основе данных графика составьте таблицу в текстовом процессоре.



Microsoft Excel остается лучшим среди электронных таблиц как в бизнесе, образовании, так и в быту. Но существует несколько альтернативных электронных таблиц. Все они доступны бесплатно, могут выполнять аналогичную работу, как в Microsoft Excel. Приведем некоторые из них.

1) **Google Таблицы** – это аналог программному обеспечению Microsoft Excel. Электронная таблица Google размещена на облачной платформе и может эффективно применяться для сотрудничества. Также возможно использование автономного приложения для персональных компьютеров и мобильных устройств с доступом к ней в любом месте и на любой платформе.

2) **Office PlanMaker** – эффективная бесплатная замена Microsoft Excel. В Office PlanMaker много красивых шаблонов для визуализации данных в 2D, а также в 3D. Отлично сочетается с Microsoft Excel 2016. В нее включены более 350 формул для математических функций, поддерживаемые платформами Windows и Linux.

3) **WPS Office**. Электронные таблицы поддерживают все известные форматы листов. Также является эффективным инструментом для совместной работы с поддержкой облачного сервиса. В настоящее время он доступен для операционных систем Windows, Mac, Linux, iOS и Android.

3.2.

ФОРМАТИРОВАНИЕ ЭЛЕМЕНТОВ ЭЛЕКТРОННЫХ ТАБЛИЦ

ВЫ НАУЧИТЕСЬ

Форматировать элементы электронной таблицы.

КЛЮЧЕВЫЕ СЛОВА

Электронная таблица	Электрондық кесте	Spreadsheet
Форматирование	Форматтау	Formatting
Рабочая книга	Жұмыс кітабы	Workbook
Рабочий лист	Жұмыс парағы	Worksheet



При работе с информацией часто требуется создавать таблицы, проводить различные вычисления с использованием формул, выполнять различные экономические, бухгалтерские и инженерные расчеты, строить графики и диаграммы, создавать математические модели, проводить анализ данных.

Для проведения расчета данных, представленных в виде таблицы, были разработаны пакеты прикладных программ – **табличные процессоры**. Это специальный комплекс программ для управления электронной таблицей. В настоящее время наиболее популярным табличным процессором является электронная таблица Excel.

Электронная таблица Excel – это приложение, позволяющее автоматизировать обработку данных, представленных в виде таблиц.



Основной документ, используемый электронной таблицей Excel для хранения и обработки данных, называется **книгой (рабочей книгой)** и имеет расширение **.xlsx, xls**.

Рабочая книга состоит из отдельных **рабочих листов**, каждый из которых может содержать данные. По умолчанию каждая вновь созданная рабочая книга содержит 3 рабочих листа, количество их можно изменять. Каждый рабочий лист можно переименовывать.

Рабочий лист разделен на **строки и столбцы**, на пересечении которых образуются ячейки. **Ячейка** – основная единица хранения данных.

Каждая ячейка имеет свой адрес, который составляется из заголовка столбца (**A, B, C, D...**), расположенного в верхней части таблицы, и заголовка строки, находящегося в левой части таблицы (**1, 2, 3, 4...**).

Активизировать электронную таблицу Excel можно разными способами, например командами **Пуск – Все программы – Microsoft Office – Microsoft Excel** или двойным щелчком по значку **Excel** на **Рабочем столе**.

После запуска приложения открывается окно программы (рис. 3.7), состоящее из:

- 1) **строки заголовка** с названием рабочей книги в центре окна;
- 2) **панели быстрого доступа** для быстрого доступа часто используемых команд;
- 3) **ленты с вкладками** для быстрого нахождения команд;
- 4) **строки формул (командная строка)** с отображением адреса активной ячейки и ее содержимым;
- 5) **рабочей зоны** текущего рабочего листа, состоящей из **Заголовка столбцов** и **Номеров строк**;
- 6) **строки состояния**, выводящей сообщение о текущей работе (**Готово**, **Ввод**, **Укажите** и т.п.).

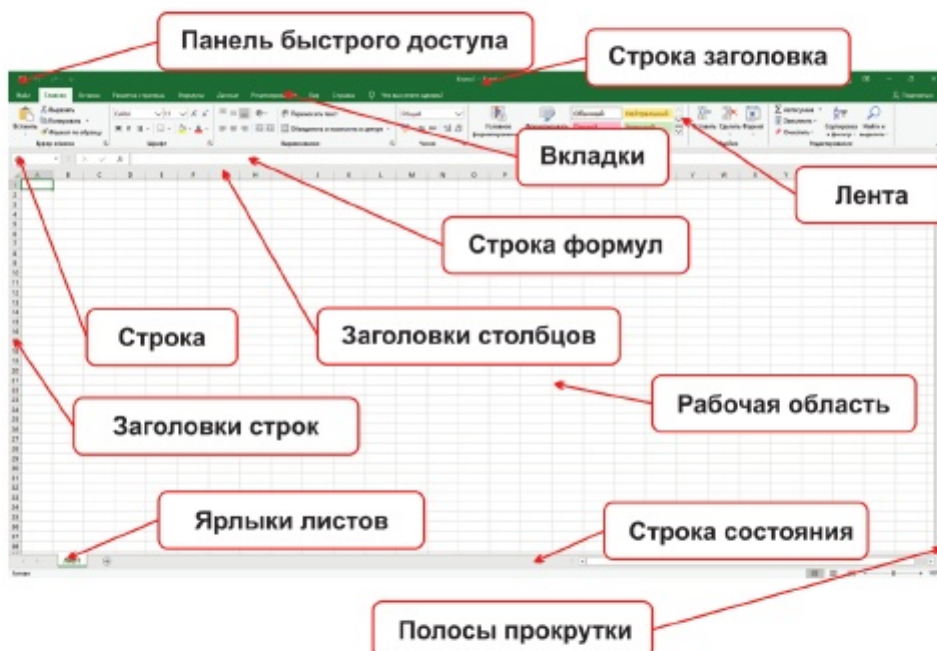


Рис. 3.7. Главное окно Excel

Одновременно можно работать на нескольких листах, вводить данные, править, форматировать, производить с ними вычисления.

Форматирование в электронной таблице Excel применяется для облегчения восприятия данных. Форматировать можно выделенные (активные) ячейки или области. Отформатированные ячейки сохраняют свой формат, пока к ним не будет применен новый формат или не удален старый. При вводе значения в ячейку к нему применяется уже используемый в ячейке формат.

Для быстрого копирования форматов из выделенных ячеек в другие ячейки можно использовать команду **Формат по образцу** (рис. 3.8) вкладки **Главная**, группа **Буфер обмена**.

Форматирование подразумевает изменение:

- цвета шрифта;
- цвета заливки;
- границ;
- центрирование и т.д.

Изменение разрядности чисел производится командами группы **Число: Увеличить разрядность** или **Уменьшить разрядность** (рис. 3.9).



Рис. 3.8. Формат по образцу

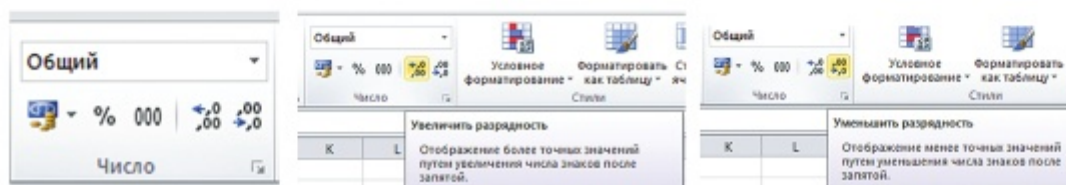


Рис. 3.9. Разрядность числа

Цвета, размер и начертание шрифта, цвета заливки и границы в выделенной области меняются командами из группы **Шрифт** (рис. 3.10) вкладки **Главная** аналогично форматированию информации в других офисных приложениях.

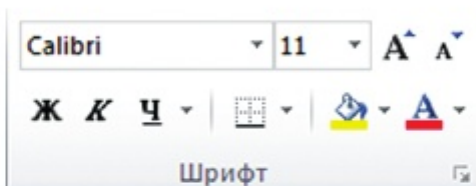



Рис. 3.10. Команды группы Шрифт

Несколько ячеек таблицы можно объединить командой **Объединить и поместить в центре** (вкладка **Главная**, группа **Выравнивание**) .

Отцентрировать информацию в ячейке можно командами из группы **Выравнивание** на вкладке **Главная** (рис. 3.11, а). Выравнивать можно по горизонтали и вертикали.

Ориентация текста внутри ячейки выполняется командой **Ориентация** из группы **Выравнивание** на вкладке **Главная** (рис. 3.11, б).

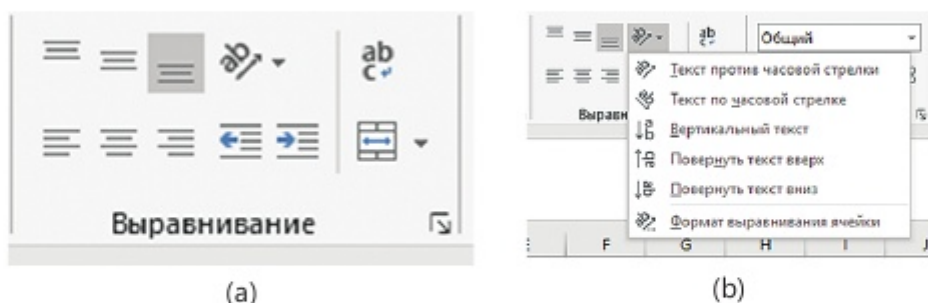


Рис. 3.11. Команды группы **Выравнивание**



Шаг за шагом

Изменение ширины столбца или высоты строки

1. Выделите столбец или несколько столбцов.
2. Поместите курсор в заголовке таблицы между столбцами так, чтобы белый крест изменился на черный со стрелками ⇄.
3. Чтобы увеличить (уменьшить) ширину столбцов, передвиньте левую кнопку мыши вправо (влево) и отпустите кнопку мыши.
4. Если выделить несколько столбцов, то при изменении ширины столбцов изменяется ширина всех столбцов.

Ширину столбца можно быстро задавать в соответствии с набранным текстом. Например:

– наведите курсор мыши на правую границу столбца возле его имени и дважды щелкните левой кнопкой мыши;

– граница столбца автоматически выравнивается по самому длинному слову, набранному в его ячейках (рис. 3.12).

	A	B
1	красный	
2	оранжевый	
3	желтый	
4	зеленый	
5	голубой	
6	синий	
7	фиолетовый	
8		

Рис. 3.12. Автоматическое выравнивание

В электронной таблице Excel предлагаются **готовые стили таблицы** (рис. 3.13), которые можно использовать для быстрого форматирования таблицы. Если такие стили не соответствуют потребностям пользователя, можно создать и применить настраиваемый стиль таблицы.

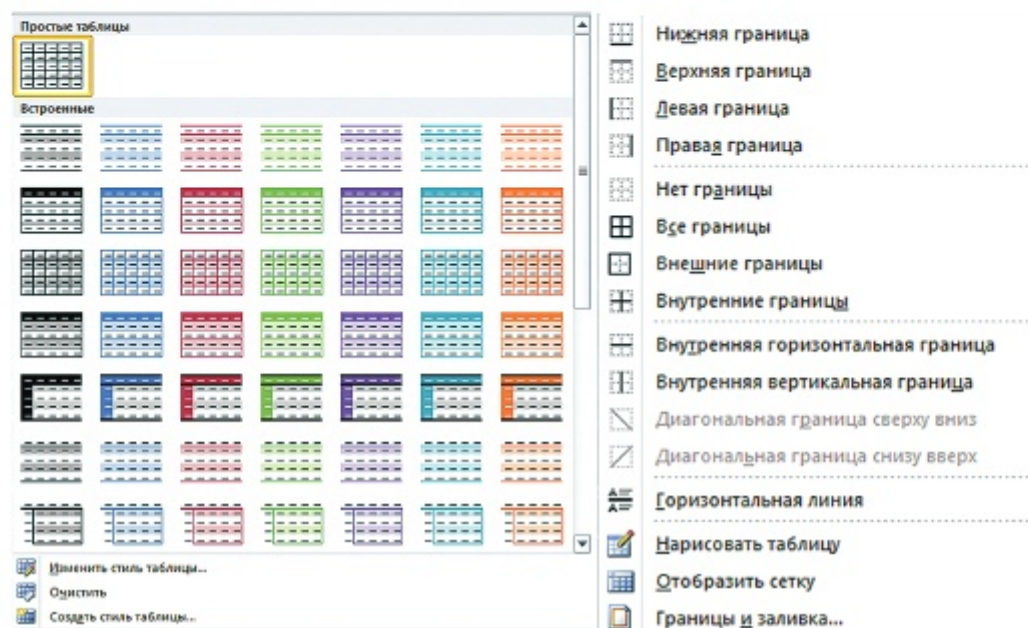


Рис. 3.13. Стили форматирования и Границы

Шаг за шагом



Для добавления границы:

- 1) выберите ячейки;
- 2) щелкните на стрелку раскрывающегося списка, выбрав команду **Границы и Заливка** на вкладке **Главная** (рис. 3.13);
- 3) выберите стиль границы, который вы хотите использовать;
- 4) вы можете нарисовать границы и менять **Тип линии** и **Цвет** границ с помощью **Нарисовать границы** инструментами в нижней части границы выпадающего меню.

Чтобы изменить цвет шрифта:

- 1) выберите ячейки, в которых нужно изменить цвет шрифта;
- 2) щелкните стрелку раскрывающегося списка рядом с цветом шрифта командой меню **Цвет текста** на вкладке **Главная**;

3) выберите цвет шрифта, который вы хотите использовать.

Одним из преимуществ электронных таблиц является то, что не все данные при создании таблиц нужно вводить вручную. Ввод данных, которые повторяются, имеет определенную закономерность и его можно автоматизировать.

Для этого используется **Маркер заполнения** (рис. 3.14).

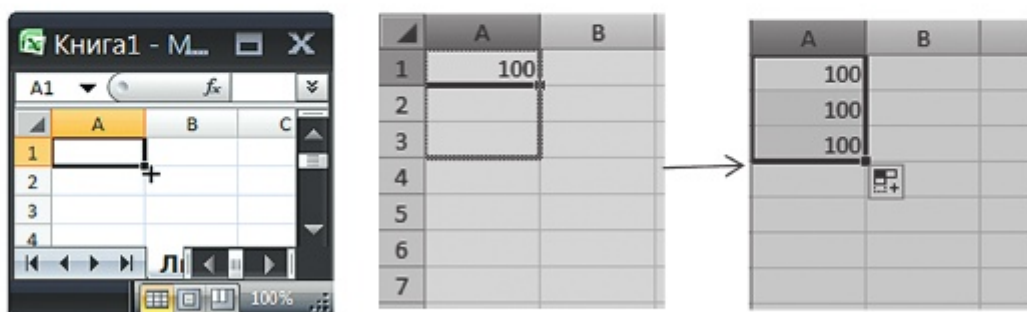


Рис. 3.14. Маркер заполнения



Маркер заполнения – небольшой черный квадрат в правом нижнем углу выделенной ячейки или выделенного диапазона.

При наведении на **Маркер заполнения** указатель мыши принимает вид тонкого черного крестика.

Маркер заполнения удобен для заполнения соседних ячеек содержимым выделенной ячейки.



Шаг за шагом

1. Наведите на **Маркер заполнения** указатель мыши, чтобы он принял вид черного креста.

2. Нажмите левую клавишу мыши и, не отпуская ее, протяните вниз (или вправо, влево, вверх) на нужное количество ячеек.

3. Отпустите кнопку. Значения будут скопированы.

С помощью **Маркера заполнения** можно быстро формировать числовые последовательности, последовательности дат и рабочих дней, работать со списками автозаполнения.

Автозаполнение – это операция ускоренного автоматического заполнения ячеек данными, текстом, числами или формулами без использования обычных способов ввода.



Автозаполнение позволяет:

- быстро вводить некоторые стандартные или созданные пользователем наборы данных (названия месяцев, названия видов товаров и т.п.);
- быстро заполнять ячейки числовыми последовательностями (нумерацией, датами, часами и т.д.);
- быстро вводить в ячейки одинаковые данные;
- быстро вводить новые формулы, используя уже введенные формулы.

Последовательность данных в Excel может быть стандартной, т.е. изначально заложенной, или же она может задаваться пользователем.

Последовательности – это введенные в соседние ячейки данные, представляющие собой некоторую закономерную последовательность.



Шаг за шагом



Введите названия месяцев с января по июнь (слева направо) с использованием стандартной последовательности. Для этого:

- в начальную ячейку введите значение **Январь**;
- нажмите на кнопку **Ввод** на панели формул (или нажмите **Enter**, а затем вернитесь в ячейку, где введено значение **Январь**);
- установите курсор мыши на **Маркер заполнения**;
- растяните выделение автозаполнения вправо на 5 ячеек и отпустите кнопку мыши.

Если растянете выделение влево или вверх, то тоже произойдет заполнение месяцами, но в обратном порядке, т.е. **Январь, Декабрь, Ноябрь** и т.д. По диагонали автозаполнение не выполняется (рис. 3.15).

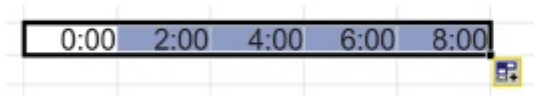
	Январь	Февраль	Март	Апрель
2001	1	3	5	7
2002	1	3	5	7
2003	1	3	5	7
2004	1	3	5	7
2005	1	3	5	7

Рис. 3.15. Автозаполнение

Чтобы по-другому создать последовательность, например с шагом два, необходимо:

- в соседние ячейки ввести два значения-образца с требуемым шагом;
- обычным способом выделить эти ячейки;
- растянуть выделение автозаполнения до требуемого размера.

В результате (если, например, ввести значения **00:00** и **02:00**) продолжением последовательности будут значения **04:00**, **06:00**, **08:00** и т.д.



Используя данный способ, можно получить следующие последовательности (рис. 3.16):

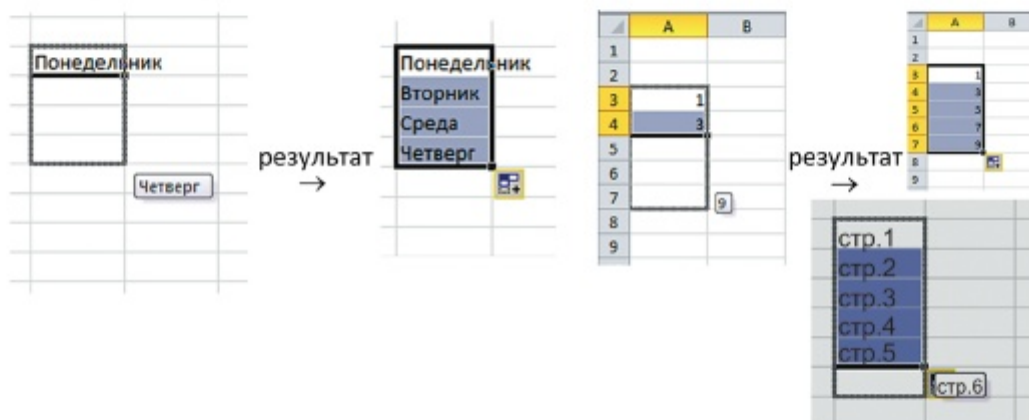


Рис. 3.16. Последовательности



Знание. Понимание



1. Что такое форматирование в электронной таблице?
2. Какой командой объединяются ячейки?
3. Можно ли в электронной таблице Excel написать текст по диагонали?
4. Как можно изменить ширину столбца или высоту ячейки?
5. Для чего используются границы?
6. Как можно изменить цвет шрифта?
7. Чем отличается таблица в текстовом процессоре от таблицы в электронной таблице?
8. Для чего применяется **Маркер заполнения**?

Применение



9. Установите соответствие между определением и названием в табл. 3.5 по электронной таблице Excel.



Таблица 3.5

Определение	Название
1. Это приложение, обрабатывающее информацию, которую можно представить в виде таблиц и автоматизировать обработку таких данных	A. Электронные таблицы или табличные процессоры
2. Это специальные пакеты программ, обрабатывающие информацию, которую можно представить в виде таблиц	B. Excel
3. Это файл, в котором на отдельных листах хранится информация (данные)	C. .xlsx
4. Это составная часть книги Excel, состоящая из строк и столбцов, на пересечении которых находятся ячейки	D. Рабочая книга Excel
5. Это расширение файлов, созданных в электронной таблице Excel	E. Ячейка
6. Это основная единица хранения данных, образуемая на пересечении столбца и строки рабочего листа	F. Рабочий лист

10. Создайте таблицу и отформатируйте данные по образцу (см. табл. 3.6).

Таблица 3.6

					Курс доллара:	
РАСЧЕТ ЗАРАБОТНОЙ ПЛАТЫ						
№ п/п	ФИО	Начислено	Налоги		К выдаче в тенге	Эквивалент, \$
			подох. 5%	пенс. 10%		
1	2	3	4	5	6	7
1		10 800				

Продолжение таблицы 3.6

1	2	3	4	5	6	7
2		105 000				
3		135 000				
Итого						
Максимальная						
Минимальная						
Средняя						



Анализ. Синтез



11. Отметьте, что относится к форматированию (табл. 3.7).

Таблица 3.7

<input type="checkbox"/>	Изменение ориентации текста в ячейке
<input type="checkbox"/>	Изменение цвета шрифта
<input type="checkbox"/>	Изменение содержимого ячейки
<input type="checkbox"/>	Выравнивание содержимого по правому краю ячейки
<input type="checkbox"/>	Изменение размера ячейки
<input type="checkbox"/>	Выделение ячеек
<input type="checkbox"/>	Установка и изменение границ
<input type="checkbox"/>	Изменение заливки ячейки
<input type="checkbox"/>	Изменение типа данных (число, дата, текст и т.п.)
<input type="checkbox"/>	Удаление каких-либо символов
<input type="checkbox"/>	Удаление информации
<input type="checkbox"/>	Изменение разрядности чисел



Оценивание



12. Проведите исследование таблицы «Список для подготовки к школе» (рис. 3.17). Дайте оценку стилю форматирования таблицы. Для чего предназначена данная таблица? Почему некоторые записи в ячейках зачеркнуты, а некоторые отмечены галочкой?

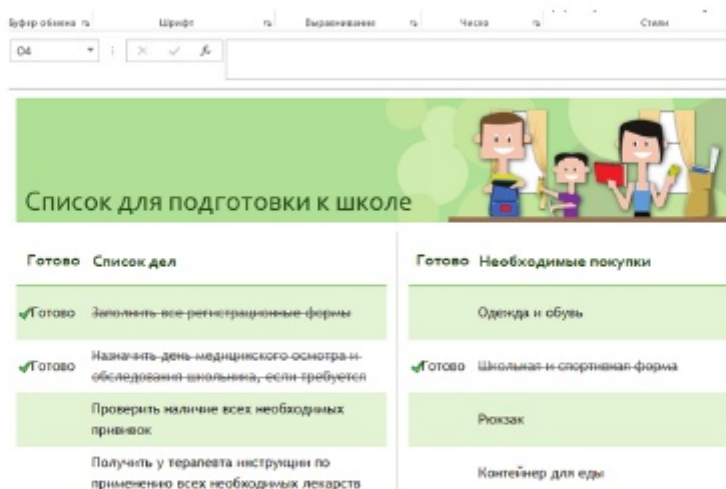


Рис. 3.17. Список для подготовки к школе

Конвертируем файлы Excel в Word



Способ 1. Ручное копирование

Один из самых простых способов конвертировать содержимое файла Excel в Word – это просто его скопировать и вставить данные.

1. Откройте файл в программе Microsoft Excel и выделите содержимое, которое нужно перенести в Word.

2. Скопируйте с помощью контекстного меню, щелкнув по пункту **Копировать** или с помощью сочетания клавиш **Ctrl + C**.

3. Запустите **Microsoft Word**. Нажмите правой кнопкой мыши по листу и в появившемся меню через параметры вставки выберите пункт **Сохранить условное форматирование**.

4. Скопированные данные будут вставлены.

Недостаток этого способа заключается в том, что не всегда преобразование выполняется корректно, особенно при наличии формул.

Способ 2. Сторонние программы

Одним из известных вариантов для преобразования документов из Excel в Word является приложение **Abex Excel to Word Converter**. Оно полностью сохраняет исходное форматирование данных и структуру таблиц при конвертировании, поддерживает пакетную конвертацию. Скачать Abex Excel to Word Converter можно с официального сайта (<https://www.softportal.com/software-31542-abex-excel-to-word-converter.html>).

3.3. ТИПЫ ДАННЫХ

ВЫ НАУЧИТЕСЬ

- Использовать различные типы данных в таблице;
- создавать формулы для вычислений в электронной таблице.

КЛЮЧЕВЫЕ СЛОВА

Форматы данных	Деректер форматтары	Data formats
Формат ячеек	Ұяшық пішімі	Cell format



Какие существуют **форматы типов данных** в электронной таблице Excel, а точнее – форматы ячеек, которые отображают данные в нужном нам виде? Давайте рассмотрим их.



Шаг за шагом

Для того чтобы указать формат отображения данных в ячейке, выделите ячейку или диапазон ячеек.

Щелкните по выделенной области правой кнопкой мыши и выберите из контекстного меню пункт **Формат ячеек**. В открывшемся окне перейдите на вкладку **Число** (рис. 3.18). Выберите интересующий формат и нажмите **ОК**.

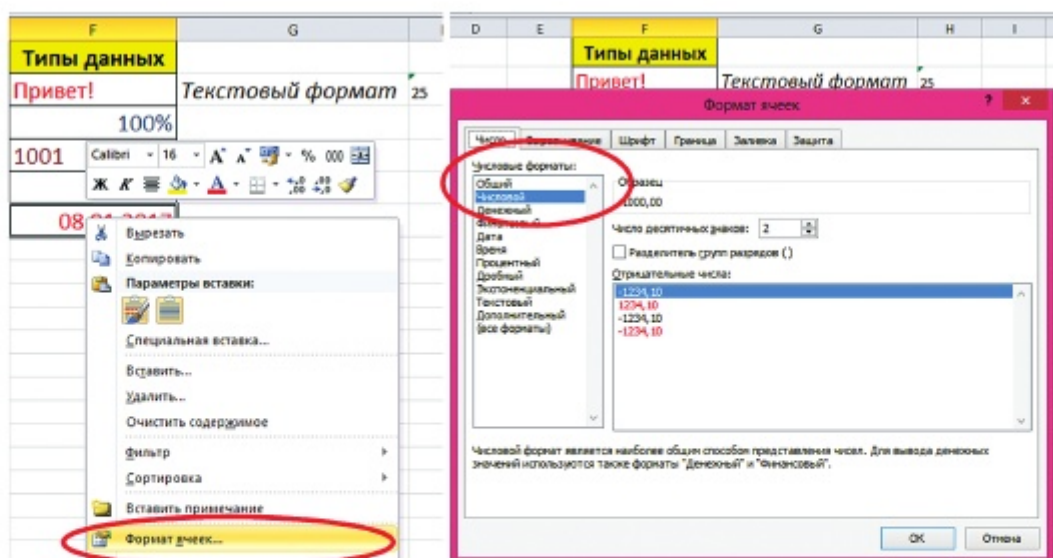



Рис. 3.18. Формат ячеек

Другой способ – это вызов из пункта выпадающего меню **Другие числовые форматы...** из вкладки **Главная** диалогового окна **Все форматы** (рис. 3.19, а). Или выполните с помощью окна **Формат ячеек** вкладки **Число**, которая вызывается командой **значка**  группы **Число** (рис. 3.19, б).

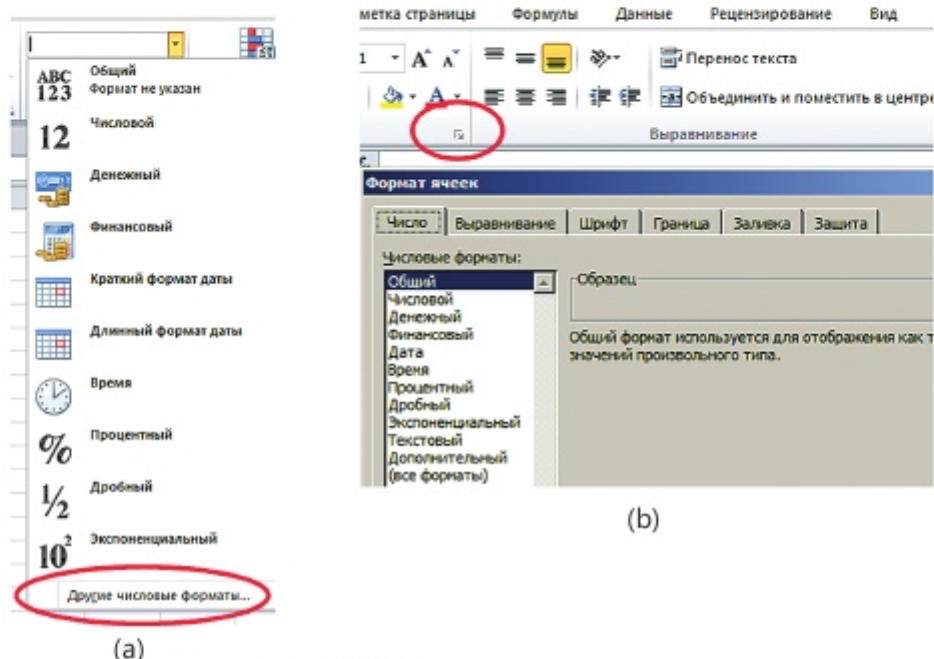


Рис. 3.19. Форматы типов данных

Классифицируем основные форматы данных (рис. 3.20).

- **Общий**. Данный формат задается по умолчанию. Изначально в новой книге все ячейки общего формата не определены;
- **Числовой** – формат представления целых или вещественных чисел. У данного формата есть возможность добавлять/убирать число десятичных знаков, ставить разделители групп разрядов;
- **Денежный** – формат представления чисел в денежных единицах;

	А	В	С	Д	Е	Ф
1	Числовой	Денежный	Дата	Процентный	Дробный	По маске
2	7712305,17	₮7 712 305,17	01.01.2017	100%	2/7	8 янв
3			02.01.17			

Рис. 3.20. Разные форматы данных

– **Дата** отображает число в формате дат. В Excel все даты представлены целыми числами. Например, дата **08.01.2020** в виде числа выглядит как **42743**, что означает количество дней отсчета дат в Excel от **01.01.1900**. Даты раньше **01.01.1900** в Excel отражаются как текст;

– **Процентный** – формат представления вещественного числа в виде процентов. Число **0** будет отображено как **0%**, **1** – как **100%**;

– **Дробный** формат отображает вещественное число в виде дроби;

– **(все форматы)**. С помощью этого варианта представления можно записать свой тип отображения числа. Например, представление даты **13.01.17** выглядит как **ДД.ММ.ГГ**, если изменить на **МММ.ГГ**, то дата будет представлена как **янв.13**.



Подумайте

Зависит ли форматирование в ячейке от типа данных, которые в ней находятся? Как отформатированы данные на рис. 3.21? Если в ячейке есть текст, то по какому краю он будет отформатирован? По какому краю будут выровнены числовые данные, другие форматы?

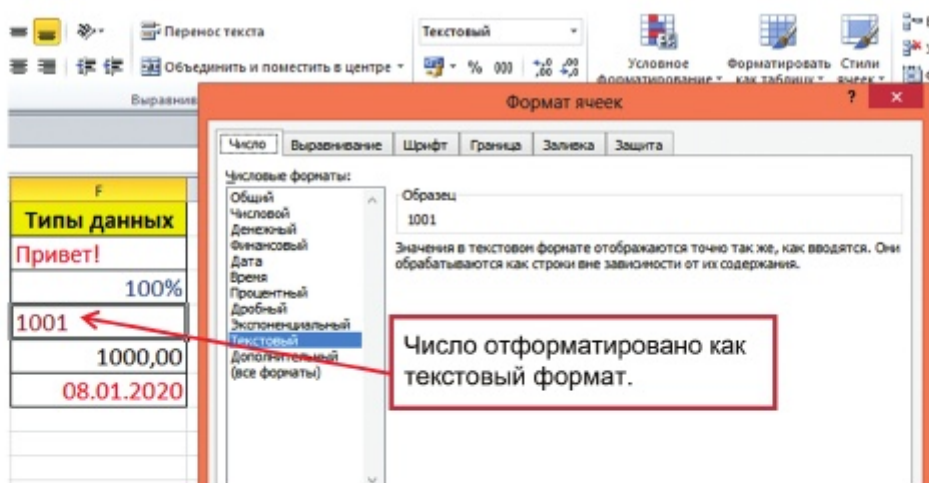


Рис. 3.21. Форматирование разных типов данных

При вводе данных в поле можно установить **автоматическую проверку** типа полей данных. Кроме того, можно установить дополнительные правила проверки вводимых данных, т.е. условие на значение, позволяющее обеспе-

чить ввод корректных данных в поле записи. Проверка значения осуществляется при переходе к другому полю или другой записи.

Если введенное значение не соответствует заданному условию, то на экран выводится сообщение об ошибке (рис. 3.22).

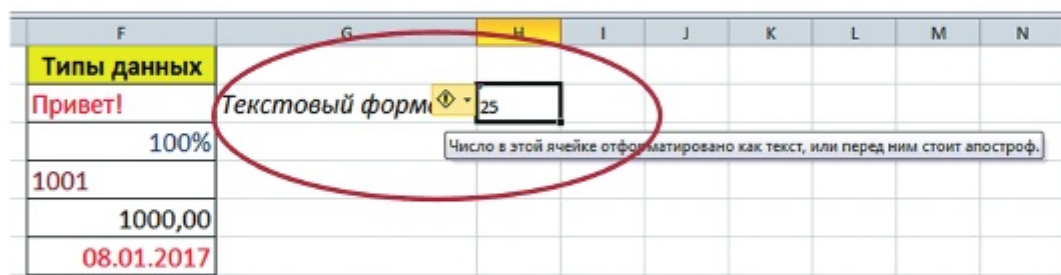


Рис. 3.22. Введенное значение не соответствует заданному условию

При расчетах в электронной таблице часто используют формулы.

Формулы представляют собой выражения, на основании которых в таблицах выполняются вычисления. После знака равенства «=» или «-», «+» следует **арифметическое выражение**, состоящее из цифр, адресов ячеек (ссылок) и знаков математических операций: «+» – сложение, «-» – вычитание, «/» – деление, «*» – умножение.

Вычисления сводятся к использованию формул и стандартных функций.

Формула – специальный инструмент в электронной таблице Excel, предназначенный для расчетов, вычислений и анализа данных.



Ввод формулы в ячейку начинается со знака «=» («равно»). Это делается для того, чтобы Excel понял, что в ячейку вводится именно формула, а не данные. Формула заканчивается нажатием клавиши **Enter**. Адреса ячеек в формулах используются тогда, когда нужно ввести числа, размещающиеся в этих ячейках.

Приоритет выполнения арифметических операций:

1	Действие в скобках	()
2	Возведение в степень	7^2
3	Умножение	*
4	Деление	/
5	Сложение	+
6	Вычитание	-

Например: $(90 - 2,5) * (2,66 - 0,06) / (1,075 + 10,3)$.



Шаг за шагом

Рассмотрим простой пример: деление двух чисел.

Выделите произвольную ячейку, например **A1**. В строке формул введите знак «=», затем введите выражение **100/5**. Нажмите **Enter**. В ячейке появится результат **20** (рис. 3.23). А в строке формул отобразится сама формула **=100/5**.

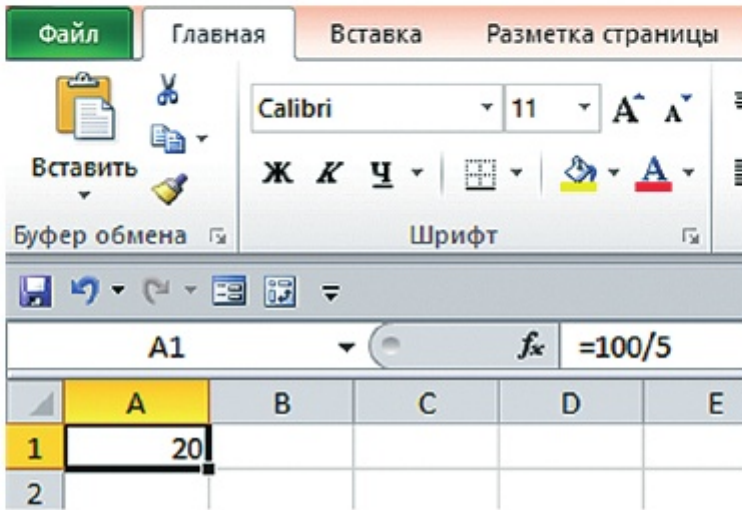


Рис. 3.23. Ввод формулы с указанными данными

Для исправления неправильно введенной формулы необходимо выделить ячейку, в которой находится формула.

Когда в строке формул отобразится введенная формула, нужно щелкнуть левой кнопкой мыши по ней и скорректировать ее.

Поработаем с различными арифметическими операторами: сложения (+), вычитания (-), умножения (*), деления (/). Чтобы их правильно использовать, надо помнить их приоритет (см. стр. 93).

Пусть в ячейках **A1**, **A2** находятся данные.

1. Выделите ячейку **A3** и введите в строке формул знак «=».
2. Щелкните на ячейке **A1**, чтобы в строке формул отобразился адрес ячейки **A1**, и введите знак сложения (+).
3. Щелкните на ячейке **A2**, поставьте скобки для указания приоритетности, умножьте на **4** и нажмите **Enter**. Получите результат **1200** (рис. 3.24).

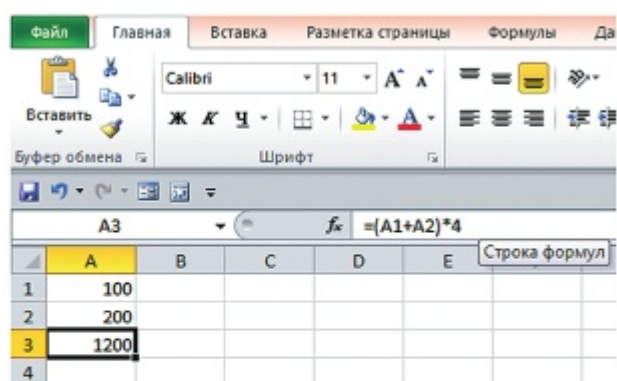


Рис. 3.24. Ввод формулы с ссылками на адреса ячеек

Ссылки в формуле выполняют роль адресов ячеек, содержимое которых используется в вычислениях. Ссылки бывают с **относительной (A1, B2, C5), смешанной (\$A1, C\$5) и абсолютной адресацией (\$B\$2)**. Диапазон обозначается **A1:C8, G1:H7** и т.д.

От метода адресации ссылок зависит, что будет с ними происходить при копировании формулы из одной ячейки в другую.

По умолчанию ссылки на ячейки в формулах рассматриваются как **относительные**. Это означает, что адреса в ссылках **изменяются**.

При **абсолютной адресации** адреса ссылок при копировании формулы **не изменяются**, так что ячейка, на которую указывает ссылка, рассматривается как **постоянная**. Для указания абсолютной адресации нажимаем на клавишу **F4**, вводится символ **\$**.

Например, формулу с помощью ссылок можно записать следующим образом: **= (A1 + B3) * 100% / C4**.

Знание. Понимание



1. В каких случаях применяются те или иные типы данных?
2. Когда применяется общий формат?
3. Как можно поменять формат данных?
4. Чем отличается форматирование числовых данных от текстовых? Как вы понимаете слово «формула»?
5. С какого знака начинаются формулы? Какие арифметические знаки используются в формулах?
6. Что такое относительная адресация?
7. Чем отличается относительная адресация от абсолютной?





Применение



8. Используя числовой формат для десятичных чисел, разделите число на группы разрядов, поставив галочку в области **Разделитель групп разрядов (,)**, например, число **125456,456** можно представить как **125 456,456**.
9. Установите для денежного формата число десятичных знаков и условное обозначение валют. Например, **р.** заменить на **Т казахстанский (тенге)**. Тогда число **56,89** будет представляться как **Т56,89**.



Анализ



10. Установите дробный формат для данных, представленных ниже, по образцу (табл. 3.8).

Таблица 3.8

Обычный формат	Формат дробный
5,6	5 3/5
30,3	30 3/10
25,5	25 1/2

11. В одну из ячеек листа должна вводиться дата рождения одного из учеников вашего класса (в формате **Дата**). Предусмотрите автоматическую проверку на правильность вводимых в эту ячейку значений.



Синтез



12. Текстовый формат позволяет отображать информацию в том виде, в каком она вносится в ячейку, без каких-либо автоматических преобразований в дату, формулу и т.д.

Для примера введите число **456** и отформатируйте как текст, затем – как число. Текст выравнивается по левому краю, а число – по правому.

456	<-текст
456	<-число

После установки типа данных в поле выровняйте значение поля во вкладке **Выравнивание** диалогового окна **Формат ячейки** (рис. 3.25).

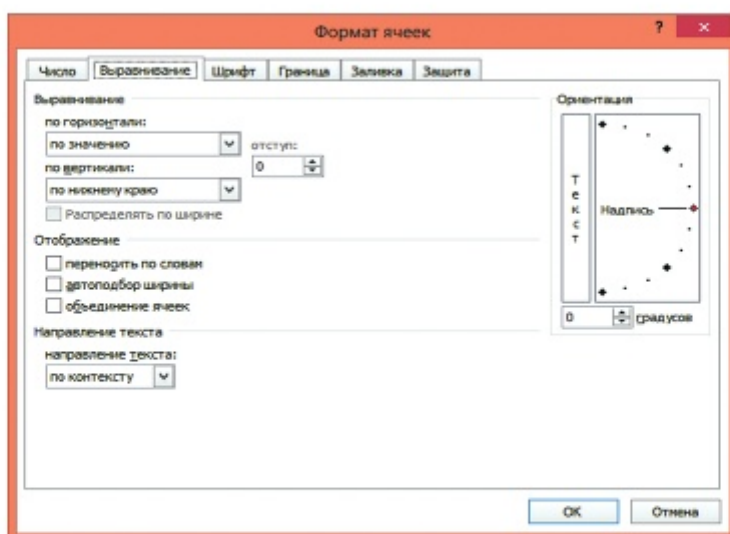


Рис. 3.25. Вкладка **Выравнивание**

Применяя вкладки **Шрифт**, **Граница**, **Заливка**, установите шрифт, границы, заливку ячеек таблицы и т.д.

Оценивание



13. В одну из ячеек листа должно вводиться время начала урока (в формате **Время**). Предусмотрите автоматическую проверку на правильность вводимых в эту ячейку значений, если известно, что первый урок начинается в 8 часов 30 минут, а последний заканчивается в 15 часов 25 минут. Продолжительность урока 40 минут.



14. Используя лист и данные, введенные при выполнении предыдущего задания, установите защиту ячейки. Примените вкладку **Защита** из диалогового окна **Формат** (рис. 3.26). От кого и для чего нужно защищать информацию в ячейках электронной таблицы?

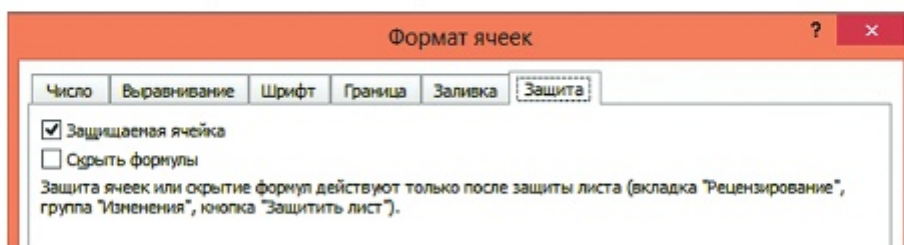


Рис. 3.26. Вкладка **Защита**



Выпадающий список в ячейке

Если в какую-либо ячейку предполагается ввод строго определенных значений из разрешенного набора (например, только «да» и «нет» или только из списка отделов компании и так далее), то это можно легко организовать при помощи выпадающего списка (рис. 3.27).

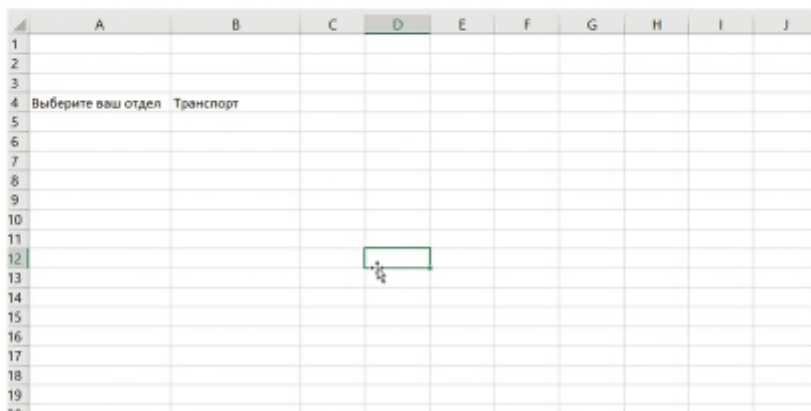


Рис. 3.27. Оформление выпадающего списка

1. Выделите ячейку (или диапазон ячеек), где должно быть такое ограничение.
2. Нажмите кнопку **Проверка данных** на вкладке **Данные (Data – Validation)**.
3. В выпадающем списке **Тип (Allow)** выберите вариант **Список (List)**.
4. В поле **Источник (Source)** задайте диапазон, содержащий эталонные варианты элементов, которые и будут впоследствии выпасть при вводе (рис. 3.28).

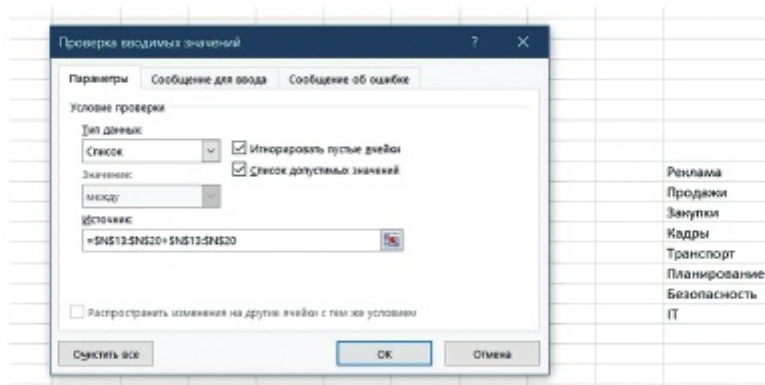


Рис. 3.28. Проверка вводимых значений

3.4.

УСЛОВНОЕ ФОРМАТИРОВАНИЕ В ЭЛЕКТРОННОЙ ТАБЛИЦЕ

ВЫ НАУЧИТЕСЬ

Использовать условное форматирование в электронной таблице.

КЛЮЧЕВЫЕ СЛОВА

Электронная таблица	Электрондық кесте	Spreadsheet
Форматирование	Форматтау	Formatting
Условное форматирование	Шартты пішімдеу	Conditional formatting

Условное форматирование – один из самых полезных инструментов в электронной таблице. Оно позволяет выполнять форматирование ячеек выборочно и автоматически, основываясь на их содержимом. Начнем изучение условного форматирования с проверки числовых значений **больше/меньше/равно/между** в сравнении с числовыми значениями.

Очень часто анализ данных подразумевает сравнение значений разных ячеек. В этом случае удобно ячейки, значения которых удовлетворяют определенным критериям, как-то выделить в документе. На рис. 3.29 показан рабочий документ со значениями ставок доходности по ценным бумагам по месяцам.

1	Условное форматирование на основе сравнения данных	
2		
3	месяц	ставка доходности (%)
4	январь	2,6
5	февраль	2,5
6	март	3,1
7	апрель	2,9
8	май	3,2
9	июнь	2,7
10	июль	2,4
11	август	2,3
12	сентябрь	2,7
13	октябрь	3,3
14	ноябрь	3,1
15	декабрь	2,9
16		

Рис. 3.29. Исходный документ с данными для условного форматирования

Зададим такое условное форматирование для ячеек **B4:B15**, чтобы те из ячеек, значения ставки доходности в которых меньше средней доходности **2,8**, выделялись светло-красным цветом, а текст – красным цветом.

Для этого выделим диапазон ячеек **B4:B15**. Для использования условного форматирования применяем кнопку **Условное форматирование** на панели **Стили** ленты **Главная**. Далее выбираем команду **Правила выделения ячеек** и **Меньше** (рис. 3.30).

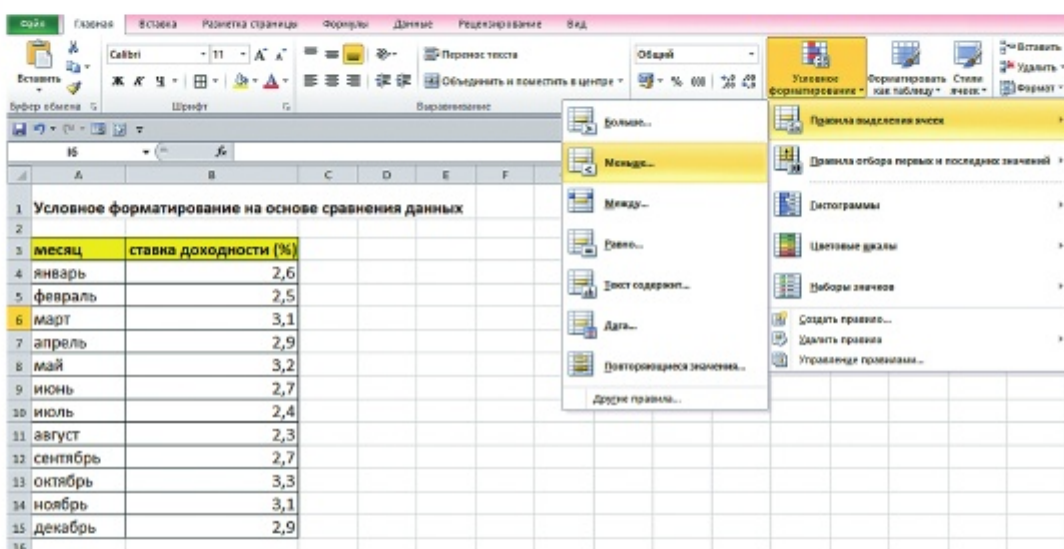


Рис. 3.30. Применение условного форматирования

Откроется диалоговое окно **Меньше...** настройки параметров условного форматирования. В поле **Форматировать ячейки, которые МЕНЬШЕ** указываем значение, с которым следует сравнивать форматлируемые ячейки (число или адрес ячейки). В данном случае указываем число **2,8**. В раскрывающемся списке справа выбираем один из predetermined способов форматирования (**Закраска цветом ячейки и текста**).

На рис. 3.31 видим, что текст отображается темно-красным цветом, а ячейка закрашена светло-красной заливкой.

The screenshot shows an Excel spreadsheet with the following data:

месяц	ставка доходности (%)
январь	2,6
февраль	2,5
март	3,1
апрель	2,9
май	3,2
июнь	2,7
июль	2,4
август	2,3
сентябрь	2,7
октябрь	3,3
ноябрь	3,1
декабрь	2,9

The dialog box 'меньше' is configured with the format: 'Светло-красная заливка и темно-красный текст'.

Рис. 3.31. Настройка условного форматирования

Это один из самых простых примеров выполнения условного форматирования.

Применение



Чтобы лучше понять, как работает условное форматирование, выделите группу ячеек с уже введенными данными, нажмите кнопку **Условное форматирование** и посмотрите различные варианты форматирования.

При выборе **Наборы значков** (5 цветных стрелок) **Условного форматирования** в ячейках рядом с числовыми данными будут отображаться стрелки с разным направлением (рис. 3.32).

По умолчанию программа автоматически определяет больше среднего значения, меньше среднего, равное среднему значению в выделенном диапазоне.

Для формирования нужных правил форматирования можно воспользоваться пунктом **Другие правила**.

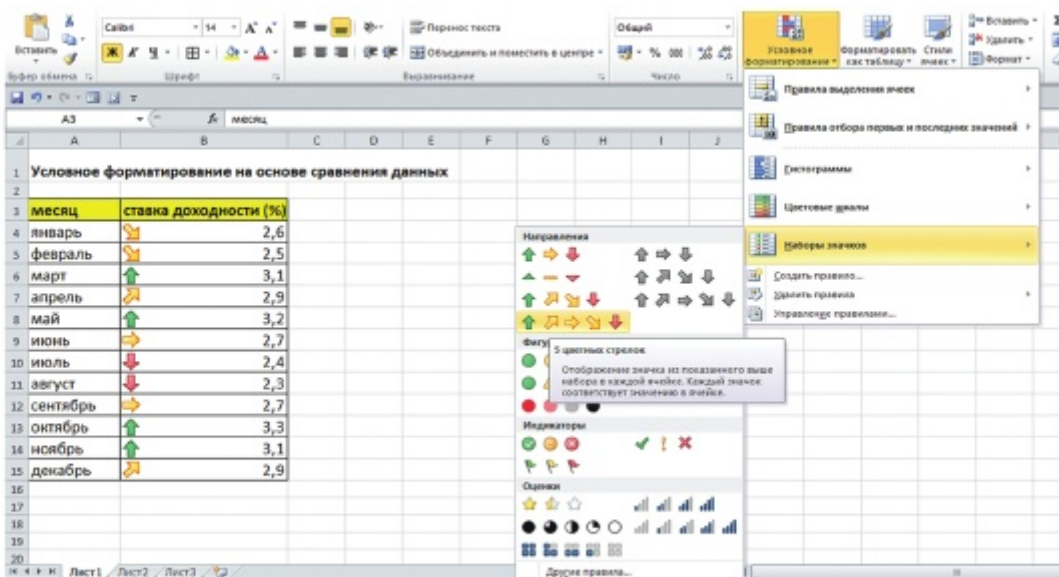


Рис. 3.32. Наборы значков

Если ячейка содержит текст, число или дату, то для применения условного форматирования необходимо выделить ячейку, нажать кнопку **Условное форматирование** и выбрать пункт **Правила отбора первых и последних значений**. Выберите пункт **Выше среднего...** (рис. 3.33, 3.34).

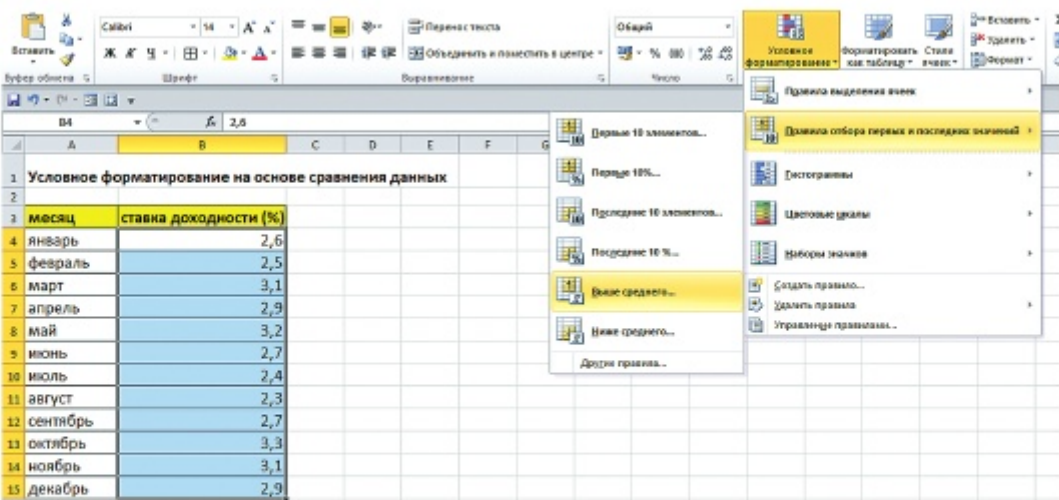


Рис. 3.33. Правила отбора первых и последних значений

Условное форматирование на основе сравнения данных

месяц	ставка доходности (%)
январь	2,6
февраль	2,5
март	3,1
апрель	2,9
май	3,2
июнь	2,7
июль	2,4
август	2,3
сентябрь	2,7
октябрь	3,3
ноябрь	3,1
декабрь	2,9

Выше среднего

Форматировать ячейки, которые находятся **ВЫШЕ СРЕДНЕГО**:
для выделенного диапазона с: Красный текст

OK Отмена

Рис. 3.34. Отбор **Выше среднего**

Знание. Понимание



1. С какой целью применяется условное форматирование?
2. Что нужно сделать с диапазоном ячеек, чтобы применить условное форматирование?
3. Каким образом можно вставить **Значки** в ячейки?
4. Какие еще способы применяются для условного форматирования?
5. Можно ли применить условное форматирование для определения ячеек с ошибочными записями?



Применение



6. Создайте таблицу «Прогноз погоды». Введите температуру воздуха на неделю (например, с 20 по 27 февраля). Задайте условное форматирование таким образом, чтобы все ячейки с отрицательными значениями закрашивались в красный цвет.



7. Составьте таблицу расходов за неделю текущего месяца в тенге (рис. 3.35).

Таблица расходов	
Месяц :	Год:
Дни недели	Расходы в тенге
понедельник	
вторник	
среда	
четверг	
пятница	
суббота	
воскресенье	

Рис. 3.35. Таблица расходов

Используя **Условное форматирование – Наборы значков**, выберите понравившийся значок и вставьте туда, где будут введены данные расходов. Что отображают 5 разных значков в ячейках?

8. В списке учащихся на рис. 3.36 ячейки с оценкой «5» залиты светло-красным цветом, «4» – желтым, «3» – зеленым, а «2» – красным цветом. Шрифт – пользовательский формат. Объясните, как была выполнена заливка и условное форматирование. Создайте таблицу по данному образцу.

Имя	четверти			
	1	2	3	4
Али	3	4	3	2
Адил	4	5	4	4
Аида	5	5	4	4
Айдар	3	4	4	3
Максим	5	4	5	4

Рис. 3.36. Условное форматирование оценок

Анализ. Синтез

9. Составьте таблицу и выделите с помощью условного форматирования диапазон значений температурных отклонений в вашем регионе.

На рис. 3.37 показан образец температурных данных с условным форматированием, в котором для обозначения высоких, средних и низких значений использовалась цветовая шкала.

	A	B	C	D
1	Температура в городе Алматы			
2	температура	март	апрель	май
3	высокая	15	20	25
4	средняя	8	7	20
5	низкая	4	5	10
6				

Рис. 3.37. Температурные отклонения

Умная таблица

Если выделить диапазон с данными и на вкладке **Главная** нажать **Форматировать как таблицу (Home – Format as Table)**, то список будет преобразован в умную таблицу, которая умеет много полезного.

1. Автоматически растягивается при дописывании к ней новых строк или столбцов.

2. Введенные формулы автоматически будут копироваться на весь столбец.

3. Шапка такой таблицы автоматически закрепляется при прокрутке, и в ней включаются кнопки фильтра для отбора и сортировки.

4. На появившейся вкладке **Конструктор (Design)** в такую таблицу можно добавить строку итогов с автоматическим вычислением (рис. 3.38).

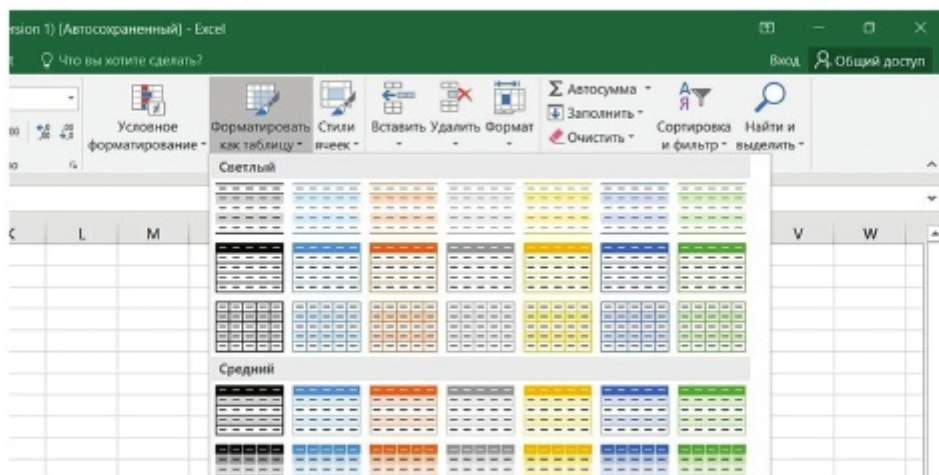


Рис. 3.38. Вкладка **Конструктор (Design)**

3.5.

ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ТАБЛИЧНЫХ ДАННЫХ

ВЫ НАУЧИТЕСЬ

Создавать диаграммы в электронной таблице.

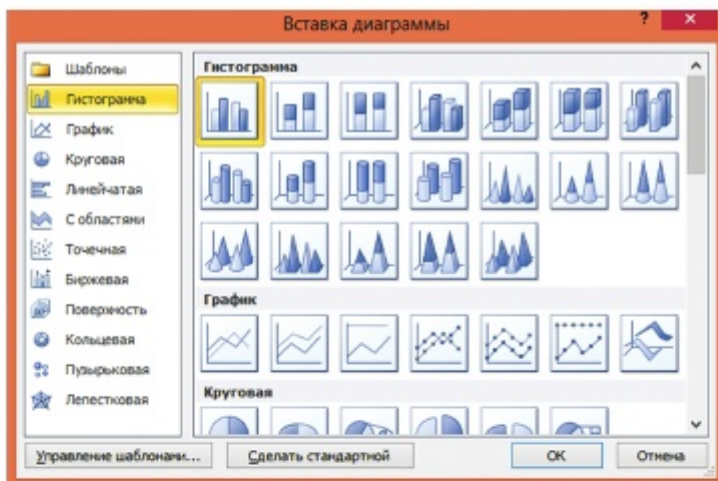
КЛЮЧЕВЫЕ СЛОВА

Диаграмма
Диаграмма
Diagram

Диаграммы – самый наглядный способ представления данных. Диаграмму качества знаний нагляднее показать графически, чем рассказать словами, что в классе из 25 человек 2 отличника, 12 хорошистов, а остальные – троечники. Диаграмма облегчает восприятие информации.



Диаграмма – это визуальное представление числовых значений.



Диаграммы бывают различного **типа** (рис. 3.39):

- гистограмма;
- круговая;
- линейчатая;
- с областями;
- график;
- точечная;
- кольцевая;
- лепестковая;
- пузырьковая.

Рис. 3.39. Типы диаграмм

Каждый тип диаграммы подразделяется на **виды**. Например, гистограмма с группировкой показывает сравнение значений по категориям, гистограмма с накоплением – вклад каждого значения в общую сумму (с нарастающим итогом), нормированная гистограмма с накоплением – процент вклада каждого значения в общую сумму. Кроме того, можно выбрать форму элементов гистограммы: плоская (обычная), объемная, цилиндрическая, коническая, пирамидальная.

Шаг за шагом



Рассмотрим **технологии построения диаграмм** на примере проведения графического анализа недельной продажи товара (газет) тремя продавцами.

Для этого построим диаграмму типа **Гистограмма** обычного вида с отображением значений различных категорий.

1. Создайте таблицу с данными, как на рис. 3.40.

	A	B	C	D	E	F	G	H
1	Продажа газет							
2		Пн	Вт	Ср	Чт	Пт	Сб	Вс
3	Продавец 1	34	35	33	30	36	39	42
4	Продавец 2	28	29	21	24	26	24	24
5	Продавец 3	25	22	24	23	25	29	30

Рис. 3.40. Таблица продажи газет

2. Выделите область **A2:H5** и выберите гистограмму вида **с группировкой** из группы **Диаграммы** на вкладке **Вставка** (рис. 3.41).

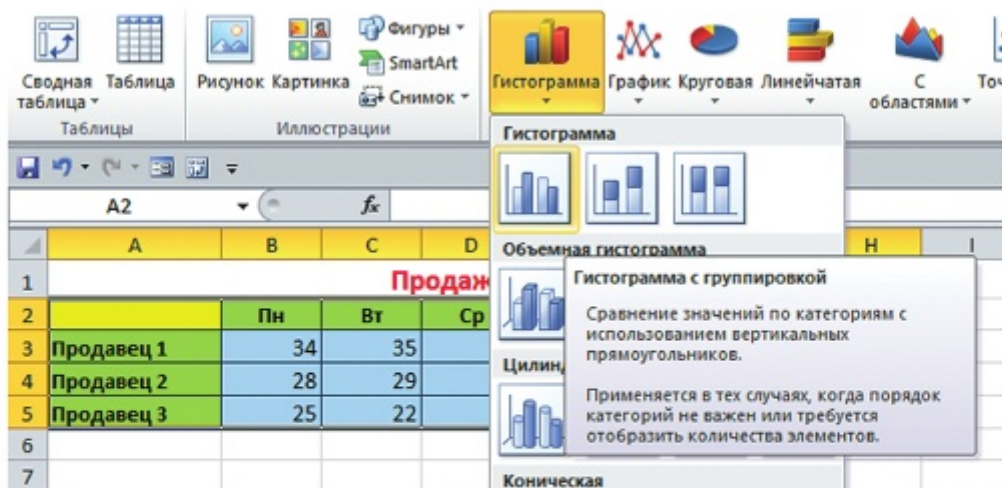


Рис. 3.41. Вставка диаграммы

3. Получите гистограмму, как на рис. 3.42.

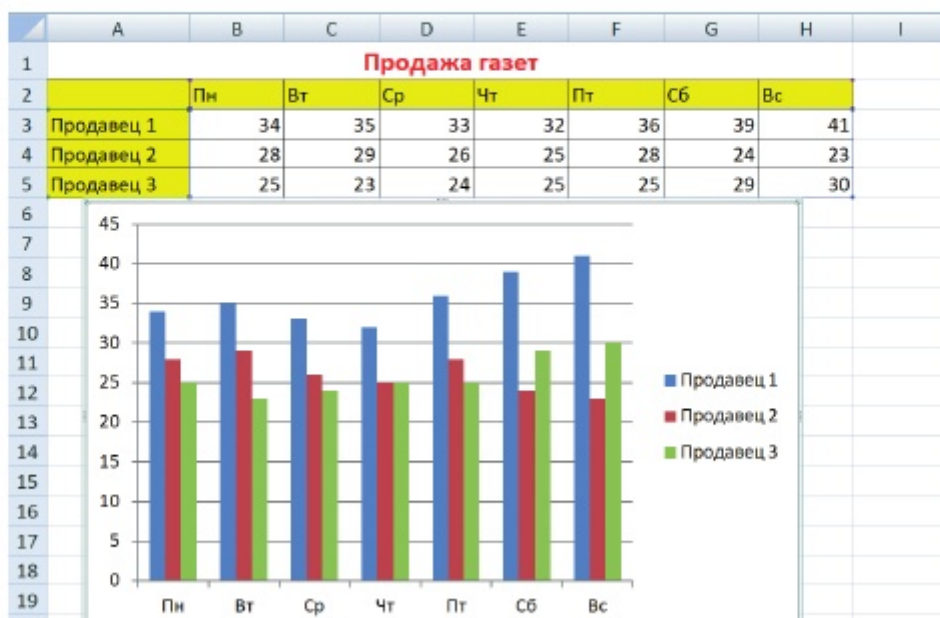
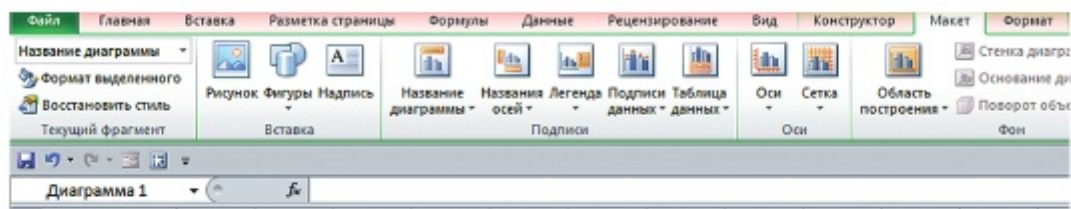


Рис. 3.42. Гистограмма

При выделении диаграммы появятся дополнительные вкладки для работы с диаграммой: **Конструктор** и **Макет** (рис. 3.43).

Рис. 3.43. Дополнительная вкладка **Макет**

Диаграмме необходимо задать разные параметры – это заголовок, оси, линии сетки, легенда, подписи данных, таблица данных для визуализации числовых данных.

Область диаграммы, где расположены условные обозначения (цвет и подписи **Продавец №**), называется **Легенда**.

Попробуйте с помощью вкладки **Конструктор** поменять стиль, тип диаграммы и переместить диаграмму на отдельный лист.

С помощью вкладки **Макет** вставьте название диаграммы, название осей, переместите легенду в другое место, установите подписи данных (рис. 3.44).

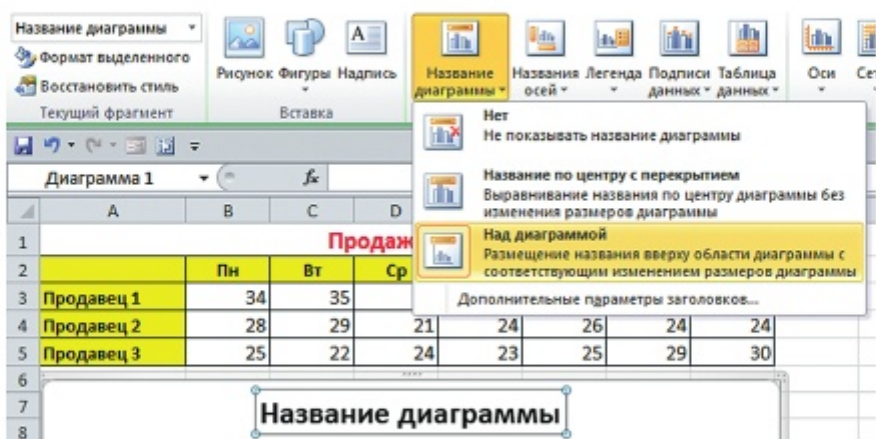


Рис. 3.44. Вставка названия диаграммы

Если выделить двойным щелчком левой кнопкой мыши что-либо на диаграмме, например ось, то можно с помощью команды **Формат оси** изменить ее параметры (рис. 3.45). Попробуйте изменить параметры оси.

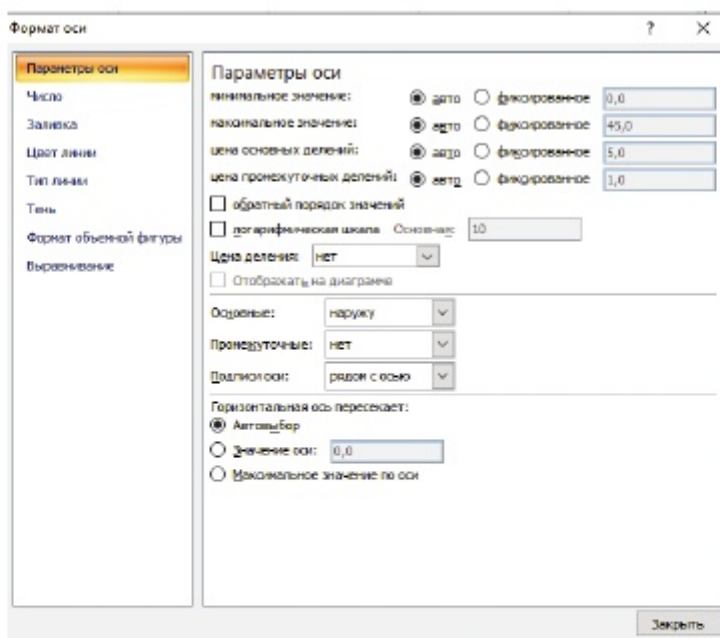


Рис. 3.45. Формат оси

В результате получите диаграмму, которая показывает, сколько товара продавал каждый продавец ежедневно (рис. 3.46).

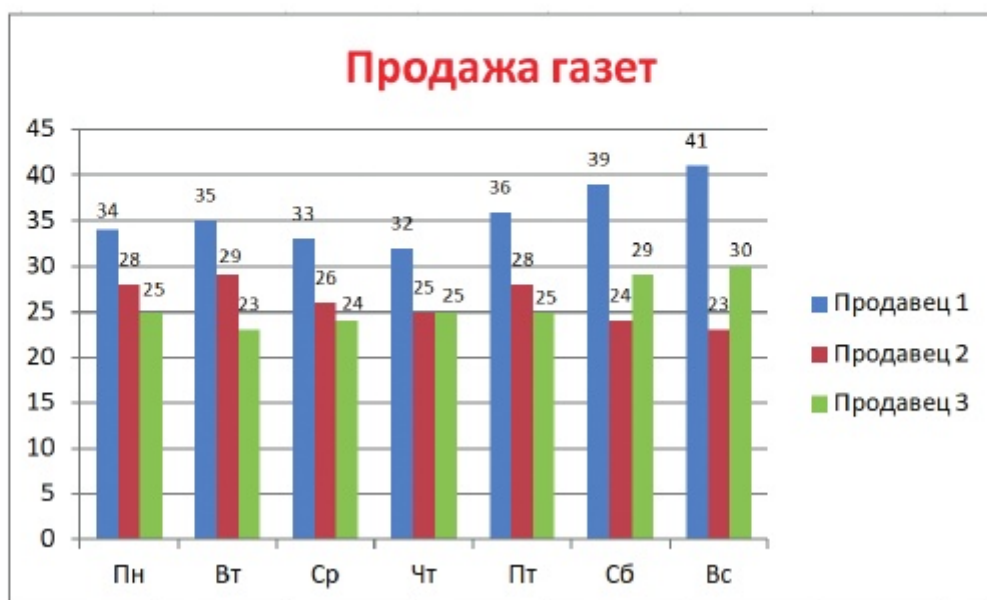


Рис. 3.46. Готовая диаграмма



Знание. Понимание



1. Какие типы диаграмм существуют?
2. Почему у одной диаграммы имеется несколько видов?
3. Можно ли поменять подписи данных у готовой диаграммы? Каким образом?
4. Как можно поменять тип диаграммы?
5. Какие дополнительные вкладки появляются в электронной таблице при выделении диаграммы?
6. Как настроить диаграмму, чтобы она могла максимально визуализировать данные?
7. Как удалить диаграмму?
8. Объясните, какие виды графиков изображены на рис. 3.47. Почему изображены два графика в одной координатной плоскости? Какой график наиболее информативный?

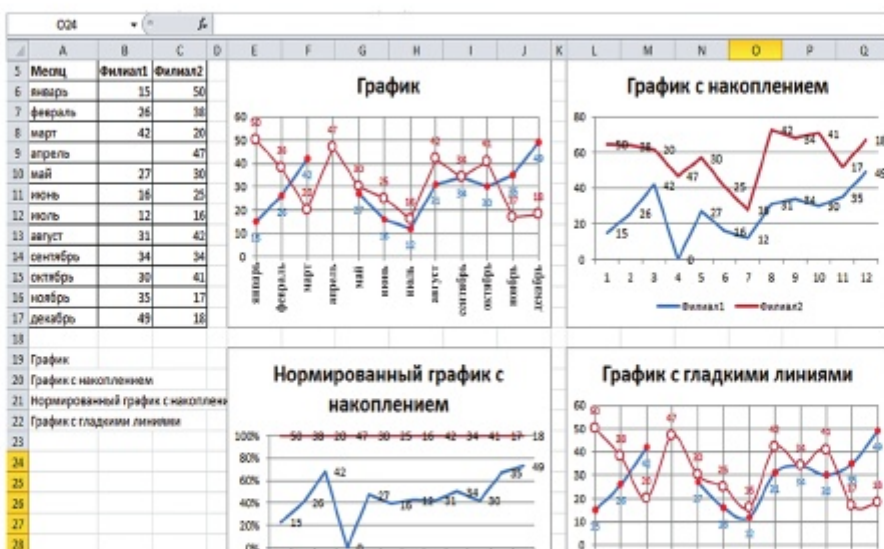


Рис. 3.47. Разные виды графиков

Применение



9. Номера 1, 2, 3, 4, 5, 6, 7 на рис. 3.48 указывают на названия элементов диаграммы. Запишите их и объясните, с какой целью эти элементы диаграммы используются.

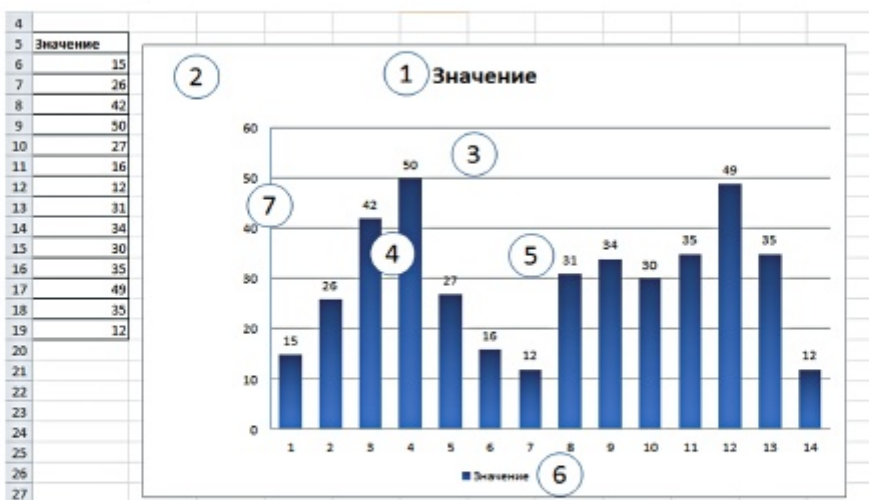


Рис. 3.48. Элементы диаграммы

10. На рис. 3.49 представлена динамика популярности поискового запроса «пневмония» в поисковой системе Google за 2020 год. На основе графических данных создайте таблицу. Выделите с помощью условного форматирования максимальное значение запросов в Казахстане и России.

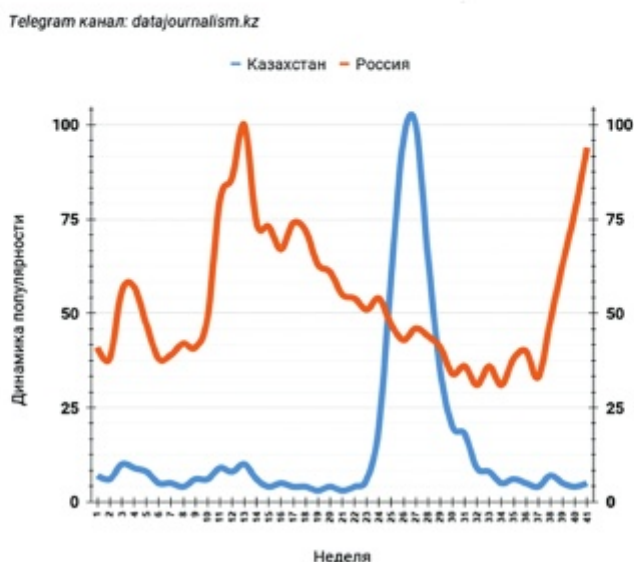


Рис. 3.49. Динамика популярности поискового запроса «пневмония» в поисковой системе Google за 2020 год



Анализ



11. Постройте круговую диаграмму по данным из табл. 3.9.

Таблица 3.9

	А	В
1	Площадь континентов	
2	Название континента	Площадь (млн. кв. км)
3	Европа	10,00
4	Азия	43,40
5	Северная Америка	24,71
6	Южная Америка	17,84
7	Австралия	7,66
8	Антарктида	14,10
9	Африка	30,30
10	Всего:	148,01

Синтез

12. Постройте круговую диаграмму распределения людей по группам крови, если людей с группой крови 0 в мире около 46%, с группой крови А – около 34%, с группой крови В приблизительно 16,5%, а с самой редкой группой крови АВ всего 3,5%.

**Оценивание**

13. Проблема экологии городов – это, в первую очередь, проблема уменьшения выбросов в окружающую среду различных загрязнителей. Для разложения в природной среде бумаги требуется до 10 лет, консервной банки – до 90 лет, фильтра от сигареты – 100 лет, полиэтиленового пакета – до 200 лет, пластмассы – до 500 лет, стекла – до 1000 лет. Помните об этом, не бросайте куда попало полиэтиленовые пакеты или бутылки. Создайте электронную таблицу Excel и заполните ее указанными выше данными. Постройте 3 различные диаграммы и выберите наиболее информативную.



Если при завершении работы в MS Excel забыли сохранить изменения в файле, то восстановить файл можно следующим образом.



В Excel 2010 для восстановления данных нажмите **Файл – Последние (File – Recent)** и найдите в правом нижнем углу экрана кнопку **Восстановить несохранённые книги (Recover Unsaved Workbooks)**.

В Excel 2013 путь: **Файл – Сведения – Управление версиями – Восстановить несохранённые книги (File – Properties – Recover Unsaved Workbooks)**.

Во всех последующих версиях Excel для восстановления нужно открыть **Файл – Сведения – Управление книгой**. И в появившемся меню выбрать нужный пункт.

3.6.

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

ВЫ НАУЧИТЕСЬ

Моделировать процессы в электронных таблицах.

КЛЮЧЕВЫЕ СЛОВА

Моделирование
Модельдеу
Modeling

В своей деятельности человек очень часто использует модели, т.е. он создает образ того объекта, явления или процесса, с которым ему предстоит иметь дело.

Существуют разные виды моделирования: информационное, математическое, компьютерное, натурное, физическое, структурное и т.д. Любая модель создается и изменяется на основе имеющейся у человека информации. Модель создается для какого-либо объекта, процесса или явления. В этом случае говорят, что человек создает информационную модель объекта.



Информационная модель – это описание моделируемого объекта на одном из языков кодирования информации.

Прототип – это объект, процесс или явление, на основе которого создается модель.

Информационная модель содержит существенные для исследователя сведения об объекте.



Моделирование – это процесс построения моделей для исследования и изучения объектов, процессов или явлений.

Создание моделей для исследования экономических, экологических, биологических, физических и других процессов является актуальной задачей в моделировании процессов. Практически во всех науках о природе, живой и неживой, об обществе построение и использование моделей является мощным орудием познания.

В настоящее время для построения различных по форме моделей используется компьютер. Поэтому нас интересует, как создать компьютерную модель.

На рис. 3.50 представлена схема этапов моделирования при исследовании различных процессов.

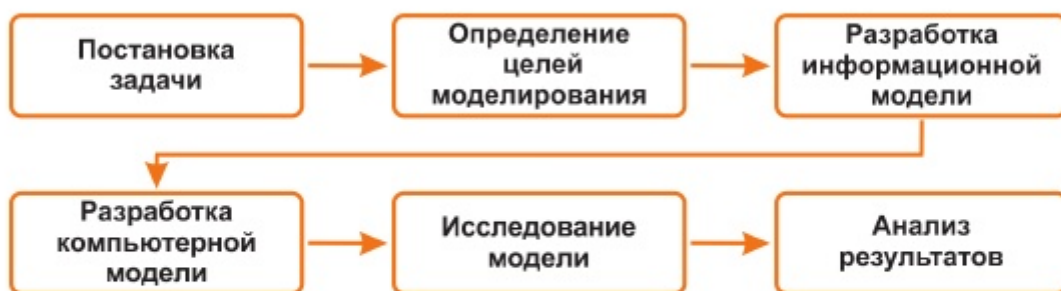


Рис. 3.50. Этапы моделирования

В качестве примера рассмотрим моделирование экологических процессов, связанных с загрязнением окружающей среды, в электронной таблице Excel.

Шаг за шагом



1. Постановка задачи.

В результате сброса промышленных стоков возрос уровень загрязнения реки. Каким он будет через сутки, двое, трое и т.д.? Когда он достигнет допустимой нормы, если известно, что за сутки он уменьшается в определенное количество раз?

Проведем исследование экологической модели при значениях параметров, представленных в табл. 3.10.

Таблица 3.10

Вещество	C_0 (мг/л)	$C_{доп}$ (мг/л)	K
Свинец	5	0,03	1,12
Мышьяк	1,5	0,05	1,05
Фтор	0,2	0,05	1,01

Условные обозначения:

C_0 – начальная концентрация вредных примесей;

$C_{доп}$ – предельно допустимая концентрация вредных примесей;

K – коэффициент суточного уменьшения концентрации вредных примесей.

2. Определение целей модели.

Объект исследования – концентрация вредных веществ. Для построения таблицы значений используем электронную таблицу Excel.

3. Разработка модели.

А) Построение математической модели:

– определение аргументов, результатов и связи между ними;

– определение математической модели задачи:

$$C_1 = C_0/K$$

$$C_2 = C_1/K = C_0/K^2$$

$$C_3 = C_2/K = C_0/K^3$$

.....

$$C_n = C_0/K^n \text{ или } C_n = C_{n-1}/K;$$

– определение целого значения n .

В формуле очередное значение C_i вычисляется по предыдущему значению C_{i-1} .

В) Построение компьютерной модели (рис. 3.51):

C12		fx	
	A	B	C
1	Экологическая модель		
2	Исходные данные		
3	5	C ₀ - начальная концентрация вредных примесей	
4	0,03	C _д - допустимая концентрация	
5	1,12	K - коэффициент суточного уменьшения концентрации	
6	Расчетная таблица		
7	n	K	C₀
8	0	1,12	5,00
9			
10	=A8+1	=B8	=C8/B9
51	43	1,12	0,04
52	44	1,12	0,03

Рис. 3.51. Экологическая модель в электронной таблице

- представьте модель в электронной таблице;
- введите последовательно исходные данные, а также начальные данные и формулы для последующих вычислений;
- получите таблицу, с помощью которой произведите вычисления, скопировав формулы с помощью **Маркера заполнения** до нужного результата;
- проведите компьютерное тестирование модели, изменяя исходные данные веществ: **свинец, мышьяк, фтор**;
- установите **Формат ячеек** для числовых данных, установив **число десятичных знаков – 2 после запятой**;
- выполните форматирование элементов таблицы;
- используйте условное форматирование, закрасив ячейку и число уровня загрязненности при достижении допустимой нормы в другой цвет;
- постройте график функции по данным столбца начальной концентрации вредных примесей (рис. 3.52).

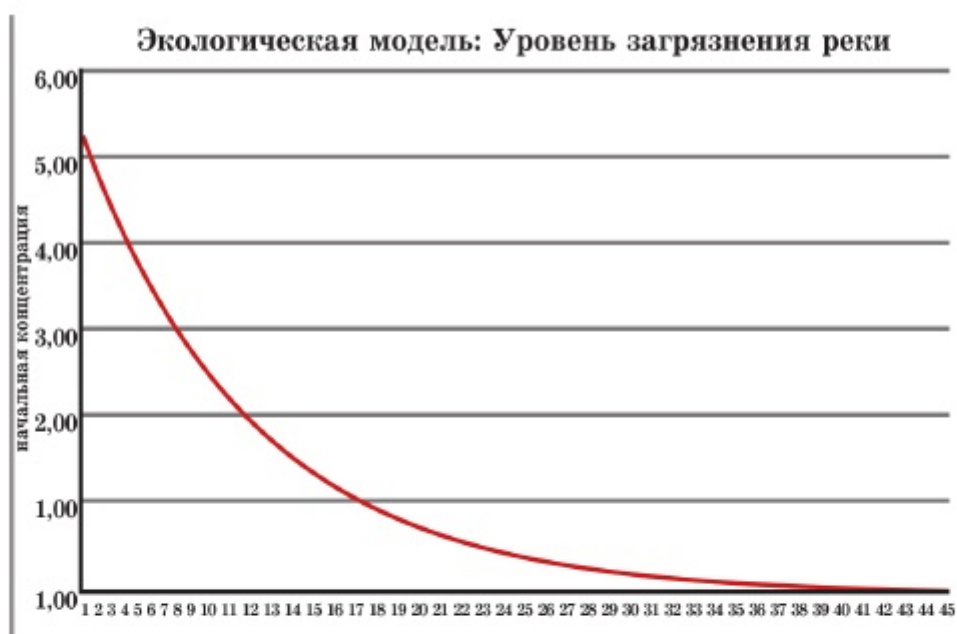


Рис. 3.52. График функции экологической модели

4. Исследование модели.

Модель построена в соответствии с формулами, представленными в условии задачи.

5. Анализ результатов.

Нужно сравнить значения столбца **C** со значениями ячейки **A4** допустимой нормой $C_{\text{доп}}$. При этом в ячейке должны получить значение C_i . Это значение должно быть меньше или равно значению ячейки **A4**.

Если такая ячейка существует, то определяем соответствующее значение ячейки **A1**. Это будет номер суток. Если такой ячейки не окажется, следует продолжить копирование ячеек до получения нужного результата.

Экологическая интерпретация полученного решения: через 44 дня уровень загрязненности достигнет допустимой нормы.



Знание. Понимание

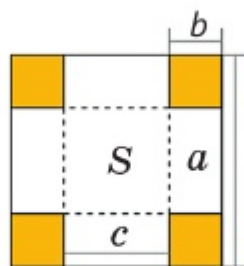


1. Что такое моделирование?
2. Из каких этапов состоит моделирование?
3. Чем отличается математическая модель от компьютерной?
4. Что такое тестирование модели? Как проводится компьютерное тестирование?
5. Почему необходимо проводить анализ результатов?
6. Для чего может быть предназначена каждая из моделей:
 - эскизы костюмов к театральному спектаклю;
 - макет книги или журнала;
 - глобус;
 - географический атлас;
 - модель (макет) строения молекулы воды;
 - уравнение химической реакции;
 - генеалогическое (родословное) древо семьи;
 - макет скелета человека;
 - формула определения площади квадрата со стороной h : $S = h^2$;
 - расписание движения поездов;
 - игрушечная модель автомобиля;
 - схема метрополитена;
 - график зависимости высоты тела, брошенного под углом к горизонту, от времени полета;
 - оглавление книги?

Применение. Анализ



7. Дана геометрическая модель (рис. 3.53), у которой длина стороны картонного листа обозначена a , длина выреза – b . Вычислите объем геометрической модели.



Математическая модель

Расчетные формулы:

$c = a - 2b$ – длина стороны дна;

$S = c^2$ – площадь дна;

$V = Sb$ – объем.

Рис. 3.53.

Геометрическая модель

8. Для моделирования используйте среду электронной таблицы, в которой информационная и математическая модели объединяются в таблицу, имеющую три области:

- исходные данные – управляемые параметры;
- промежуточные расчеты;
- результаты.

Задание

- 1) Заполните область исходных данных по предложенному образцу (рис. 3.54).
- 2) Составьте таблицу расчета со столбцами в электронной таблице, используя расчетные данные математической модели.
- 3) Какие типы данных будут использоваться?
- 4) Выполните форматирование данных в ячейках.
- 5) Как можно отобразить данные из таблицы графически?

Задача о склеивании коробки

Исходные данные	
Длина стороны	40
Шаг увеличения выреза	1

Расчет

Промежуточные расчеты			Результаты
Длина выреза	Длина стороны дна	Площадь дна	Объем
1			

Рис. 3.54. Образец в электронной таблице**Синтез**

9. Необходимо покрасить стены кухни. Сколько потребуется банок краски, если известно, что:

- размеры кухни $405 \times 310 \times 285 \text{ см}^3$;
- 88% площади стен занимает кафельная плитка;
- 1 банка краски предназначена для покраски площади 5 м^2 ?

**Оценивание**

10. Вспомните, как был открыт закон всемирного тяготения. Легенда гласит, что Исаак Ньютон сидел под яблоней, когда на него упало яблоко, подсказавшее ему правильное решение.

Как известно, закон всемирного тяготения говорит о том, что сила притяжения двух тел прямо пропорциональна их массам и обратно пропорциональна квадрату расстояния между ними:

$$F = G \cdot \frac{m_1 \cdot m_2}{r^2}$$

- 1) Является ли данная формула математической моделью такого физического явления, как тяготение?
- 2) Что нужно сделать, чтобы она стала адекватной моделью притяжения Земли и висящего на дереве яблока?



Электронные таблицы позволяют создавать диаграммы нескольких типов, основными из которых являются графики, круговые диаграммы, гистограммы (рис. 3.55).

Графики используются для отображения зависимости значений одной величины (функции) от другой (аргумента). Они позволяют отслеживать динамику изменения данных.

Круговые диаграммы используются для отображения величин (размеров) частей некоторого целого; в них каждая часть целого представляется как сектор круга, угловой размер которого прямо пропорционален величине (размеру) части.

Гистограммы (столбчатые диаграммы) используются для сравнения нескольких величин. В них величины отображаются в виде вертикальных или горизонтальных столбцов. Высоты (длины) столбцов соответствуют отображаемым значениям величин.



Рис. 3.55, а. Виды диаграмм: график



Рис. 3.55, б. Виды диаграмм: диаграмма



Рис. 3.55, с. Виды диаграмм: гистограмма

ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ III

Из предложенных альтернативных ответов выберите только один верный. Каждый правильный ответ оценивается в 1 балл. Всего 15 баллов.

1. Текстовый редактор – это:

- а) программа для работы преимущественно с текстовой информацией;
- б) программа для создания мультфильмов;
- в) программа для обработки изображений;
- г) программа для управления ресурсами ПК при создании рисунков;
- д) художник-график.

2. Операция, нарушающая признак, по которому подобраны все остальные операции из приводимого ниже списка.

- а) Сохранение текста.
- б) Форматирование текста.
- в) Перемещение фрагмента текста.
- г) Удаление фрагмента текста.
- д) Копирование фрагмента текста.

3. Продолжите логическую цепочку:

Библиотека – каталог; Книга – оглавление; Текстовый редактор – ?:

- а) окно;
- б) текст;
- в) меню;
- г) курсор;
- д) рабочее поле.

4. Электронная таблица – это:

- а) прикладная программа, предназначенная для обработки структурированных в виде таблицы данных;
- б) прикладная программа обработки кодовых таблиц;
- в) устройство персонального компьютера, управляющее процессом обработки данных в табличной форме;
- г) системная программа, управляющая ресурсами персонального компьютера при обработке таблиц;
- д) экранная форма представления записи базы данных.

5. Диапазон в электронной таблице – это:

- а) все ячейки одной строки;
- б) все ячейки одного столбца;
- в) множество допустимых значений;

г) совокупность клеток, образующих в таблице область прямоугольной формы;

д) область таблицы произвольной формы.

6. Количество ячеек электронной таблицы в диапазоне A2:B4.

а) 16.

б) 2.

в) 8.

г) 4.

д) 6.

7. Электронная таблица представляет собой:

а) совокупность нумерованных строк и поименованных с использованием букв латинского алфавита столбцов;

б) совокупность поименованных с использованием букв латинского алфавита строк и нумерованных столбцов;

в) совокупность пронумерованных строк и столбцов;

г) совокупность строк и столбцов, именуемых пользователем произвольным образом.

8. Строки электронной таблицы:

а) именуется пользователем произвольным образом;

б) обозначаются буквами русского алфавита А ... Я;

в) обозначаются буквами латинского алфавита;

г) нумеруются.

9. Столбцы электронной таблицы:

а) обозначаются буквами латинского алфавита;

б) нумеруются;

в) обозначаются буквами русского алфавита А ... Я;

г) именуется пользователем произвольным образом.

10. Для пользователя ячейка электронной таблицы идентифицируется:

а) путем последовательного указания имени столбца и номера строки, на пересечении которых располагается ячейка;

б) адресом машинного слова оперативной памяти, отведенного под ячейку;

в) специальным кодовым словом;

г) именем, произвольно задаваемым пользователем.

11. Форма графического представления числовых значений, позволяющая облегчить восприятие и интерпретацию числовых данных, называется:

а) чертежом;

- б) блок-схемой;
- в) картой;
- г) таблицей;
- д) диаграммой.

12. Диаграмма, отдельные значения которой представлены точками в декартовой системе координат, называется:

- а) гистограммой;
- б) линейчатой диаграммой;
- в) круговой диаграммой;
- г) объемной диаграммой;
- д) точечной диаграммой.

13. Гистограмма – это диаграмма:

- а) в которой отдельные значения представлены вертикальными столбцами различной высоты;
- б) из параллелепипедов, размещенных вдоль оси X;
- в) в которой используется система координат с тремя координатными осями, что позволяет получить эффект пространственного представления рядов данных;
- г) в которой отдельные значения представлены полосами различной длины, расположенными горизонтально вдоль оси X;
- д) представленная в виде круга, разбитого на секторы.

14. Моделирование – это процесс:

- а) замены реального объекта (процесса, явления) его образом, создаваемым для решения конкретной задачи средствами языка в широком смысле слова;
- б) демонстрации моделей одежды в салоне мод;
- в) неформальной постановки конкретной задачи;
- г) замены реального объекта (процесса, явления) другим материальным или идеальным объектом, отражающим все без исключения признаки данного объекта;
- д) выявления существенных признаков рассматриваемого объекта.

15. К информационным моделям, описывающим организацию учебного процесса в школе, можно отнести:

- а) классный журнал;
- б) расписание уроков;
- в) список учащихся школы;
- г) перечень учебников;
- д) список наглядных учебных пособий.

4.1. РАБОТА С ФАЙЛАМИ

ВЫ НАУЧИТЕСЬ

Осуществлять чтение и запись файла на языке программирования Python (пайтон).

КЛЮЧЕВЫЕ СЛОВА

Файл	Файл	File
Текстовый файл	Мәтіндік файл	Text file
Двоичный файл	Екілік файл	Binary file



До сих пор мы работали с данными, которые сохранялись в оперативной памяти компьютера. Но как известно, после завершения работы программы все действия с данными удаляются. Возникает вопрос: «Как сохранять данные?». Для удобства работы данные хранятся в виде файлов на запоминающих устройствах.

Что такое файл? Для чего используют файл?



Файл – это определенное количество информации (программа или данные), имеющее имя и хранящееся в долговременной (внешней) памяти.



При вводе большого количества данных в программу, а также при выводе результата их обработки удобно использовать файловый ввод и вывод. В программировании различают файлы двух типов (рис. 4.1).



Рис. 4.1. Виды файлов

Рассмотрим работу с текстовыми файлами. Для успешного взаимодействия программы и файла существуют определенные этапы работы (рис. 4.2).

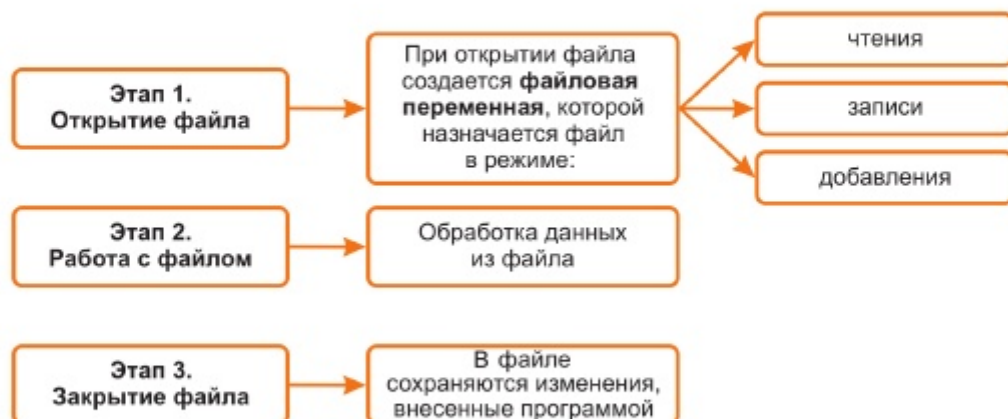


Рис. 4.2. Этапы работы с файлами

Для открытия файла можно выбрать следующие виды режима:

- **'r' (read)** – открытие на чтение (является значением по умолчанию);
- **'w' (write)** – открытие на запись. Содержимое файла удаляется, если файла не существует, создается новый;
- **'a' (append)** – открытие на дозапись: информация добавляется в конец файла.

Рассмотрим более подробно, как осуществляется чтение и запись файла на языке программирования Python.

Этап 1. На рис. 4.3 представлена общая схема **открытия файла** с помощью функции **open()**.

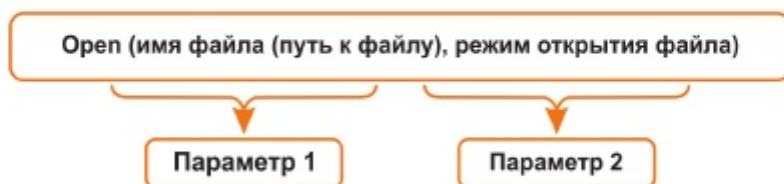
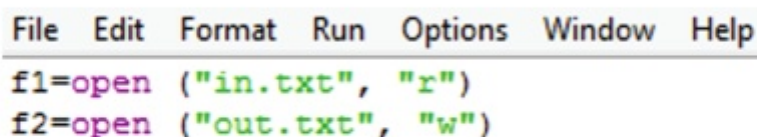


Рис. 4.3. Общая схема открытия файла

Первый параметр может быть относительным. В этом случае Python автоматически осуществляет поиск в папке относительно запущенного скрипта для **in.txt** (рис. 4.4).



```
File Edit Format Run Options Window Help
f1=open("in.txt", "r")
f2=open("out.txt", "w")
```

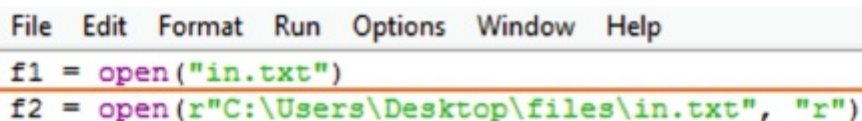
Рис. 4.4. Открытие файлов для чтения (**in.txt**) и записи (**out.txt**) с относительным параметром

Здесь:

- **f1, f2** – файловые переменные для операций;
- **in, out** – имя файла на диске (+ путь...);
- **r, w** – режимы работы файла.

Если файл с относительным параметром не будет найден, то генерируется сообщение об ошибке **FileNotFoundError**.

Абсолютный параметр представлен в виде полного пути к файлу, который может начинаться с буквы диска, как на рис. 4.5.



```
File Edit Format Run Options Window Help
f1 = open("in.txt")
f2 = open(r"C:\Users\Desktop\files\in.txt", "r")
```

Рис. 4.5. Открытие файла с абсолютным параметром

Здесь **in** – **имя файла** на диске с указанием полного пути к нему. Необходимо обратить внимание, что параметр начинается с **r**. Это указывает в среде программирования Python, чтобы строка была обработана как исходная.

Этап 2. Обработка данных из файла осуществляется двумя способами:

- 1) с помощью метода **read()** – чтение всей информации до конца файла. Метод **read(n)** – чтение из файла **n** следующих символов;
- 2) метод **readline()** – чтение из файла построчно (рис. 4.6).

```
File Edit Format Run Options Window Help
f1=open ("in.txt", "r")
z=f1.readline()
```

Рис. 4.6. Обработка данных методом `readline()`

Метод `write()` – запись в файл.

Необходимо помнить, что вся информация, которая хранится в файле, воспринимается программой как строковая. **Строковые типы данных** – это объекты, состоящие из последовательности символьных данных.



Для работы с числами необходимо их преобразовывать с помощью функций `int(x)` или `float(x)`. А при выводе результата в файл их снова нужно преобразовывать в строку `str(x)`.

Этап 3. После завершения работы с файлом (рис. 4.7) его необходимо закрыть методом `close()`, чтобы освободить используемые ресурсы, связанные с файлом.

```
File Edit Format Run Options Window Help
F = open ( "in.txt", "r" )
x= F.readline()
F.close()|
```

Рис. 4.7. Завершение работы с файлом



Даны два числа во входном файле `in.txt`. Сложите их и выведите результат в выходном файле `out.txt`.



Шаг за шагом

1. Создайте на **Рабочем столе** папку с именем `files`.
2. В папке `files` создайте текстовый документ с именем `in` и расширением `txt` в приложении **Блокнот**.

3. В текстовый файл **in.txt** введите построчно два числа (рис. 4.8).

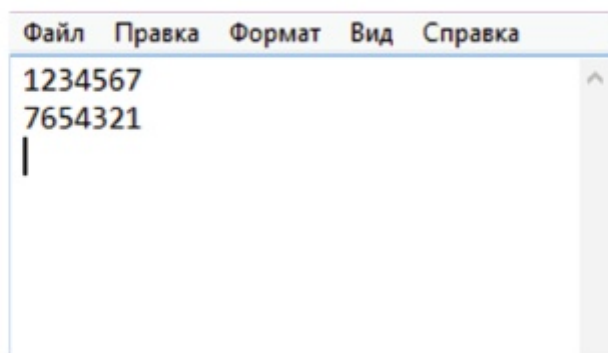


Рис. 4.8. Ввод двух чисел в приложении **Блокнот**

4. В меню **IDLE Python** выберите **File – New File**. Появится окно текстового редактора, в котором наберите команды на языке Python.

5. Сохраните файл программы в созданной папке **files** на **Рабочем столе**.

6. Переменной **f1** присвойте команду открытия файла в режиме чтения:

```
f1=open("in.txt","r")
```

7. Переменной **f2** присвойте команду открытия файла в режиме записи:

```
f2=open("out.txt","w")
```

8. Считайте первую строку данных из файла **f1**:

```
x=f1.readline()
```

9. Считайте вторую строку данных из файла **f1**:

```
y=f1.readline()
```

10. Преобразуйте переменные текстового типа в целочисленный тип данных:

```
x=int(x)
```

```
y=int(y)
```

11. Сложите полученные числа:

```
c=x+y
```

12. Преобразуйте числовую переменную **c** в строковую:

```
c=str(c)
```

13. Запишите результат в файл **f2**:

```
f2.write(c)
```

14. Закройте оба файла:

f1.close()

f2.close()

15. Запустите программу, выполнив на вкладке **Run** команду **Run Module (F5)**. Откройте файл **out.txt** из папки **files**, в котором должен появиться результат суммы двух чисел (рис. 4.9).

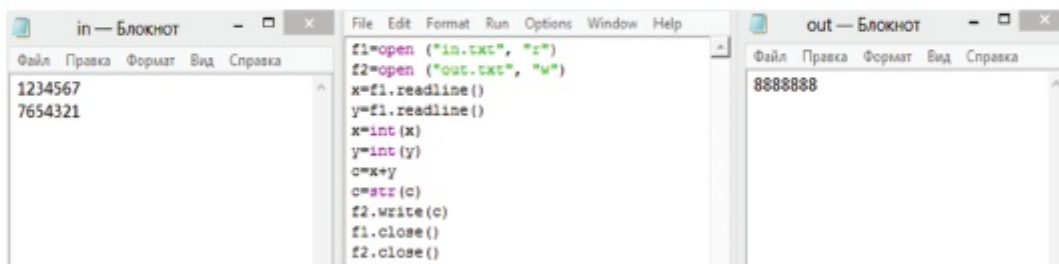


Рис. 4.9. Результат суммы в файле **out.txt**



Знание. Понимание



1. Что такое файл?
2. Произнесите ключевые слова темы на трех языках.
3. Почему в программировании используют работу с файлами?
4. Какие виды файлов существуют в программировании?
5. Перечислите этапы работы с файлами.
6. Какие виды режима открытия файла вам известны?
7. В чем различие методов **read()** и **readline()**?
8. Какой командой необходимо закрывать файл?



Применение. Анализ



9. Посмотрите код программы (рис. 4.10), проанализируйте и найдите ошибки.

```
File Edit Format Run Options Window Help
f1=open("in.txt","r")
x=f1.readline()
y=f1.readline()
x=int(x)
y=int(y)
c=x+y
f2.write(c)
f2.close()
```

Рис. 4.10. Код программы

10. Сравните методы **read()** и **readline()**. Составьте программы, показывающие их различия.
11. В файле **in.txt** содержатся два числа. Найдите их среднее арифметическое и выведите результат в тот же файл.

Синтез. Оценивание



12. Создайте программу, считывающую из файла **f1** двухзначное число и выводящую в файл **f2** число десятков и единиц этого числа. Оцените полученный результат.
13. Напишите эссе «Всё о файлах».



Коротко о файлах



Сохранение и считывание файлов с диска

```
f = open("fil.txt", "w", encoding = "utf8"),
```

где **f** – файловая переменная для операций;

fil.txt – имя файла на диске (+ путь);

w – режим работы ("r" – read, "w" – write, "a" – append...);

utf8 – кодировка символов в текстовых файлах: utf8 ascii cp1251 ...

Запись

```
f.write("hello")
```

текстовый файл → чтение/запись только **строк**, преобразуйте требуемые типы

Чтение

```
s = f.read(4)
```

где **s** – пустая строка в конце файла,
(4) – количество символов для чтения, если количество символов не указано, прочитает весь файл.

```
s = f.readline()
```

прочитать следующую строку

```
f.close()
```

не забывайте закрывать после использования

4.2. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ВЕТВЛЕНИЯ

ВЫ НАУЧИТЕСЬ

Записывать разветвляющиеся алгоритмы на языке программирования Python (пайтон).

КЛЮЧЕВЫЕ СЛОВА

Ветвление	Тармақтау	Branching
Условие	Шарт	Condition
Истина	Ақиқат	True
Ложь	Жалған	False



В шестом классе вы познакомились с линейными алгоритмами. Но не все алгоритмы являются линейными. Часто в жизни встречаются такие ситуации, когда необходимо выполнить те или иные действия в зависимости от выбора одного условия. Например, съесть одну конфету или две (рис. 4.11)? В таких случаях используется **алгоритмическая конструкция ветвления**.



Рис. 4.11. Выбор одного условия

Приведите примеры ветвления из окружающей среды или из вашей жизни.



Ветвление – это алгоритмическая конструкция, в которой в зависимости от некоторого условия происходит исполнение одной из двух последовательностей действий (ветвей).



Разветвляющийся алгоритм – это алгоритм, содержащий конструкцию ветвления.

В зависимости от количества веток ветвление бывает **полным** и **неполным**. При наличии веток «Да» и «Нет» ветвление считается **полным** (рис. 4.12). При отсутствии ветки «Нет» ветвление называют **неполным** (рис. 4.13).

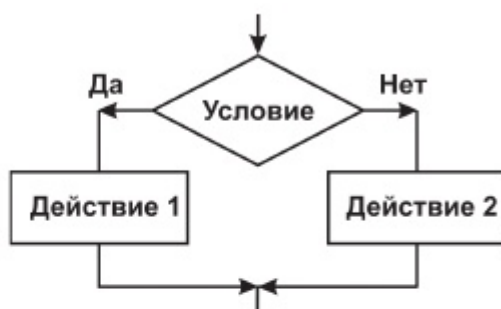


Рис. 4.12. Блок-схема полного ветвления

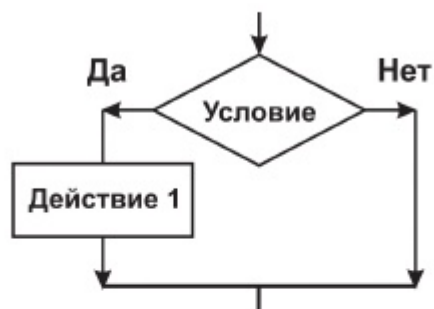


Рис. 4.13. Блок-схема неполного ветвления

Условие – это **логическое выражение**, которое в конечном итоге получается «Истина» (True) или «Ложь» (False). Логическое выражение в Python образуется с помощью логических операторов (табл. 4.1) для сравнения переменных с числами, строками или с другими переменными.

Таблица 4.1. Логические операторы

Операция сравнения	Логические операторы в Python	Название
>	>	Оператор «больше чем»
<	<	Оператор «меньше чем»
=	==	Оператор «равно»
≠	!=	Оператор «не равно»
≥	>=	Оператор «больше или равно»
≤	<=	Оператор «меньше или равно»

Конструкция ветвления в Python выглядит следующим образом (рис. 4.14).

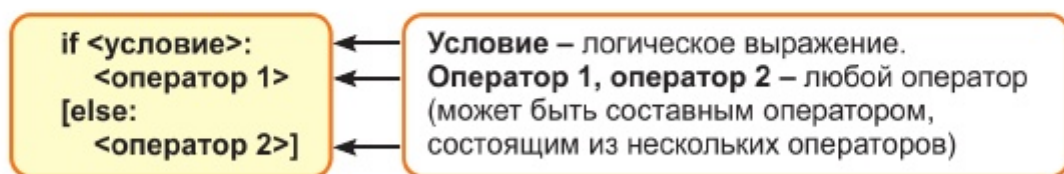


Рис. 4.14. Конструкция ветвления if

На рисунке 4.15 показан принцип работы команды ветвления.

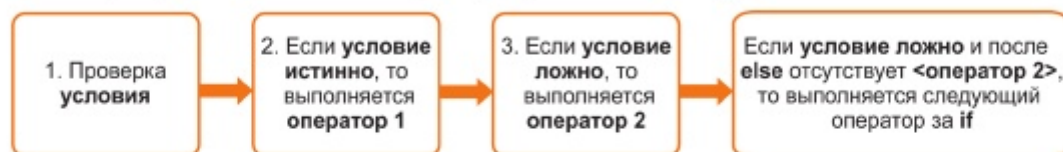


Рис. 4.15. Принцип работы оператора ветвления

Если ветвление неполное, то служебное слово **else** не используется. А после условия используется **двоеточие (:)**. При написании кода программы с ветвлением блок после команды **if** всегда дается с отступом вправо на **четыре пробела**. Либо можно использовать клавишу табуляции **Tab** на клавиатуре.

Остальной код программы и служебное слово **else** будут иметь такой же отступ, как у команды **if**. Рассмотрим пример программы с использованием ветвления.

Введите с клавиатуры некоторое число. Определите, является ли введенное число положительным. Выведите сообщение об этом в виде **Число положительное**, если условие истинное, иначе **Число неположительное** (рис. 4.16).



Шаг за шагом

1. Начало программы.
2. Введите некоторое число **a**.
3. Проверьте условие:
 $a > 0$?
Если **$a > 0$** , то выводится сообщение **Число положительное**, иначе **Число неположительное**.
4. Конец программы.

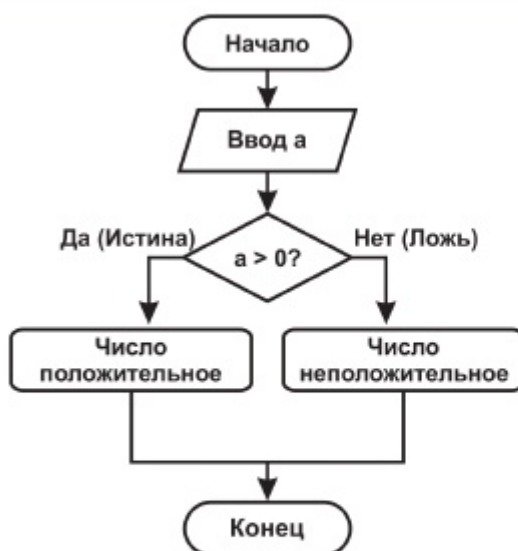
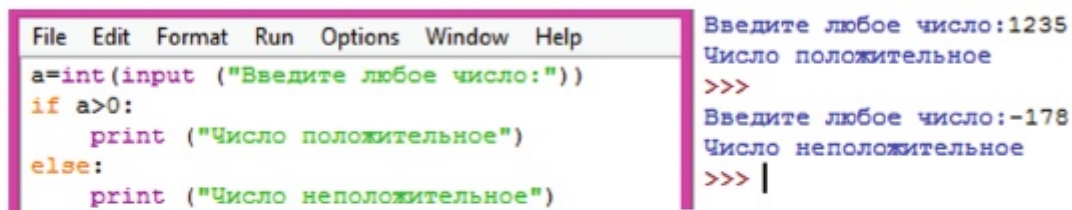


Рис. 4.16. Блок-схема решения задачи



5. По блок-схеме составьте и проверьте код программы в среде программирования Python (4.17).



```
File Edit Format Run Options Window Help
a=int(input ("Введите любое число:"))
if a>0:
    print ("Число положительное")
else:
    print ("Число неположительное")
```

```
Введите любое число:1235
Число положительное
>>>
Введите любое число:-178
Число неположительное
>>> |
```

Рис. 4.17. Код программы и результаты ее выполнения



Знание. Понимание



1. Произнесите на трех языках ключевые слова.
2. Что такое ветвление?
3. Что такое разветвляющийся алгоритм?



4. Какие виды ветвлений существуют?
5. Как записывается конструкция ветвления в Python?
6. Расскажите принцип работы оператора ветвления.
7. Какие операции сравнения вам известны?
8. Зачем используются отступы у команд, находящихся внутри ветвления?



Применение. Анализ



9. Имангали и Максим поспорили между собой, кто выше ростом. Учитель дал им словесный алгоритм и сказал, чтобы они написали программу в Python, которая будет определять наибольший рост.

Алгоритм:

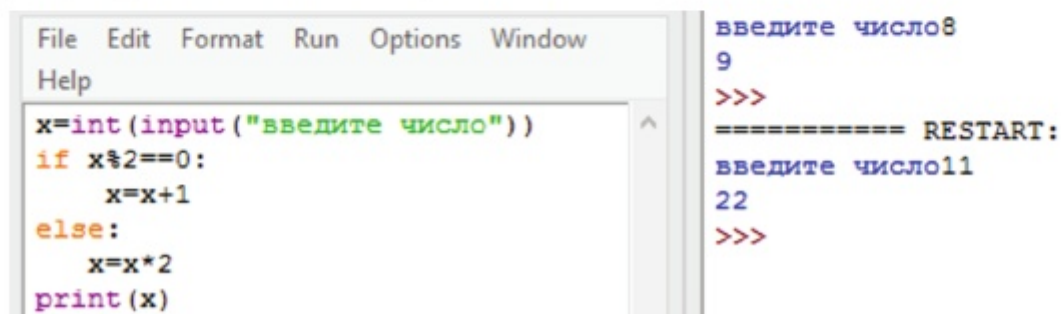
- 1) Введите число **a**.
- 2) Введите число **b**.
- 3) Сравните два числа (**a > b?**).
 - 3.1. Если условие истинно, то нужно присвоить переменной **m** значение переменной **a**, то есть **m = a**.
 - 3.2. Иначе присвоить переменной **m** значение переменной **b**, **m = b**.

4) На экране должен быть выведен результат, как на рис. 4.18.

```
Введите рост Имангали в сантиметрах=157
Введите рост Максима в сантиметрах=135
Высокий рост равен 157
>>> |
```

Рис. 4.18. Результат программы

10. На рис. 4.19 дан код программы и его результат. Напишите формулировку задачи и словесный алгоритм.



```
File Edit Format Run Options Window
Help
x=int(input("введите число"))
if x%2==0:
    x=x+1
else:
    x=x*2
print(x)

введите число8
9
>>>
===== RESTART:
введите число11
22
>>>
```

Рис. 4.19. Код и результат программы в Python

11. Составьте программу, которая определяет, является ли данное число делителем числа **3**.

12. Определите, кратно ли введенное число **5**.

13. Даны два числа. Если первое число больше второго, то замените его их разностью, в противном случае – суммой. Второе число не изменять.

14. Составьте программу для вычисления значений функции:

$$y = \begin{cases} x^2, & \text{если } x > 0 \\ -x, & \text{если } x < 0 \end{cases}$$

Синтез. Оценивание



15. Составьте программу, проверяющую условие «Могут ли три введенных числа быть длинами сторон треугольника?».

16. Напишите эссе на тему «Ветвление в нашей жизни».





Программа «Вход по паролю» с ветвлением

Как написать программу, которая при входе запрашивает пароль? Если пароль не совпадает с заданным значением, то программа выдает сообщение о том, что пароль неверный (рис. 4.20). Иначе, если пароль совпадает с заданным значением, то программа приветствует участника (рис. 4.21). При использовании строкового типа данных его заключают в апострофы (').

<pre>File Edit Format Run Options Window Help password=input('Привет! Введите пароль\n') if password=='user': print('Привет, user!') else: print('Простите, но Ваш пароль неверный')</pre>	<pre>===== RESTART: C:/User Привет! Введите пароль 12345 Простите, но Ваш пароль неверный >>> </pre>
--	--

Рис. 4.20. Пароль не совпадает

<pre>File Edit Format Run Options Window Help password=input('Привет! Введите пароль\n') if password=='user': print('Привет, user!') else: print('Простите, но Ваш пароль неверный')</pre>	<pre>===== RESTART: Привет! Введите пароль user Привет, user! >>> >>> </pre>
--	---

Рис. 4.21. Пароль совпадает

4.3. ПРОГРАММИРОВАНИЕ ВЛОЖЕННЫХ УСЛОВИЙ

ВЫ НАУЧИТЕСЬ

Использовать вложенные условия на языке программирования Python (пайтон).

КЛЮЧЕВЫЕ СЛОВА

Вложенные условия	Кірістірілген шарттар	Nested conditions
Операторы сравнения	Салыстыру операторлары	Comparison operators
Каскадные ветвления	Каскадты тармақталу	Cascading branches



Вам известно, что с помощью разветвляющихся алгоритмов можно реализовать как полный, так и неполный алгоритм ветвления. Если возникает необходимость сделать выбор больше, чем из двух путей, то в таком случае реализуется **множественное ветвление**. Приведем пример из жизни.

Допустим, вам дали задание, которое рекомендует просмотр видеоконтента в Интернете. Просмотр видеоконтента рекомендуется проводить в зависимости от возраста пользователя по годам, например, от 2 – мультфильмы, от 5 – развивающие игры, от 8 – робототехника, от 11 – программирование в Python, от 14 – программирование игр, от 17 – финансовая грамотность.

Попробуйте составить словесный алгоритм решения задачи с помощью конструкции ветвления.



При решении задачи вы будете применять несколько раз условный оператор ветвления **if**. Он будет появляться внутри другого оператора **if**. В таких случаях операторы считаются **вложенными**.

Вложенное ветвление – это ветвление, находящееся в одной из ветвей другого ветвления.



Во вложенном ветвлении после одной развилки в ходе исполнения программы появляется другая развилка. При этом вложенные блоки имеют больший размер отступа (например, 8 пробелов). Из-за неэффективности выполнения вложенных операторов рекомендуют использовать один-два уровня вложения (рис. 4.22).

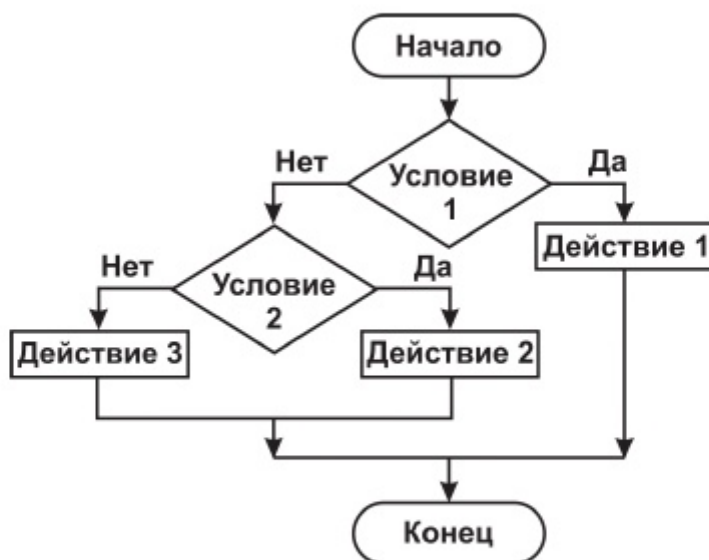


Рис. 4.22. Блок-схема вложенного ветвления

В конструкции вложенных операторов каждому оператору **if** соответствует только один оператор **else**. Совокупность этих операторов **else-if** означает, что если не выполняется предыдущее условие, то проверяется данное. Если ни одно из условий неверно, то выполняется тело оператора **else**.

Структура вложенных операторов

```

if <условие 1>:
  <инструкции 1>
else
  if <условие 2>:
    <инструкции 2>
  else
    <инструкции 3>
  
```



Определите, в какой из четвертей координатной плоскости находится точка с координатами (x, y) .

Шаг за шагом



1. Введите числа x и y .
 2. Разберите конструкцию вложенного ветвления, поскольку необходимо проверить более одного условия:

1) Если точка располагается в **первой** координатной четверти, то ее координаты удовлетворяют условию $x > 0, y > 0$.

2) Если точка располагается в **четвертой** координатной четверти, то ее координаты удовлетворяют условию $x > 0, y < 0$.

3) Если точка располагается во **второй** координатной четверти, то ее координаты удовлетворяют условию $x < 0, y > 0$.

4) Если точка располагается в **третьей** координатной четверти, то ее координаты удовлетворяют условию $x < 0, y < 0$.

3. Первое и второе действия можно объединить в один **if**, так как у них одинаковое **условие** $x > 0$. В зависимости от значения y точка будет лежать в **первой** или в **четвертой** координатной четверти. Таким образом получается **вложенное ветвление**.

4. Если условие $x > 0$ является ложным, то выполняются операторы после **else**, где вновь выполняется проверка условия $y < 0$ или $y > 0$, и от этого условия зависит, во **второй** или **третьей** четверти будет находиться точка.

5. Напишите код программы (рис. 4.23) с вложенным условием.

```

File Edit Format Run Options Window Help
x = int(input())
y = int(input())
if x > 0:
    if y > 0:
        # x > 0, y > 0
        print("Первая четверть")
    else:
        # x > 0, y < 0
        print("Четвертая четверть")
else:
    if y > 0:
        # x < 0, y > 0
        print("Вторая четверть")
    else:
        # x < 0, y < 0
        print("Третья четверть")
Ln: 12 Col: 32
1
1
Первая четверть
>>>
===== RESTART: F:/1
2
-2
Четвертая четверть
>>>
===== RESTART: F:/1
-3
3
Вторая четверть
>>>
===== RESTART: F:/1
-5
-3
Третья четверть

```

Рис. 4.23. Код программы вложенного ветвления



По заданному количеству глаз и ножек нужно отличить собаку, паука, морского гребешка и жука. У морского гребешка бывает более сотни глаз, а у паука их восемь. Также у паука восемь ног, а у морского гребешка их нет совсем. У собаки четыре ноги, а у жука – шесть ног, но глаз у них по два.

Внутри блока команд могут находиться другие условные операторы. Рассмотрим пример.

На рис. 4.24 представлен код программы. Вложенных условных операторов в программе несколько. К какому из них относится **else**, можно увидеть по отступу. Отступ у **else** должен быть такой же, как у **if**, к которому он относится.

```

File Edit Format Run Options Window Help
eyes = int(input())
legs = int(input())
if eyes >= 8:
    if legs == 8:
        print("spider")
    else:
        print("scallop")
else:
    if legs == 6:
        print("bug")
    else:
        print("dog")

```

----- RESTART:-----

```

100
0
scallop
>>> |

```

----- RESTART:-----

```

2
4
dog
>>> |

```

Рис. 4.24. Код программы «Глаза и ножки»

В некоторых ситуациях необходимо осуществить выбор больше чем из двух вариантов. В язык Python встроена возможность множественного ветвления на одном уровне вложенности, которое реализуется с помощью специальной конструкции **if... elif... else**.

Оператор **elif** образован от двух первых букв оператора **else**, к которым присоединено слово **if**. Это можно перевести как «**иначе если**». После него записывается **условие**. Конструкций **elif** может быть несколько, условия в них проверяются последовательно. Как только выполняется условие, запускается соответствующий этому условию блок команд и дальнейшая проверка не выполняется.

Блок **else** является необязательным, как и в обычном **if**. В отличие от **else**, в заголовке **elif** обязательно должно быть логическое выражение, как в заголовке **if**.

Такая конструкция введена в язык Python для улучшения читаемости. При применении записи **if-else** увеличивается количество отступов, что приводит к ухудшению читаемости. Например, программу, определяющую для точки четверть координатной плоскости, можно оптимизировать, используя «каскадную» последовательность операций **if... elif... else**.

Каскадные ветвления используются для разделения кода более чем на две ветви.



Структура каскадного ветвления

```

if <условие 1>:
<инструкции 1>
elif <условие 2>:
<инструкции 3>
...
else:
<инструкции 2>

```

Алгоритм выполнения каскадного ветвления показан на рис. 4.25.

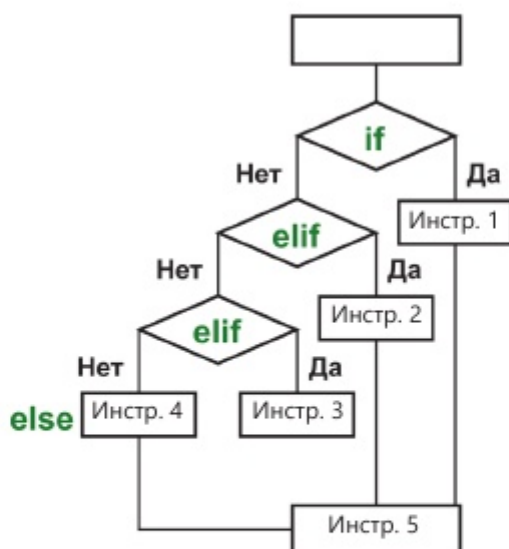


Рис. 4.25. Блок-схема конструкции **if-elif...-elif-else**

- 1) Сначала проверяется условие **1** в блоке **if**.
- 2) Если **условие 1** выполняется, то будет выполнена **инструкция 1**, и программа вернется в основную ветку **инструкции 5**.
- 3) Если **условие 1** не выполняется, то проверяется **условие 2** в блоке **elif**.
- 4) Если **условие 2** выполняется, то будет выполнена **инструкция 2**, и программа вернется в основную ветку **инструкции 5**.
- 5) Если **условие 2** не выполняется, то будет выполнено **условие 3**.
- 6) Если **условие 3** выполняется, то выполняется **инструкция 3**.
- 7) Если **условие 3** не выполняется, то будет выполнена **инструкция 4** из блока **else**.

Приведем пример.



При вводе числа **1** или **2** необходимо вывести словом название числа или что это другое число (рис. 4.26).

```
File Edit Format Run Options Window ===== RESTART
number = int(input())
1
if number == 1:
    print('Один')
    >>>
elif number == 2:
    print('Два')
    ===== RESTART
2
else:
    print('Другое число')
    Два
    >>>
    ===== RESTART
100
    Другое число
    >>>
```

Рис. 4.26. Вывод чисел с сообщением



Знание. Понимание



1. Произнесите ключевые слова на трех языках.
2. Какие условия называются вложенными?
3. Перечислите структуру вложенных операторов.



4. Каким образом выполняются вложенные операторы?

5. Что такое каскадное ветвление?
6. Перечислите структуру каскадного ветвления.
7. Поясните принцип работы каскадного ветвления.
8. Почему удобнее использовать каскадное ветвление?

Применение. Анализ



9. Составьте программу для вычисления значений функции, используя вложенные ветвления:

$$y = \begin{cases} x^2, & \text{если } x > 1 \\ 2x + 1, & \\ -3x + 5, & \text{если } x < -2, \\ x, & \text{в остальных случаях} \end{cases}$$



10. Напишите программу, которая запрашивает на ввод число. Если оно положительное, то на экран выводится цифра **1**. Если число отрицательное, выводится **-1**. Если введенное число **0**, то на экран выводится **0**. Используйте вложенное ветвление.

Синтез. Оценивание



11. Даны три целых числа. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: **3** (если все совпадают), **2** (если два совпадают) или **0** (если все числа различны). Используйте вложенное и каскадное ветвления. Сравните и оцените полученный результат.
12. Определите возможность существования треугольника по трем сторонам. Составьте программу, применяя вложенное или каскадное ветвление.



Дополнительные задания по программированию



1. Дано целое число **n**. Выведите следующее за ним четное число.

10-5997

2. Составьте программу для вычисления функции, заданной графиком (рис. 4.27), применяя вложенное или каскадное ветвление.

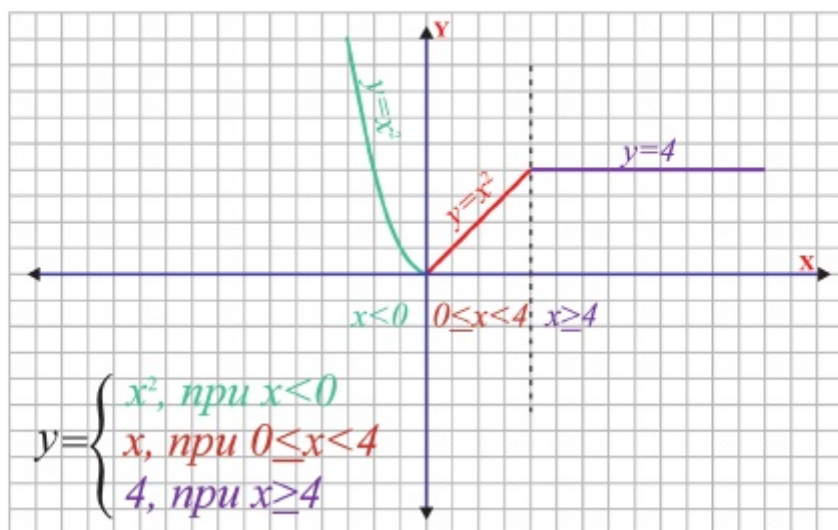


Рис. 4.27. График функции

3. Напишите программу, которая получает на вход три целых числа и выводит в три строки сначала максимальное, потом минимальное, после чего – оставшееся число. На ввод могут подаваться и повторяющиеся числа. Проанализируйте, какой способ решения данной задачи является наиболее оптимальным.
4. Дано четырехзначное число. Определите, является ли его десятичная запись симметричной. Если число симметричное, то выведите **1**, иначе выведите любое другое целое число. Число может иметь меньше четырех знаков. Тогда нужно считать, что его десятичная запись дополняется слева нулями.

4.4. ПРОГРАММИРОВАНИЕ СОСТАВНЫХ УСЛОВИЙ

ВЫ НАУЧИТЕСЬ

Использовать составные условия на языке программирования Python (пайтон).

КЛЮЧЕВЫЕ СЛОВА

Логическое высказывание	Логикалық тұжырым	Logical statement
Логическая операция	Логикалық амал	Logical operation
Составное условие	Күрделі шарт	Compound condition



Как вам известно, условие – это логическое выражение, состоящее из логических операций. Условие может быть **простым** или **составным**. Простое условие состоит из двух чисел, двух переменных или двух логических выражений, которые сравниваются между собой с использованием операций сравнения ($=$, $>$, $<$, $>=$, $<=$, $<>$). Например: $1 < 2$ (**Истина**), $5 < 0$ (**Ложь**), $x >= 5$, $x * y == x + y$.

Логическое высказывание – это утверждение, истинность которого можно определить однозначно.



Часто при решении некоторых задач возникает необходимость проверять выполнение двух и более условий.

Как вы думаете, какое условие будет называться составным?



Составное логическое выражение можно записывать так:

if $a > 0$ and $a < b$:

Это означает, что вложенный оператор выполняется, если **a** будет одновременно **больше нуля** и **меньше b**. В Python, в отличие от других языков программирования, существует также сокращенная запись составного логического выражения в виде **if $0 < a < b$:**

Рассмотрим конкретный пример.



В зависимости от возраста человек проходит разные стадии обучения, которые должны быть выведены в виде сообщений. Выделим условно группы, которые разделены на четыре диапазона:



- 1) от 3 до 6 лет – детский сад;
- 2) от 6 до 10 – начальная школа;
- 3) от 10 до 15 – основная школа;
- 4) от 15 и больше – старшая школа.

Как реализовать задачу с конструкцией ветвления с составным условием, показано на рис. 4.28.

```
old = int(input('Введите Ваш возраст: '))
print('Ваша ступень образования:', end=' ')

if 3 <= old < 6:
    print("Детский сад")
elif 6 <= old < 10:
    print("Начальная школа")
elif 10 <= old < 15:
    print("Основная школа")
elif 15 <= old:
    print("Старшая школа")
```

Введите Ваш возраст: 3
Ваша ступень образования: "Детский сад"
>>>

Рис. 4.28. Конструкция ветвления с составным условием

Из кода программы вы видите, что условие состоит из нескольких простых условий, которые объединены логическими операторами. Такие условия называют **составными**.

Для записи составного условия применяются логические операции **and (и)**, **or (или)**, **not (не)**, которые возвращают **Истина (True)** или **Ложь (False)** операнду:

Операнды: истина (T)=1, ложь (F)=0		и (логическое умножение)	или (логическое сложение)	не (логическое отрицание)
x	y	x and y	x or y	not x
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Чтобы лучше понимать, как работают составные условия со сложными логическими выражениями, разберите приведенные ниже примеры с их решениями (табл. 4.2).

Таблица 4.2. Примеры с их решениями

Пример	Решение
$2 > 5$ and $2 > 3$	$2 > 5 = \text{False}$, $2 > 3 = \text{False}$, $\text{False and False} = \text{False}$
$1 < 5$ and $1 > 0$	$1 < 5 = \text{True}$, $1 > 0 = \text{True}$, $\text{True and True} = \text{True}$
$3 >= 2$ or $3 < 1$	$3 >= 2 = \text{True}$, $3 < 1 = \text{False}$, $\text{True or False} = \text{True}$
$1 > 2$ or $1 < 0$	$1 > 2 = \text{False}$, $1 < 0 = \text{False}$, $\text{False or False} = \text{False}$
not $5 > 6$	$5 > 6 = \text{False}$, not $\text{False} = \text{True}$
not $6 > 5$	$6 > 5 = \text{True}$, not $\text{True} = \text{False}$

Приведем еще один пример, в котором нужно определить значение логического выражения: **not(x > y) and not (x < 0) or (y ≥ 10)**, если **x = -1, y = 10**.

На порядок действий влияют также скобки. Желательно не составлять очень сложные логические выражения и всегда разбивать их на части. Для решения примера отметим порядок выполнения операций (см. табл. 4.3).

Таблица 4.3. Решение логического выражения

Порядок выполнения	4	1	6	5	2	7	3
Логическое выражение	not	(x > y)	and	not	(x < 0)	or	(y ≥ 10)
Выполняем в скобках		$(-1 > 10)$			$(-1 < 0)$		$(10 \geq 10)$
		F			T		T
Выполняем not	not (F)			not (T)			
	T			F			
Выполняем and			T and F				
			F				
Выполняем or						F or T	
Ответ						True	

Приведем код программы с использованием логических операций по определению точки в одной из четвертей координатной плоскости, рассмотренной в предыдущей теме (рис. 4.29).

```

File Edit Format Run Options Window Help
print("Введите координаты точки")
x = int(input())
y = int(input())
if x > 0 and y > 0:
    print("Первая четверть")
elif x > 0 and y < 0:
    print("Четвертая четверть")
elif y > 0:
    print("Вторая четверть")
else:
    print("Третья четверть")

```

```

>>>
===== RESTART: F:/7класс
Введите координаты точки
4
5
Первая четверть
>>>
===== RESTART: F:/7класс
Введите координаты точки
-4
3
Вторая четверть
>>>
===== RESTART: F:/7класс
Введите координаты точки
-2
-3
Третья четверть
>>>
===== RESTART: F:/7класс
Введите координаты точки
5
-6
Четвертая четверть
>>>

```

Рис. 4.29. Код программы с использованием логических операций



Определите, принадлежит ли введенное число заданному интервалу $-50 < x \leq 0$ или $50 \leq x < 100$.



Шаг за шагом

1. Ввести число x .
2. Если выполняется условие $-50 < x \leq 0$ или $50 \leq x < 100$, то число принадлежит заданному промежутку.
3. Записать условие на языке программирования Python, используя логические операции $x > -50$ and $x < 0$ or $x > 50$ and $x < 100$.

Данное условие можно записать в сокращенной форме: $-50 < x \leq 0$ or $50 \leq x < 100$.

4. Вывести на экран ответ **True**, если условие истинно, и **False** в противном случае.

5. Записать код программы и выполнить его (см. рис. 4.30).

<pre>File Edit Format Run Options Window Help print("Введите число") x = int(input()) if x > -50 and x <=0 or x >= 50 and x <100: print("TRUE") else: print("FALSE")</pre>	<pre>File Edit Format Run Options Window Help print("Введите число") x = int(input()) if -50 < x <=0 or 50<=x <100: print("TRUE") else: print("FALSE") #сокращенная форма записи</pre>	<pre>----- RESTART: Введите число 0 TRUE >>> ----- RESTART: Введите число 10 FALSE >>> ----- RESTART: Введите число 70 TRUE >>> ----- RESTART:</pre>
--	--	---

Рис 4.30. Код программы о принадлежности числа заданному интервалу

Знание. Понимание



1. Назовите на трех языках ключевые слова.
2. Что такое составное условие?
3. В каких случаях будут применяться логические операции?
4. Какие правила выполнения логических операций существуют?
5. Какова приоритетность логических операций?
6. Определите значение логического выражения $(x < y) \text{ or } (y > 0) \text{ and not } (x \geq 0)$, если $x = 0$, $y = -20$.



Применение. Анализ



7. Составьте программу для вычисления значений выражений. Проанализируйте, при каких значениях x нельзя вычислить значение выражений. При вводе этих значений выведите соответствующее сообщение.



- 1) $\frac{x+3}{x-2}$;
- 2) $\frac{2x^2}{3x+18}$;
- 3) $\frac{3(x+1)}{5x^2}$.

8. Определите значение переменной **a** после выполнения фрагмента программы (рис. 4.31).

<pre>File Edit Format Run Options a = 10 b = 5 if a > 5 and a < b: a -= 5 print (a)</pre>	<pre>File Edit Format Run Options a = 7 b = -10 if a > 1 or a < b: a = 5 elif a > 1 and a == b: a -= 5 print (a)</pre>	<pre>File Edit Format Run Options Wir a = -1 b = -10 if a > 1 or a < b: a = 5 elif a > 1 and a == b: a -= 5 print (a)</pre>
a)	b)	c)

Рис. 4.31. Фрагменты программы



Синтез. Оценивание



9. Дан код программы. Сформулируйте условие задачи.

```
a = int(input())
if a < -5:
    print('Low')
elif -5 <= a <= 5:
    print('Mid')
else:
    print('High')
```

10. Составьте программу, позволяющую перевести баллы, полученные в процентах, в пятибалльную оценку:

85–100% – оценка «5»;
 65–84% – оценка «4»;
 46–64% – оценка «3»;
 21–45% – оценка «2»;
 0–20% – оценка «1».



Квадратное уравнение

Написать программу для решения квадратного уравнения $ax^2 + bx + c = 0$. При этом $a \neq 0$.



Шаг за шагом

Составим блок-схему (рис. 4.32).

1. Ввод коэффициентов уравнения: **a, b, c**.
2. **D** (дискриминант) присвоим значение $b^2 - 4ac$.
3. Если **D < 0**, т.е. отрицательный, то уравнение **не имеет решений**.
4. Если **D = 0**, то уравнение имеет единственное решение $x = -b / 2a$.
5. Если **D** положительный, то уравнение имеет два решения: x_1, x_2 , вычисляемые по формуле:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

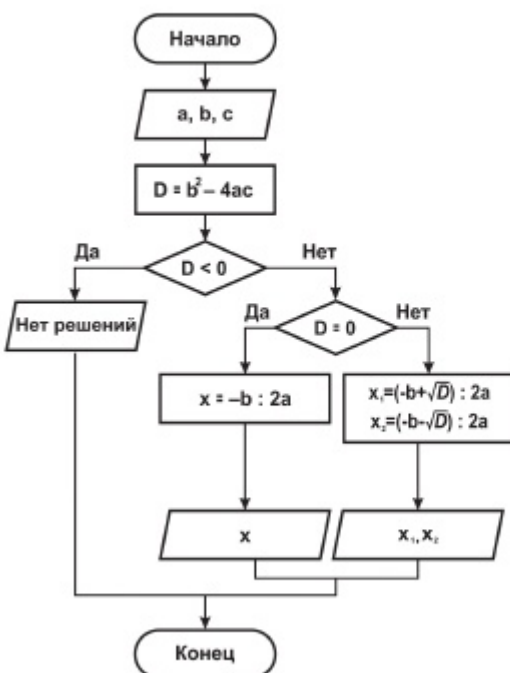


Рис. 4.32. Блок-схема решения квадратного уравнения

На блок-схеме мы видим, что вложенное ветвление делится на три ветви, которые реализуются с помощью каскадного ветвления в коде программы (рис. 4.33).

```

File Edit Format Run Options Window Help
a = float (input ("1-й коэффициент="))
b = float (input ("2-й коэффициент="))
c = float (input ("3-й коэффициент="))
D = b ** 2 - 4 * a * c
from math import sqrt
if D < 0:
    print ('Уравнение не имеет решений.')
elif D == 0:
    x = -b / (2 * a)
    print ('Уравнение имеет одно решение x =', x)
else:
    x1 = (-b + sqrt (D)) / (2 * a)
    x2 = (-b - sqrt (D)) / (2 * a)
    print ('Уравнение имеет два корня: x1 =', x1, 'x2 =', x2)
  
```

Рис. 4.33. Код программы решения квадратного уравнения

Решение квадратных уравнений

1) $x^2 + 2x - 3 = 0$

1-й коэффициент=1
2-й коэффициент=2
3-й коэффициент=-3
Уравнение имеет два корня: $x_1 = 1.0$ $x_2 = -3.0$
>>>

2) $x^2 + 6x + 9 = 0$

1-й коэффициент=1
2-й коэффициент=6
3-й коэффициент=9
Уравнение имеет одно решение $x = -3.0$
>>>

3) $x^2 + 2x + 17 = 0$

1-й коэффициент=1
2-й коэффициент=2
3-й коэффициент=17
Уравнение не имеет решений.
>>>

4.5. ОРГАНИЗУЕМ ВЫБОР

ВЫ НАУЧИТЕСЬ

Записывать разветвляющиеся алгоритмы на языке программирования Python (пайтон).

КЛЮЧЕВЫЕ СЛОВА

Выбор	Таңдау	The choice
Разветвляющийся алгоритм	Тармақталу алгоритмі	Branching algorithm



В некоторых языках программирования существует конструкция выбора **switch – case**. В языке программирования Python для реализации выбора из нескольких альтернатив можно использовать конструкцию **if – elif – else**, с которой вы уже познакомились при программировании вложенных условий. Разветвленные алгоритмы можно объединять и создавать из них комбинации операторов.

Попробуем решить задачу, в которой необходимо выставить оценку ученику в зависимости от результатов теста. Так как оценок несколько, то в зависимости от набранных баллов ученик может получить ту или иную оценку. Например, если количество баллов больше или равно 85, то ставится оценка «отлично»; больше или равно 70 баллам – оценка «хорошо»; больше или равно 55 баллам – оценка «удовлетворительно»; больше или равно 40 баллам, то оценка «неудовлетворительно».



```
points=int(input("Вы набрали количество баллов="))
if points >= 85:
    print("Молодец! Ваша оценка – "отлично")
elif points >= 70:
    print("Ваша оценка – "хорошо")
elif points >= 55:
    print("Ваша оценка – "удовлетворительно")
elif points >= 40:
    print("Ваша оценка – "неудовлетворительно")
else:
    print("Вам необходимо пересдать тест.")
```

Как вы думаете, как значение переменной **points** будет сравниваться со всеми условиями подряд? В каком случае на экране появится сообщение об оценке? В каком случае на экране появится текст **"Вам необходимо пересдать тест"**?





Ниже дан код программы. Сформулируйте условие задачи. Каким образом выполняется оператор выбора в программе? Проведите эксперимент. Что означает параметр `end=' '` в операторе вывода? Если его убрать, то как результат будет выведен на экран?



```
temperature=int(input("Какая температура на улице?"))
weather=int(input("Если идет дождь, то введите 1, иначе 0\n"))
if weather==1:
    print("Возьмите зонт и ",end='')
if temperature > 20:
    print("одевайтесь легко - на улице тепло!")
elif 1<=temperature<=20:
    print("одевайте легкую куртку!")
elif -15<=temperature<=0:
    print("одевайтесь очень тепло, так как на улице холодно.")
```



Напишите программу **Калькулятор**, которая в зависимости от ввода значений в переменные будет выполнять соответствующие действия и выводить их на печать. Например, если введено значение «**add**», то выполняется сложение, «**sub**» – вычитание, «**mul**» – умножение, «**div**» – деление. В случае ввода другого значения будет выведено сообщение об ошибке «**error**» (рис. 4.34).



```
a1=int(input('Введите первое число: '))
a2=int(input('Введите второе число: '))
operation=input('add/sub/mul/div: ')
if operation == 'add':
    a3=a1+a2
elif operation == 'sub':
    a3=a1-a2
elif operation == 'mul':
    a3=a1*a2
elif operation == 'div':
    a3=a1/a2
else:
    a3='ошибка!'
print('Ответ = ', a3)
```

```
Введите первое число: 55
Введите второе число: 7
add/sub/mul/div: div
Ответ = 7.857142857142857
>>>
===== RESTART:
Введите первое число: 88
Введите второе число: 33
add/sub/mul/div: sub
Ответ = 55
>>>
===== RESTART:
Введите первое число: 26
Введите второе число: 38
add/sub/mul/div: 22
Ответ = ошибка!
```

Рис. 4.34. Программный блок условного оператора `elif`

Знание. Понимание



1. Почему операция вида $a < b = c$ недопустима, а операция вида $a < b == c$ допустима?
2. Чем отличаются результаты операций «/» и «//» для целых чисел? А для вещественных чисел?
3. Какая структура является результатом работы функции `divmod()`?
4. Установите соответствие: расположите операторы в правильном порядке.



1	a) <code>a = inprint("сколько вам лет?")</code>
2	b) <code>a = int(a)</code>
3	c) <code>else:</code>
4	d) <code>print("вы можете управлять автомобилем")</code>
5	e) <code>print("вы все еще не можете водить машину")</code>
6	f) <code>if a < 18:</code>

5. Запишите результаты условного оператора (рис. 4.35).

<pre>a = 22 b = -22 if a > b: print(a/b) else : print(a/2)</pre>	<pre>a = 10 b = 2 if a % b == 0: print(a*2) else: print(a/2)</pre>	<pre>a = 10 b = 2 if a % b != 0: print(a*2) else: print(a/2)</pre>
a)	b)	c)

Рис. 4.35. Условные операторы

Применение. Анализ



6. Напишите программу, которая будет выводить перевод из одной величины в другую. К примеру, если ввести 1 версту, то выводится на экран перевод в сажени:

- 1 верста – 500 сажений;
- 1 сажень – 3 аршина;
- 1 аршин – 16 вершков;
- 1 вершок – 4,45 см.



7. Какой результат будет выведен на печать, если для переменной **grade** ввели значение, равное **96**? Код программы представлен ниже.

```

grade=int(input("Введите Ваш балл: "))
if grade >= 65:
    print("Проходной балл:")
    if grade >= 90:
        if grade > 96:
            print("A+")
        elif grade > 93 and grade <= 96:
            print("A")
        elif grade >= 90:
            print("A-")

```

8. Код предыдущей программы представлен в виде словесного алгоритма. Проанализируйте, какое значение принимает условие: **true** или **false**? Какой оператор будет выполнен после пункта 6?

- 1) Проверяет, является ли оценка больше или равной **65** (**true** или **false**).
- 2) Выведет "**Проходной балл:**"
- 3) Проверяет, является ли оценка больше или равной **90** (**true** или **false**).
- 4) Проверяет, превышает ли оценка **96** (**true** или **false**).
- 5) Проверяет, является ли оценка больше **93**, а также меньше или равна **96** (**true** или **false**).
- 6) Выведет на печать.



Синтез. Оценивание



9. Ниже представлен код программы. Сформулируйте условие задачи. Какие ошибки допущены в ней? Какие правила необходимо применять при записи вложенного условного оператора? Как можно модернизировать программу?

```

x = input ("Введите первое число: ")
y = input ("Введите второе число: ")
z = input ("Введите третье число: ")
if x > y :
    if y > z :
        print (x , y , z )
    else :
        if x > z :

```

```

print ( x , z , y )
else :
print ( z , x , y )
else :
if x > z :
print ( y , x , z )
else :
if y > z :
print ( y , z , x )
else :
print ( z , y , x )

```

10. Напишите программу определения дня недели по введенному номеру. Является он рабочим днем или выходным? Например, если введен номер **1**, то понедельник, номер – **6**, то суббота.

11. У вас есть 1000 тенге. Если стоимость билета в кино составит 1000 тенге, тогда вы купите билет, если стоимость билета в кино – 700 тенге, то купите билет и напиток. Если стоимость билета в кино – 1200 тенге, тогда кино посмотрите в другой раз. Напишите программу, используя вложенный оператор.

Задачи с файлами



1. «Високосный год». Данные вводятся с клавиатуры или из файла **input.txt**, выводятся на экран или в файл **output.txt**.

Дано натуральное число. Требуется определить, является ли год с данным номером високосным. Если год является високосным, то выведите **YES**, иначе выведите **NO**. Напомним, что в соответствии с григорианским календарем год является високосным, если его номер кратен **4**, но не кратен **100**, а также если он кратен **400**.

2. «Входные данные». Вводится одно натуральное число.

3. «Выходные данные». Выведите ответ на задачу.

4. «Четные и нечетные числа». Данные вводятся с клавиатуры или из файла **input.txt**, выводятся на экран или в файл **output.txt**.

Даны три целых числа **A**, **B**, **C**. Определите, есть ли среди них хотя бы одно четное и хотя бы одно нечетное.

5. «Входные данные». Числа **A**, **B**, **C** – не превышающие по модулю 10 000.

6. «Выходные данные». Одна строка – «**YES**» или «**NO**».

ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ IV**1. Язык Python относится:**

- а) к структурному программированию;
- б) к объектно ориентированному программированию;
- в) к логическому программированию;
- г) к процедурному программированию.

2. Правильное название переменной в Python:

- а) 123alfa;
- б) alfa;
- в) @alfa;
- г) print.

3. Тип данных в Python, являющийся целым.

- а) float.
- б) int.
- в) bool.
- г) integer.

4. Результат выполнения следующего кода.

```
>>> 2/3
```

- а) 0.
- б) 1.
- в) 0.6666666666666666.
- г) 0.66.

5. Число -77 после выполнения алгоритма

```
x = float(input())
```

```
print(x)
```

будет записано как:

- а) 77;
- б) 77.0;
- в) -77.0;
- г) 77.00.

6. Функция, которая выводит что-либо на экран монитора.

- а) write().
- б) log().

- в) out().
- г) print().

7. Как получить данные от пользователя?

- а) Использовать метод **get()**.
- б) Использовать метод **cin()**.
- в) Использовать метод **read()**.
- г) Использовать метод **input()**.

8. Каков будет результат выполнения следующего кода?

```
>>> a=b=c=3
```

```
>>> b=a/3
```

```
>>> print c
```

- а) 3
- б) SyntaxError: Missing parentheses in call to "print"
- в) -3
- г) 1

9. Что будет выведено в результате выполнения кода программы?

```
a = 10
```

```
b = a + 10
```

```
a = b * 100
```

```
print(a)
```

- а) 120.
- б) 2000.
- в) 20000.
- г) 1000.

10. Результатом вычисления print(24 // 3) будет число:

- а) 4;
- б) 8;
- в) 12;
- г) 9.

11. Каким будет результат выполнения алгоритма?

```
a = 5
```

```
b = 7
```

```
s = a * b
```

```
print(s)
```

- а) 57.
- б) 12.
- в) 35.
- г) 2.

12. Что такое условие в алгоритме ветвления?

- а) Это выражение, которое в конечном итоге получается «Истина» (True) или «Ложь» (False).
- б) Это логическое выражение, которое в конечном итоге получается «Истина» (True) или «Ложь» (False).
- в) Это логическое выражение.
- г) Это арифметическое выражение, которое в конечном итоге получается «Истина» (True) или «Ложь» (False).

13. Что будет в результате выполнения алгоритма?

Входные данные: $x = -100$

```
x = int(input())
```

```
if x > 0:
```

```
    print(x)
```

```
else:
```

```
    print(-x)
```

- а) -100.
- б) 100.
- в) 10.
- г) -10.

14. Что будет в результате выполнения программы?

Входные данные: $a = -5$, $b = -10$

```
a = int(input())
```

```
b = int(input())
```

```
if a < b:
```

```
    print(a)
```

```
else:
```

```
    print(b)
```

- а) 10.
- б) -5.
- в) 15.
- г) 10.

15. Для записи составного условия применяются логические операторы и, или, не. Выберите верную запись.

- а) or, not, if.
- б) and, or, not.
- в) and, or, if.
- г) and, else, not.

16. Что будет напечатано в результате выполнения следующего алгоритма программы?

Входные данные: a=15, b=-10

a = int(input())

b = int(input())

if a % 10 == 0 or b % 10 == 0:

print('yes')

else:

print('no')

- а) yes.
- б) no.
- в) 15.
- г) -10.

17. Какой результат будет напечатан после выполнения кода?

x = 23

num = 0 if x > 10 else 11

print(num)

- а) 0.
- б) 10.
- в) 23.
- г) Ошибка.

18. Что будет выведено в результате?

name = "John"

print("Hi, %s" % name)

- а) "Hi, name"
- б) "Hi, "
- в) Ошибка
- г) "Hi, John"

19. Что такое файл?

- а) Файл – это определенное количество информации, имеющее имя и хранящееся в долговременной памяти.
- б) Файл – это определенное количество информации, имеющее имя и хранящееся в кратковременной памяти.
- в) Файл – это имя, хранящееся в долговременной памяти.
- г) Файл – это определенное количество информации, хранящееся в долговременной памяти.

20. Для открытия на запись файла нужно выбрать режим:

- а) r;
- б) w;
- в) a;
- г) i.

5.1. ПОСТАНОВКА ПРОБЛЕМЫ

ВЫ НАУЧИТЕСЬ

Формулировать проблему для исследования.

КЛЮЧЕВЫЕ СЛОВА

Модель	Модель	Model
Проблема	Мәселе	Problem
Этапы решения	Шешу қадамдары	Solution steps



Что такое проблемная ситуация?

Какие этапы решения задачи в среде программирования Python необходимо пройти?



Очень часто в жизни вам приходится сталкиваться с проблемными ситуациями. Например, вы хотите прочитать книгу на иностранном языке, но ваш словарный запас еще недостаточен, чтобы понять ее содержание. Или вы хотите решить задачу по алгебре, но не знаете способ решения. В любом случае необходимо сначала четко сформулировать свою проблему. Считается, что правильная постановка проблемы – залог успеха, т.е. первый шаг решения задачи.

Проблема – это форма научного отображения проблемной ситуации.

Проблемная ситуация – это затруднение, которое возникает из-за несоответствия между имеющимися знаниями и теми, которые необходимы для решения возникшей задачи.



Компьютер был создан для решения задач и обработки данных. Как решить ту или иную задачу с помощью компьютера? Решение любой задачи с помощью компьютера можно разделить на пять основных этапов (рис. 5.1).

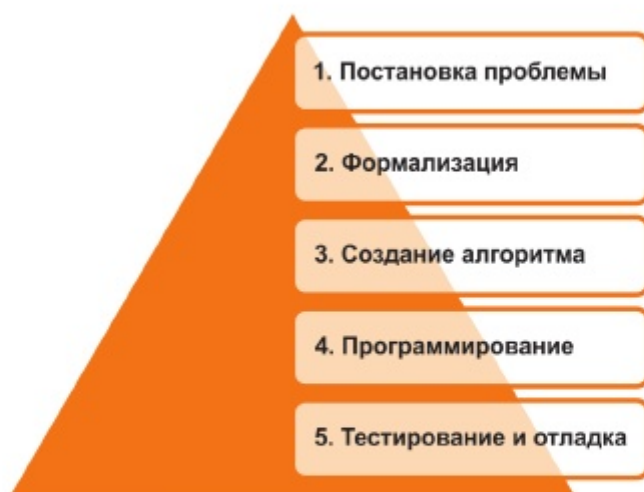


Рис. 5.1. Этапы решения задачи

1. На **этапе постановки проблемы** необходимо:

- определить цель;
- определить необходимый объем информации;
- дать точную формулировку задачи;
- предложить идею решения задачи;
- описать исходные данные и способы их хранения;
- определить форму выдачи результатов.

На этом этапе нужно сформулировать вопросы:

- 1) «Что нужно найти по условию задачи?»
- 2) «Что при этом дано?»
- 3) «Что я знаю для решения этой задачи?»
- 4) «Что мне нужно узнать?»

Очень важно найти ответы на эти вопросы. Поэтому постановка проблемы включает в себя следующие моменты: сбор информации о задаче, формулировку условия задачи, определение конечных целей решения задачи, определение формы выдачи результатов.

2. **Этап формализации** предполагает создание математической модели решаемой задачи. Чтобы решить задачу, необходимо описать объект в математических терминах (формулах), т.е. построить его модель, которая позволяет математически решить задачу.

Этот этап выполняет человек, способный разработать математическое описание поставленной задачи.

На этапе построения математической модели нужно определить:

- ограничения и допущения;
- что дано – перечисление исходных данных;
- что найти – результат;
- как найти математические соотношения, которые связывают исходные данные с результатами.

На этом этапе можно воспользоваться уже известными методами и формулами или разработать собственные.

3. На **этапе записи алгоритма** нужно представить алгоритм в виде блок-схемы, таблицы, словесным способом или на алгоритмическом языке.

Алгоритм решения задачи разрабатывается на основе построенной математической модели и представляет конечную последовательность предписаний (правил), которая определяет процесс преобразования исходных данных в результаты решения задачи.

4. **Этап программирования** выполняется на компьютере и включает в себя выбор языка программирования и запись алгоритма на выбранном языке программирования.

5. **Тестирование и отладка программы** – это проверка программы на наличие технических ошибок.

Тестирование – процесс проверки правильности работы всей программы или ее составных частей.



Отладка программы – процесс поиска и устранения ошибок в программе, который производится после ее прогона на компьютере.

Ошибки могут быть **алгоритмическими**, **грамматическими** или **техническими**. Отладка предполагает совершенствование программы, а ошибки в ходе отладки исправляют путем **контрольного тестирования**.

Проверка правильности работы программы осуществляется сравнением заранее известных результатов тестов с результатами, полученными компьютером. Этот этап – совместная работа человека и компьютера. Все этапы решения задачи связаны между собой (рис. 5.2).

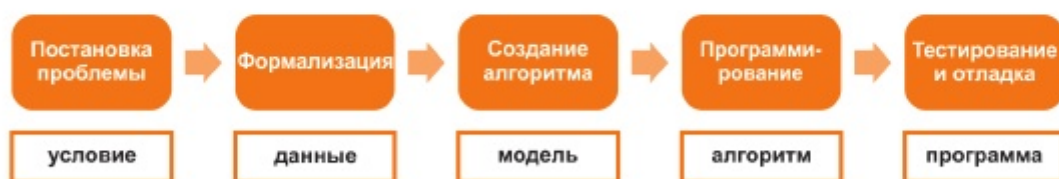


Рис. 5.2. Этапы решения задачи



Составьте программу решения линейного уравнения $ax + b = 0$.



Шаг за шагом

1. **Постановка проблемы.** Необходимо составить программу решения линейного уравнения $ax + b = 0$, где a, b – коэффициенты уравнения и могут быть заданы любыми числами. Необходимо ввести коэффициенты и найти корень уравнения x . На уроках алгебры вы узнали, как решить данное уравнение: $x = -b/a$. Но необходимо рассмотреть все возможные варианты решения уравнения. Какой ответ можно получить при $a = 0$, при $b = 0$ и $b \neq 0$?

Решая эту задачу, вы вспомните, как составляются программы с использованием вложенного условия и научитесь выполнять тестирование программы.

2. **Формализация.** Если $a \neq 0$, то $x = -b/a$. Если $a = 0$, то нужно рассмотреть варианты $b = 0$, $b \neq 0$. Если $b = 0$, то корнем уравнения может быть **любое число**, если $b \neq 0$, тогда уравнение **не имеет корней**.

3. На рисунке 5.3 показана **блок-схема** решения линейного уравнения.

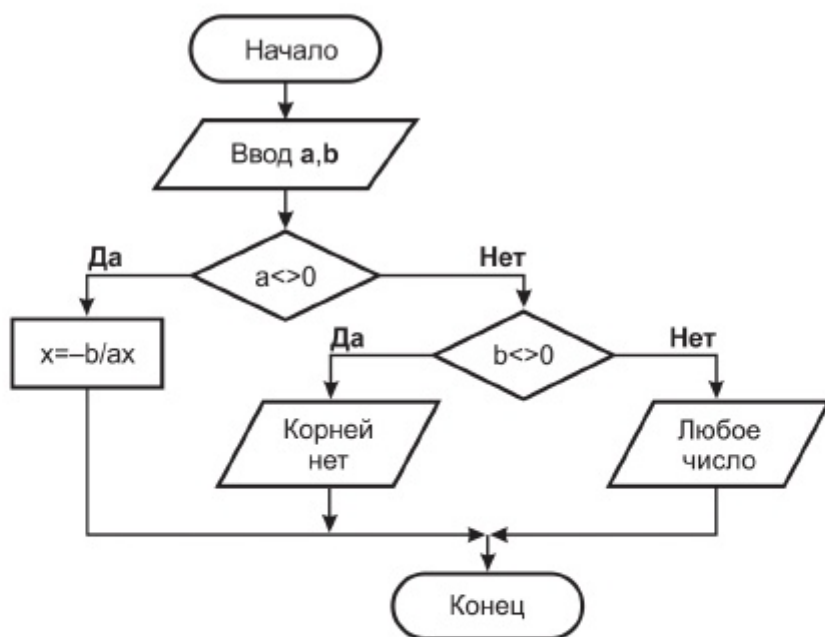


Рис. 5.3. Блок-схема решения

4. Программа решения задачи (рис. 5.4).

```

File Edit Format Run Options Window Help
print ("Введите коэффициенты уравнения")
a = int(input())
b = int(input())
if a!= 0:
    print ("x=", -b/a)
elif b!= 0:
    print ("Корней нет")
else:
    print ("Любое число")

>>>
===== RESTART: F:/7класс_u
Введите коэффициенты уравнения
5
7
x= -1.4
>>>
===== RESTART: F:/7класс_u
Введите коэффициенты уравнения
0
7
Корней нет
>>>
===== RESTART: F:/7класс_u
Введите коэффициенты уравнения
0
0
Любое число

```

Рис. 5.4. Код программы

5. **Тестирование и отладка программы.** Введем данные и проверим работу программы (табл. 5.1).

Таблица 5.1. Таблица тестирования программы

№	a	b	Результат
1	5	7	-1,4
2	-50	12	0,24
3	0	7	Корней нет
4	0	0	Любое число

Знание. Понимание



1. Произнесите ключевые слова на трех языках.
2. Перечислите этапы решения задачи на компьютере.
3. Что необходимо сделать на этапе постановки проблемы?
4. Что такое проблемная ситуация?
5. Что нужно сделать на этапе формализации?
6. Какими способами можно представить алгоритм?
7. Как выполнить проверку правильности работы программы?
8. Какие этапы решения задачи выполняются без компьютера?





Применение. Анализ



9. Приведите примеры проблемных ситуаций из жизни. Как вы решали эти проблемы?

10. Фирма набирает сотрудников от **25** до **40** лет включительно. Введите возраст человека и определите, подходит ли он фирме (выведите ответ «**подходит**» или «**не подходит**»).

11. Заданы размеры **A** и **B** прямоугольного отверстия и размеры кирпича **X**, **Y**, **Z**. Определите, пройдет ли кирпич через отверстие. Проанализируйте все возможные варианты решения задачи. Составьте программу. Опишите этапы решения задачи.



Синтез. Оценивание



12. Приведите пример задачи, при решении которой нужно применять условный оператор. Попросите одноклассника решить эту задачу, составив все этапы. Оцените его решение.

13. Найдите координаты точки пересечения двух прямых $y_1 = k_1x + b_1$ и $y_2 = k_2x + b_2$.

В каких координатных четвертях находится точка пересечения? Если такой точки нет, то выведите соответствующее сообщение. Выполните все этапы решения задачи. Постройте чертеж в тетради. Составьте программу и оцените полученный результат.



Для правильного выбора той или иной задачи приведем их классификацию по предметным областям. Можно выделить следующие типы задач.

1. Задачи с практическим содержанием. Для них известны:
 - математические формулы;
 - физические законы.
2. Задачи, для решения которых неизвестны точные методы решения:
 - расчет площадей;
 - решение уравнений;
 - дискретизация непрерывных процессов.
3. Задачи из различных областей деятельности человека:
 - экономики, экологии, лингвистики и т.д.;
 - актуализация знаний по информатике для решения задач (экологических, шифровка текста и т.д.).
4. Задачи информационного моделирования:
 - организация данных;
 - алгоритмы функционирования модели;
 - записная книжка;
 - модель исполнителя и т.д.

5.2. РАЗРАБОТКА АЛГОРИТМА

ВЫ НАУЧИТЕСЬ

Разрабатывать алгоритм решения задачи.

КЛЮЧЕВЫЕ СЛОВА

Алгоритм	Алгоритм	Algorithm
Ветвление	Тармақталған	Branching
Последовательность	Жүйелі	Succession
Блок-схема	Сұлба	Flowchart
Если ... то, иначе	Егер ... онда, немесе	If...then else



Какой этап следует за этапом разработки модели?
Что такое алгоритм?



После постановки проблемы и построения математической модели необходимо составить алгоритм решения задачи. Алгоритм решения вычислительной задачи представляет собой совокупность правил преобразования исходных данных в виде результата. Человек постоянно автоматически пользуется алгоритмами при решении задач, не задумываясь над этим. Сегодня в качестве исполнителей алгоритмов человеку служат многие автоматические устройства и, прежде всего, компьютер. Алгоритм составляется для исполнителя на формальном языке. Это означает, что в формальном языке имеется конечное количество синтаксических конструкций. Формальное выполнение алгоритма означает, что алгоритм можно выполнять, не вникая в содержание поставленной задачи, а только строго выполняя последовательность действий, описанных в алгоритме.

Алгоритм – это точное описание упорядоченной последовательности действий, приводящей за конечное число шагов к необходимому результату.

Общие характеристики алгоритмов называются **свойствами**.



Мир алгоритмов очень разнообразен. Но, несмотря на это, можно выделить общие свойства, которыми обладает любой алгоритм.

На рис. 5.5. показаны **основные свойства алгоритмов**.



Рис. 5.5. Основные свойства алгоритма

Алгоритмы можно записывать по-разному. Форма записи, состав и количество операций алгоритма зависят от того, кто будет его исполнителем. Алгоритмы, составленные для некоторого исполнителя, можно представить в форме записей, показанных на рис. 5.6.



Рис. 5.6. Формы записи алгоритмов

Словесные и табличные формы записи алгоритмов составляются для исполнителя «Человек», а программная форма записи используется на языке программирования, понятном исполнителю «Компьютер». Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным и табличным.

Блок-схемой называется графическое изображение алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций.



При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий (рис. 5.7).







Вид стандартного графического объекта	Назначение
	Начало и конец алгоритма
	Процесс выполнения действий
	Условие выполнения действий
	Ввод или вывод данных
	Счетчик
	Линия перехода

Рис. 5.7. Графические объекты блок-схемы

Рассмотрим форму записи алгоритма на примере.

Составьте алгоритм и программу перевода одних единиц измерения информации в другие.



Шаг за шагом



1. Постановка проблемы. Пользователь вводит количество байт или килобайт, мегабайт или гигабайт. Следует перевести значение во все другие единицы измерения.

2. **Формализация.** Вы знаете, что 1 Кб = 1024 байт, 1 Мб = 1024 Кб, 1 Гбайт = 1024 Мбайт, или $1024 = 2^{10}$.

В зависимости от единицы измерения, которую введет пользователь (обозначим ее через **k**), осуществляется перевод в другие единицы измерения:

– если пользователь введет **байты (k = 1)**, то для перевода в килобайты используем формулу $a = x/2^{10}$, в мегабайты – $a = x/2^{20}$, в гигабайты – $a = x/2^{30}$;

– если пользователь введет **килобайты (k = 2)**, то для перевода в байты используем формулу $a = x * 2^{10}$, в мегабайты – $a = x/2^{10}$, в гигабайты – $a = x/2^{20}$;

– если пользователь введет **мегабайты (k = 3)**, то для перевода в байты используем формулу $a = x * 2^{20}$, в килобайты – $a = x * 2^{10}$, в гигабайты – $a = x/2^{10}$;

– если пользователь введет **гигабайты (k = 4)**, то для перевода в байты используем формулу $a = x * 2^{30}$, в килобайты – $a = x * 2^{20}$, в мегабайты – $a = x * 2^{10}$.

3. Составим **блок-схему** (рис. 5.8).

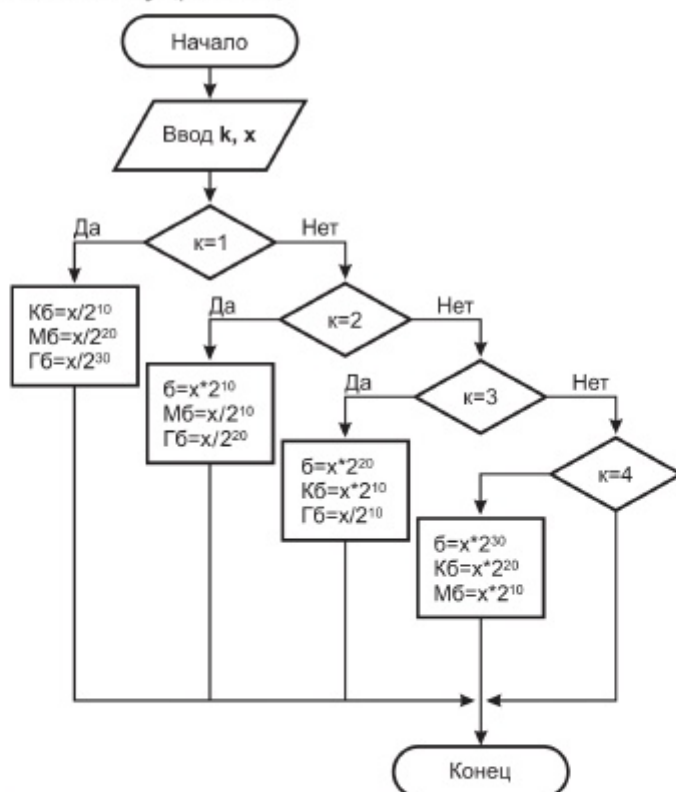


Рис. 5.8. Блок-схема перевода в другие единицы измерения

4. Составим программу решения задачи.

```

File Edit Format Run Options Window Help
k = input("Выберите единицы измерения:\n \
\t1 - байты, \n \
\t2 - килобайты, \n \
\t3 - мегабайты, \n \
\t4 - гигабайты. \n#: ")

x = float(input("Введите значение: "))

if k == '1':
    print("Килобайты: %10.3f" % (x / 2**10))
    print("Мегабайты: %10.3f" % (x / 2**20))
    print("Гигабайты: %10.3f" % (x / 2**30))
elif k == '2':
    print("Байты: %14d" % (x * 2**10))
    print("Мегабайты: %10.3f" % (x / 2**10))
    print("Гигабайты: %10.3f" % (x / 2**20))
elif k == '3':
    print("Байты: %14d" % (x * 2**20))
    print("Килобайты: %10d" % (x * 2**10))
    print("Гигабайты: %10.3f" % (x / 2**10))
elif k == '4':
    print("Байты: %14d" % (x * 2**30))
    print("Килобайты: %10d" % (x * 2**20))
    print("Мегабайты: %10d" % (x * 2**10))

===== RESTART: F:/7класс_учеб:
Выберите единицы измерения:
    1 - байты,
    2 - килобайты,
    3 - мегабайты,
    4 - гигабайты.
#: 1
Введите значение: 1048576
Килобайты: 1024.000
Мегабайты: 1.000
Гигабайты: 0.001
>>>
===== RESTART: F:/7класс_учеб:
Выберите единицы измерения:
    1 - байты,
    2 - килобайты,
    3 - мегабайты,
    4 - гигабайты.
#: 3
Введите значение: 5624000
Байты: 5897191424000
Килобайты: 5758976000
Гигабайты: 5492.188
>>>
===== RESTART: F:/7класс_учеб:
Выберите единицы измерения:
    1 - байты,
    2 - килобайты,
    3 - мегабайты,
    4 - гигабайты.
#: 4
Введите значение: 120547777
Байты: 129437189955125248
Килобайты: 126403505815552
Мегабайты: 123440923648

```

Рис. 5.9. Код программы

Код программы можно увидеть на рис. 5.9.

Знание. Понимание



1. Произнесите на трех языках ключевые слова.
2. Что такое алгоритм?
3. Перечислите основные свойства алгоритма.
4. Какие формы записи алгоритма вам известны?
5. Что такое блок-схема?
6. Почему графический способ алгоритма наиболее распространен?
7. Назовите стандартные элементы блок-схем.





Применение. Анализ



8. Составьте табличный алгоритм решения задачи. «Пусть имеется некоторый исполнитель, например **Арифмометр**, который понимает только две команды (**прибавь 1**, **умножь на 2**), и индикатор, на котором высвечиваются числа. Первоначально на индикаторе горит цифра **0**. Как с его помощью за восемь команд получить число **68**» (табл. 5.2)?

Таблица 5.2. Табличный алгоритм

№	Команда	Число
1	прибавь 1	1
2		
3		
4		
5		
6		
7		
8		68

9. Заполните таблицу 5.3, составив табличный алгоритм решения задачи «Число 99»: «Получите число 99 за 10 команд».

Таблица 5.3. Алгоритм «Число 99»

№	Команда	Число
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		99

10. Рассмотрите программу (рис. 5.10). Проанализируйте, какую задачу решает эта программа. Составьте блок-схему.

```
File Edit Format Run Options Window Help
a=int(input())
if a%2==0:
    print(" число", a, "четное")
else:
    if a==0:
        print(" число", a, "=0")
    else:
        print(" число", a, "нечетное")
```

Рис. 5.10. Код программы

Синтез. Оценивание



11. Составьте блок-схему и программу решения задачи: «Шахматный конь ходит буквой "Г", т.е. на две клетки по вертикали в любом направлении и на одну клетку по горизонтали, или наоборот. Даны две различные клетки шахматной доски. Определите, может ли конь попасть с первой клетки на вторую одним ходом». Введите данные, оцените полученный результат.

12. Составьте блок-схему и программу решения задачи: «На оси **OX** расположены три точки **a**, **b**, **c**. Определите, какая из точек (**b** или **c**) расположена ближе к **a**».



С понятием «формализация» мы часто сталкиваемся в процессе общения и познания. Например, формулируем мысли и предложения, оформляем отчеты и проекты, заполняем формуляторы списка или базы данных, применяем и преобразуем формулы и т.п.



Под формализацией понимается как сведение некоторого содержания в виде выбранной формы: текста, правил, теоремы, научной теории, сигналов и др. План действий в результате формализации переводится в алгоритм.

Приведем пример. Оглавление учебника – это формализация его содержательных частей, а сам текст учебника – формализация посредством языковых конструкций мыслей, идей, размышлений автора. Итогом формализации научной теории является совокупность формул, программ, графиков, схем, блок-схем, таблиц, инфографики и др.

5.3. ПРОГРАММИРОВАНИЕ АЛГОРИТМА

ВЫ НАУЧИТЕСЬ

Записывать код программы по готовому алгоритму.

КЛЮЧЕВЫЕ СЛОВА

Программа	Программа	Program
Комментарий	Түсініктеме	Comment
Идентификатор	Идентификатор	Identifier



Какой этап следует за этапом построения алгоритма?

Как записывается разветвляющийся алгоритм на языке программирования Python?

После составления графической записи алгоритма (блок-схемы) составляют программу.



Программа – запись алгоритма на машинном языке, понятном компьютеру. Она состоит из набора последовательных команд (операторов) для исполнителя «Компьютер», который должен их выполнить для решения поставленной задачи.

Составляют программы люди – **программисты**. Сам процесс создания или написания программ называется **программированием**. С шестого класса вы начали изучать язык программирования Python. Вы увидели и оценили, что Python легок в освоении. У этого языка имеется ряд особенностей:

- в процессе выполнения исходный код частями преобразуется в машинный специальной программой – **интерпретатором**;
- читать код легко благодаря понятному синтаксису;
- это универсальный язык программирования, поддерживающий объектно ориентированное программирование.

Программа в Python состоит из последовательности допустимых символов – **лексем**, записанных в определенном порядке и по определенным правилам.

Лексемы включают в себя:

1) **комментарии**, предназначенные для пояснения кода. Они начинаются с символа # и действуют до конца строки (рис. 5.11);

2) **литералы** – значения, представленные в коде программы, например числа или строки (рис. 5.12);

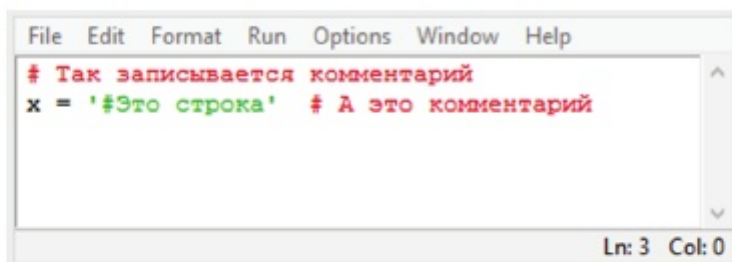


Рис. 5.11. Пример комментария

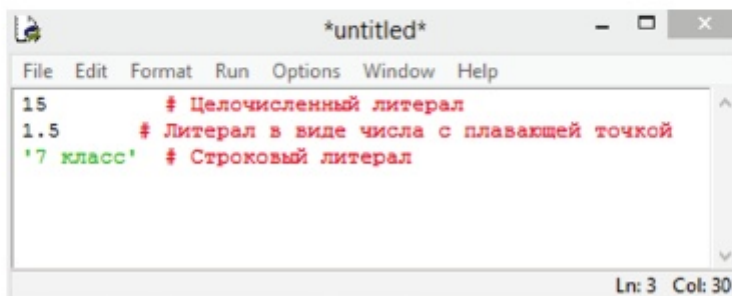


Рис. 5.12. Пример литералов

3) **знаки пунктуации**, которые используются для различных целей. Это ' " + * () , . и другие символы;

4) **идентификаторы** – имена, которые даются объектам.

Идентификатор имеет следующие правила:

- может состоять из букв (ASCII или Unicode), знаков подчеркивания `_` или цифр 0–9;

- цифра не может быть на первом месте;

- чувствителен к выбору регистра букв: `UserName`, `username` и `USERNAME` – это всё разные идентификаторы;

- он не должен совпадать с каким-либо из ключевых слов языка Python (табл. 5.4);

Таблица 5.4. Примеры имен идентификаторов

Допустимые	Недопустимые
a1	2a
my_name	здесь есть пробелы
name_7	>a1b2_c3
a1b2_c3	“текст_в_кавычках”
anna123	print

5) **ключевые слова** – это часть языка. Их нельзя использовать для названия переменных, функций. В табл. 5.5 перечислены некоторые ключевые слова.

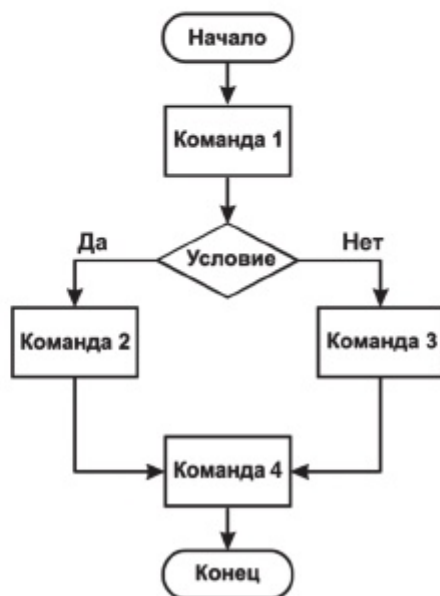
Таблица 5.5. Ключевые слова

Ключевое слово	Описание
False	ложь
True	правда
None	«пустой» объект
and	логическое и
with / as	менеджер контекста
assert условие	возбуждает исключение, если условие ложно
break	выход из цикла
class	пользовательский тип, состоящий из методов и атрибутов
continue	переход на следующую итерацию цикла
def	определение функции
del	удаление объекта
elif	в противном случае, если
else	иначе
except	перехватить исключение
for	цикл for
from	импорт нескольких функций из модуля
if	если
import	импорт модуля
in	проверка на вхождение
is	ссылаются ли 2 объекта на одно и то же место в памяти
lambda	определение анонимной функции
not	логическое не
or	логическое или
pass	ничего не делающая конструкция
raise	возбудить исключение
return	вернуть результат
try	выполнить инструкции, перехватывая исключения
while	цикл while
yield	определение функции-генератора

Все решаемые задачи можно классифицировать по процессам, лежащим в алгоритме их решения: **линейным**, **разветвляющимся**, **циклическим** (рис. 5.13).

Линейный

Линейный алгоритм, в котором все команды выполняются последовательно одна за другой.

Разветвляющийся

Разветвляющийся алгоритм, в котором та или иная последовательность команд выполняется в зависимости от условия.

Циклический

Циклический алгоритм, в котором последовательность команд выполняется несколько раз в зависимости от заданной величины.

Рис. 5.13. Классификация алгоритмов по процессам

Напишите программу «Високосный год», в которой нужно определить, является ли год с данным номером високосным.

**Шаг за шагом**

1. Постановка проблемы. Дано натуральное число. Требуется определить, является ли год с данным номером високосным. Если год является високосным, то выведите **високосный год**, иначе выведите **обычный год**.

2. **Формализация.** В соответствии с григорианским календарем год является високосным, если его номер кратен **4**, но не кратен **100**, а также, если он кратен **400**. То есть если остаток от деления на **4** не равен нулю, значит, год не делится нацело на **4** и не является високосным. Если год – столетие, но не делится на **400**, то этот год обычный. Алгоритм решения этой задачи является разветвляющимся.

3. **Блок-схема** (рис. 5.14) определения високосного года.

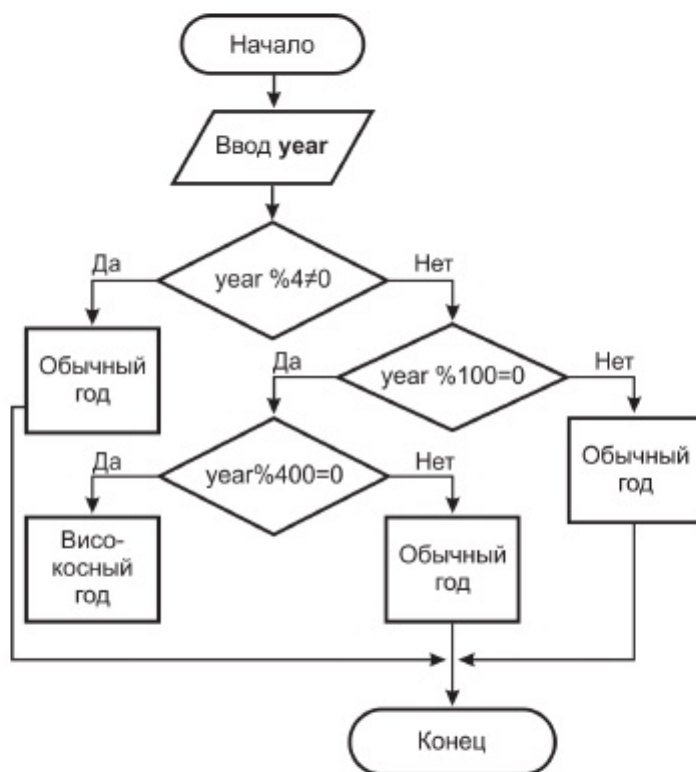


Рис. 5.14. Блок-схема «Високосный год»

4. **Код программы** решения задачи (рис. 5.15).

```

File Edit Format Run Options Window Help
# Вводится год, преобразуется к целому числу
year = int(input())

# Если остаток от деления на 4 не равен нулю,
# значит год не делится нацело на 4 и
# не является високосным, т. е. он обычный.
if year % 4 != 0:
    print("обычный год")
# Исключаем столетия, которые не делятся на 400
elif year % 100 == 0: # является ли столетием?
    if year % 400 == 0: # Делится ли на 400?
        # В таком случае год високосный
        print("високосный год")
    else: # Если столетие, но не делится на 400,
        # то год обычный
        print("обычный год")
# Во всех остальных случаях год високосный
else:
    print("високосный год")

===== RESTART:
2019
обычный год
>>>
===== RESTART:
2021
обычный год
>>>
===== RESTART:
2000
високосный год
>>>
===== RESTART:
1968
високосный год
>>>
===== RESTART:
1999
обычный год
>>>
===== RESTART:
300
обычный год
>>>

```

Рис. 5.15. Код программы «Високосный год»

5. Тестирование и отладка программы. Для тестирования подберите годы, которые являются как високосными, так и обычными.

Знание. Понимание



1. Произнесите на трех языках ключевые слова.
2. Что такое программа?
3. Какие типы алгоритмов вам известны?
4. Для чего составляются комментарии? Приведите примеры.
5. Какие требования к идентификаторам нужно соблюдать?
6. Что такое ключевые слова? Приведите примеры.





Применение. Анализ



7. Вычислите стоимость покупки с учетом скидки: при покупке товара на сумму больше 5000 тенге предоставляется скидка 10%. Проанализируйте блок-схему задачи (рис. 5.16). Найдите ошибку. Составьте программу решения задачи.

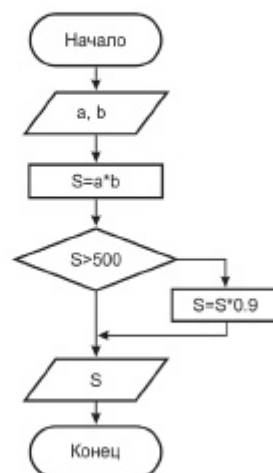


Рис. 5.16. Блок-схема задачи

8. Рассмотрите блок-схему на рис. 5.17. Составьте условие задачи. Напишите код программы.

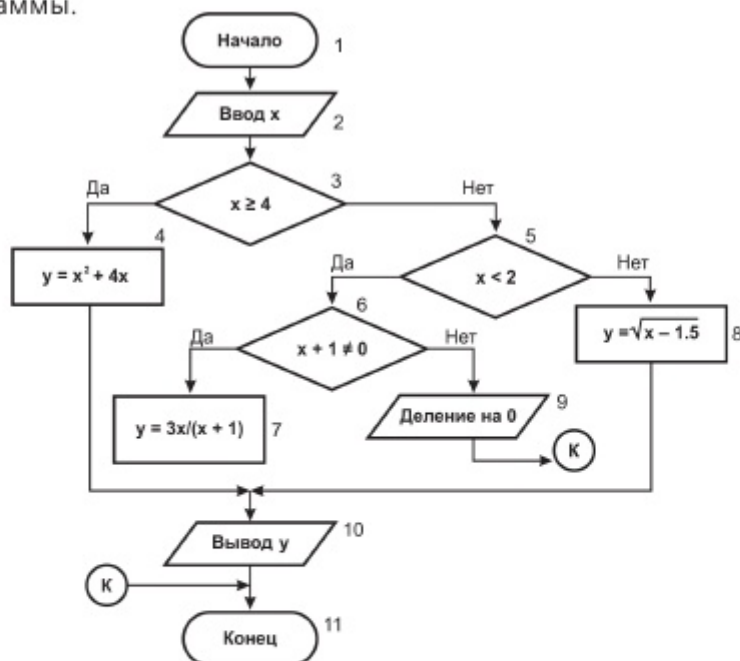


Рис. 5.17. Блок-схема для составления условия задачи

9. Даны три целых числа. Выведите значение наименьшего из них. Проанализируйте код программы (рис. 5.18). Составьте блок-схему. Напишите комментарии к программе.

```

File Edit Format Run Options Window Help
x = int(input())
y = int(input())
z = int(input())
if x < y:
    if x < z:           #
        print(x)
    else:              #
        print(z)
else:
    if y < z:           #
        print(y)
    else:              #
        print(z)

```

Рис. 5.18. Код программы с комментариями

Синтез. Оценивание



10. Составьте алгоритм и программу решения задачи «Шахматная доска». Заданы две клетки шахматной доски. Если они покрашены в один цвет, то выведите слово **YES**, а если в разные цвета – то **NO**. Программа получает на вход четыре числа от **1** до **8**, которые задают номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Проведите тестирование задачи. Оцените полученный результат.



11. Самостоятельно придумайте программу, в которой используется инструкция **if** с веткой **else**. Вложенный код должен содержать не менее трех выражений. Предложите однокласснику составить блок-схему и программу решения. Оцените полученный результат.

Таблица 5.6. Операторы присваивания в Python



Оператор	Описание	Пример
=	Присваивает значение правого операнда левому	c = 27 присвоит переменной c значение 27

Продолжение таблицы 5.6

Оператор	Описание	Пример
<code>+=</code>	Прибавит значение правого операнда к левому и присвоит эту сумму левому операнду	<code>c=25; a=23; c+=a</code> равносильно <code>c=c+a</code> . c будет равна 48
<code>--</code>	Отнимает значение правого операнда от левого и присваивает результат левому операнду	<code>c--=a</code> равносильно <code>c=c-a</code> . c будет равна 2
<code>*=</code>	Умножает правый операнд на левый и присваивает результат левому операнду	<code>c*=a</code> равносильно <code>c=c*a</code> . c будет равна 575
<code>/=</code>	Делит левый операнд на правый и присваивает результат левому операнду	<code>c=100; a=20; c/=a</code> равносильно <code>c=c/a</code> . c будет равна 5
<code>%=</code>	Делит по модулю операнды и присваивает результат левому	<code>c=5; a=2; c%=a</code> равносильно <code>c=c%a</code> . c будет равна 1
<code>**=</code>	Возводит левый операнд в степень правого и присваивает результат левому операнду	<code>c=3; a=2; c**=a</code> равносильно <code>c=c**a</code> . c будет равна 9
<code>//=</code>	Производит целочисленное деление левого операнда на правый и присваивает результат левому операнду	<code>c=11; a=2; c//=a</code> равносильно <code>c=c//a</code> . c будет равна 5

5.4. ТЕСТИРОВАНИЕ ПРОГРАММЫ

ВЫ НАУЧИТЕСЬ

Осуществлять проверку программы с помощью тестовых данных.

КЛЮЧЕВЫЕ СЛОВА

Тестирование	Тестілеу	Testing
Трассировка	Із	Trace
Верификация	Растан	Verification



Что такое тестирование и отладка программы?

В чем сходство и различия между ними?

Для чего выполняют тестирование программы?



При решении задачи на компьютере необходимым этапом является доказательство правильности работы программы. Для этого проводят тестирование программы.

Тестирование (англ. *test* – «испытание») – это испытание, проверка правильности работы программы в целом либо ее составных частей.



Цель тестирования – найти в программе ошибки. При тестировании необходимо в программу вводить различные входные данные.

Если программа выдает результат, который соответствует тестовым данным, то программа написана верно. Этот процесс называется **верификацией**.

Отладка программы – процесс поиска и устранения ошибок в программе, который производится после ее запуска на компьютере.

Термины «тестирование» и «отладка» часто используют вместе. Но нужно различать, что **тестирование** – это средство обнаружения ошибок, а **отладка** является средством поиска и устранения причин уже обнаруженных ошибок.

Для тестирования предварительно должны быть подготовлены тесты, обеспечивающие проверку всех возможных условий возникновения ошибок:

- на начальном этапе нужно использовать простые тестовые данные для обеспечения проверки работы программы;
- должны быть упрощены арифметические операции для снижения объема вычислений в тестах;
- необходимо постепенно усложнять тестовые данные;
- нужно проверить работоспособность программы, не содержащей ошибок.

Особенности отладки:

- при отладке происходит исправление синтаксических ошибок и ошибок кодирования;
- ошибки, которые возникают, необходимо устранять по очереди;
- следует помнить, что главная причина ошибок – человек, а не компьютер.

Отладку можно представить в виде многократного повторения трех процессов в виде формулы (рис. 5.18).

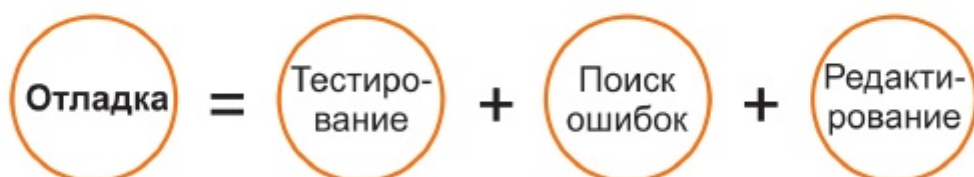


Рис. 5.18. Процесс отладки

На этапе тестирования будет констатировано наличие ошибки, на втором этапе происходит поиск места ошибки в программе, а на третьем – редактирование программы с целью устранения обнаруженной ошибки.

Одним из способов тестирования программы является **трассировка**. Для выполнения трассировки нужно пошагово выполнить алгоритм и записать все значения переменных в таблицу. Полученная таблица называется **трассировочной**. Трассировочная таблица является моделью работы процессора при выполнении программы человеком вручную.

Трассировка программы вручную

1. Сформировать таблицу, состоящую из названия столбцов: команды алгоритма, все переменные и выполняемые действия.
2. Выписать все операторы программы (алгоритма).
3. Выписать начальные значения переменных.
4. Заполнить таблицу значениями (табл. 5.7).

Таблица 5.7. Трассировка программы

№	Команда алгоритма	Переменные			Выполняемые действия
1					
2					
3					
4					
5					

Даны два числа. Удвойте первое число, если оно больше второго, и оба числа, если это не так.



Шаг за шагом



1. Постановка проблемы. Пусть исходными данными являются переменные **a** и **b**. Их значения будут вводиться с клавиатуры. Необходимо сравнить эти числа и удвоить число **a**, если оно больше **b**. В этом случае число **b** нужно оставить без изменения. Если число **a** меньше **b**, то нужно удвоить оба числа.

2. Формализация. Вводим два вещественных числа **a** и **b**. Если $a > b$, то $a = 2 * a$, $b = b$, иначе $a = 2 * a$ и $b = 2 * b$.

3. Блок-схема «Сравнение и замена чисел» (рис. 5.19).

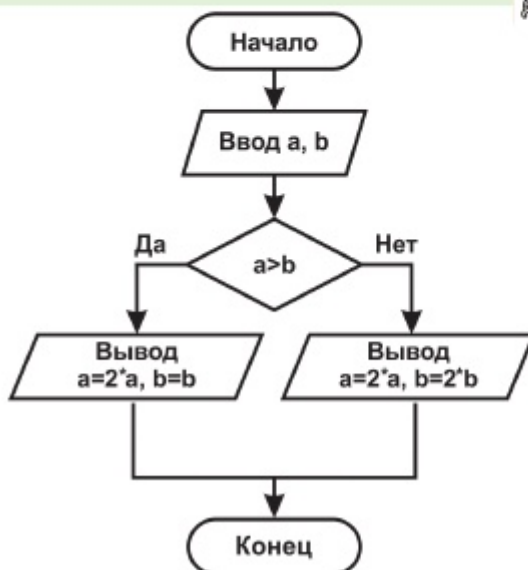


Рис. 5.19. Блок-схема «Сравнение и замена чисел»

4. Код программы решения задачи показан на рис. 5.20.

```

File Edit Format Run Options Window Help
a = int(input("Введите число a: "))
b = int(input("Введите число b: "))

if a > b:
    print("a=", 2*a, "b=", b)
else:
    print("a=", 2*a, "b=", 2*b)
  
```

```

>>>
===== RESTART: C:/1
Введите число a: 6
Введите число b: 2
a= 12 b= 2
>>>
===== RESTART: C:/1
Введите число a: 10
Введите число b: 20
a= 20 b= 40
>>>
===== RESTART: C:/1
Введите число a: 50
Введите число b: 50
a= 100 b= 100
>>> |
  
```

Рис. 5.20. Код программы «Сравнение и замена чисел»

5. Тестирование и отладка программы.

Проведем трассировку программы при значениях переменных: $a > b$, $a < b$, $a = b$ (табл. 5.8–5.10):

Таблица 5.8. Трассировка программы при $a > b$

№	Команда алгоритма	Переменные		Выполняемые действия
		a	b	
1	Ввод a, b	6	2	
2	$a > b$			$6 > 2$ (true)
3	$a = 2 * a$			$a = 2 * 6$
4	$b = b$			$b = 2$

Таблица 5.9. Трассировка программы при $a < b$

№	Команда алгоритма	Переменные		Выполняемые действия
		a	b	
1	Ввод a, b	10	20	
2	$a > b$			$10 > 20$ (false)
3	$a = 2 * a$			$a = 2 * 10$
4	$b = 2 * b$			$b = 2 * 20$

Таблица 5.10. Трассировка программы при $a = b$

№	Команда алгоритма	Переменные		Выполняемые действия
		a	b	
1	Ввод a, b	50	50	
2	$a > b$			$50 > 50$ (false)
3	$a = 2 * a$			$a = 2 * 50$
4	$b = 2 * b$			$b = 2 * 50$

Знание. Понимание

1. Произнесите на трех языках ключевые слова.
2. Какова цель тестирования и отладки программы?
3. В чем сходство и различия тестирования и отладки?
4. Что означает термин «верификация»?
5. Как выполняется трассировка алгоритма?
6. По каким правилам составляется трассировочная таблица?

**Применение. Анализ**

7. Даны три числа. Если их произведение меньше **100**, то числа увеличить в 5 раз. В противном случае оставить числа без изменения. Составьте программу решения задачи. Заполните трассировочную таблицу для значений: 1) **a = 2, b = 5, c = 8**; 2) **a = -10, b = -5, c = 4**; 3) **a = 10, b = -5, c = -2**. Проанализируйте результат.



8. Ученик составил программу решения задачи и при тестировании получил следующие результаты (см. рис. 5.21). Найдите и исправьте ошибку в программе, проведите ее тестирование. Составьте трассировочную таблицу.

Задача. В школу программистов принимают учащихся от 10 до 13 лет и набравших на экзамене более 50 баллов. Определите, будет ли поступающий принят в школу программистов.

<pre>File Edit Format Run Options Window Help a = int(input("введите возраст ученика")) b = int(input("введите количество баллов")) if a>10 and a<13 and b>50: print("принят") else: print("не принят")</pre>	<pre>введите возраст ученика12 введите количество баллов51 принят >>> ===== RESTART: F:/7класс_ введите возраст ученика10 введите количество баллов55 не принят >>> ===== RESTART: F:/7класс_ введите возраст ученика13 введите количество баллов60 не принят >>></pre>
--	--

Рис. 5.21. Программа задачи «Школа программистов»

Таблица 5.11

Трассировочная таблица № 1 задачи «Школа программистов»

№	Команда алгоритма	Переменные		Выполняемые действия
		a	b	
1	Ввод a, b	12	51	
2	a > 10 and a < 13 and b > 50			12 > 10 and 12 < 13 and 51 > 50 (true)
3	Вывод			принят

Таблица 5.12

Трассировочная таблица № 2 задачи «Школа программистов»

№	Команда алгоритма	Переменные		Выполняемые действия
		a	b	
1	Ввод a, b	10	55	
2	a > 10 and a < 13 and b > 50			10 > 10 and 10 < 13 and 55 > 50 (false)
3	Вывод			не принят

Таблица 5.13

Трассировочная таблица № 3 задачи «Школа программистов»

№	Команда алгоритма	Переменные		Выполняемые действия
		a	b	
1	Ввод a, b	13	60	
2	a > 10 and a < 13 and b > 50			13 > 10 and 13 < 13 and 60 > 50 (false)
3	Вывод			не принят

9. Сколько тестов потребуется для минимального подтверждения работоспособности функции?

$$f(x) = \begin{cases} x^2 + 2x + 9, & \text{при } x \leq 3 \\ \frac{1}{x^3 + 6}, & \text{при } x > 3 \end{cases}$$

Составьте программу и трассировочную таблицу.

Синтез. Оценивание



10. Составьте программу, проверяющую уровень знаний по информатике. Задайте 3–5 вопросов. Подсчитайте количество правильных ответов. Заполните табл. 5.14.



Таблица 5.14. «Что я хочу протестировать и как?»

Проверяемая функция программы	Тестовые данные (действия пользователя)	Реакция программы (результат работы)	Выявленные ошибки/внесенные изменения

11. Сравните процессы тестирования и отладки. Наиболее принципиальные позиции зафиксируйте в табл. 5.15.

Таблица 5.15. Сравнение процессов тестирования и отладки

Основания для сравнения	Тестирование	Отладка
Назначение процесса		
Инструменты выполнения процесса		
Методы (способы) выполнения процесса		
Ошибки, выявляемые в процессе		



Документирование программы

Документирование – это неотъемлемая часть каждого шага программирования. Его используют для облегчения программирования и включения описания для результирующей программы. Процесс документирования сопровождается:

- постановку задачи;
- документы проекта;
- алгоритмы и программы.

Большой объем разнообразной внешней документации необходим при передаче информации между разработчиками программы. Например, **пользовательская документация** программы объясняет, как пользователи должны действовать, используя данную программу.

Внутренняя документация, включенная в программу, облегчает чтение кода. Описание разработки программы называют **документацией по сопровождению программы**. С помощью этой документации можно изучить, как сконструирована программа.

ЗАДАНИЯ К РАЗДЕЛУ V

1. Установите очередность действий при постановке проблемы.

- Дать точную формулировку задачи.
- Предложить идею решения задачи.
- Описать исходные данные и способы их хранения.
- Определить форму выдачи результатов.
- Определить цель.
- Определить необходимый объем информации.

2. Приведите примеры на основные свойства алгоритмов.

- 1) Понятность.
- 2) Дискретность.
- 3) Определенность.
- 4) Результативность.
- 5) Массовость.

3. Помните сказку, в которой на камне было написано:

*«Налево пойдешь – коня потеряешь,
Направо пойдешь – жизнь потеряешь,
Прямо пойдешь – счастье найдешь!».*

Напишите программу с применением ветвления.

4. Что будет выведено на печать по коду программы?

```
a = 5 < 6
b = 6 >= 7
c = 5 != 5
d = 5 == 15
print (a,b,c,d)
```

```
if a and b:
    print ("A")
else:
    print ("B")
```

5. Можно ли составной оператор *if age == 10 or age == 11 or age == 12 or age == 13*: записать компактнее, воспользовавшись ключевым словом *and*, а также операторами «больше или равно» (\geq)?

6. Дан код программы:

```
pH=float(input())
if pH == 8.0:
    print(pH, "Вода")
elif 7.36 < pH < 7.44:
    print(pH, "Кровь")
```

Напишите к ней постановку задачи и составьте блок-схему.

7. Выведите три целых числа в убывающем порядке. Проведите формализацию, создайте алгоритм, напишите код программы, проведите тестирование и отладку.

ГЛОССАРИЙ

Автозаполнение – это операция ускоренного автоматического заполнения ячеек данными, текстом, числами или формулами без использования обычных способов ввода.

Алгоритм – это точное описание упорядоченной последовательности действий, приводящей за конечное число шагов к необходимому результату.

Архив – это файл, содержащий в себе один или несколько файлов в сжатом виде.

Блок-схема – графическое изображение алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций.

Ветвление – это алгоритмическая конструкция, в которой, в зависимости от некоторого условия, происходит исполнение одной из двух последовательностей действий (ветвей).

Видеопамять (VRAM) – это разновидность оперативного запоминающего устройства, в котором хранятся закодированные изображения.

Вирус – это программа, которая активизируется, копируя саму себя в исполняемые объекты. В процессе выполнения исходный код частями преобразуется в машинный специальной программой – **интерпретатором**.

Вложенное ветвление – это ветвление, находящееся в одной из ветвей другого ветвления.

Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных. Целостность ее содержимого не зависит от того, включен или выключен компьютер, и не имеет прямой связи с процессором.

Вредоносная программа (Malware – сокращение от «malicious software») – любое программное обеспечение, специально созданное для того, чтобы причинять ущерб компьютеру, серверу или компьютерной сети и хранящимся на них данным.

Диаграмма – это визуальное представление числовых значений.

Интерфейс (от англ. *interface* – место соприкосновения) – совокупность средств и методов взаимодействия между элементами системы.

Информационная модель – это описание моделируемого объекта на одном из языков кодирования информации.

Канал связи – технические средства, позволяющие осуществлять передачу данных на расстоянии.

Компьютер-сервер – это центральный компьютер, который распределяет общие ресурсы между многими пользователями.

Компьютерная память – это устройство хранения информации, или запоминающее устройство. Память является обязательной частью компьютера.

Компьютерные сети – это объединение компьютеров для обмена информацией и совместного использования ресурсов.

Кэш-память, или сверхоперативная память, – очень быстрое запоминающее устройство небольшого объема. Она используется при обмене данными между микропроцессором и оперативной памятью.

Логическое высказывание – это утверждение, истинность которого можно определить однозначно.

Маркер заполнения – небольшой черный квадрат в правом нижнем углу выделенной ячейки или выделенного диапазона.

Моделирование – это процесс построения моделей для исследования и изучения объектов, процессов или явлений.

Оперативная память (*Random Access Memory* – RAM) – это быстрое запоминающее устройство не очень большого объема, связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Отладка программы – процесс поиска и устранения ошибок в программе, который производится после ее прогона на компьютере.

Пользовательский интерфейс (на англ. *user interface* или сокращенно *UI*) – система правил и средств, обеспечивающая взаимодействие программы с пользователем.

Последовательности – это введенные в соседние ячейки данные, представляющие собой некоторую закономерную последовательность.

Постоянная память (ПЗУ, англ. ROM, *Read Only Memory* – память только для чтения) – энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения.

Проблема – это форма научного отображения проблемной ситуации.

Проблемная ситуация – это затруднение, которое возникает из-за несоответствия между имеющимися знаниями и теми, которые необходимы для решения возникшей задачи.

Программа – запись алгоритма на машинном языке, понятном компьютеру.

Программы-шпионы устанавливаются скрытым путем на компьютеры для сбора информации о пользователе без его ведома.

Прототип – это объект, процесс или явление, на основе которого создается модель.

Разветвляющийся алгоритм – это алгоритм, содержащий конструкцию ветвления.

Распаковка архива – процесс восстановления сжатых файлов.

Сервер – абонент сети, отдающий в сеть свой ресурс и имеющий или не имеющий доступ к ресурсам сети.

Создание архива – процесс сжатия файлов.

Таблица – это объект, который добавляется в документ для структурирования данных.

Тестирование (англ. *test* – испытание) – это испытание, проверка правильности работы программы в целом либо ее составных частей.

Топология характеризует физическое расположение компьютеров, кабелей и других компонентов сети.

Троянские программы – это вредоносные программы, выполняющие несанкционированные пользователем действия.

Файл – это определенное количество информации (программа или данные), имеющее имя и хранящееся в долговременной (внешней) памяти.

Формализация – процесс построения информационных моделей с помощью формальных языков программирования.

Форматирование – это изменение свойств данных, определяющих внешний вид, без изменения содержимого.

Формула – специальный инструмент в электронной таблице Excel, предназначенный для расчетов, вычислений и анализа данных.

Черви – это программы, которые распространяют свои копии по сети.

Электронная таблица Excel – это приложение, позволяющее автоматизировать обработку данных, представленных в виде таблиц.

Эргономичность – способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях.

Юзабилити (от англ. *usability*) – удобство использования, пригодность использования, эргономичность.

Как геймификация помогает обучению

Как компьютерные игры применять с пользой для образования? Как из большого многообразия игр выбрать те, которые будут мотивировать и вовлекать в образовательный процесс? Как с помощью игр превратить решение скучных и сложных задач в интересные и простые?

Игра в обучении – инструмент, а геймификация – это процесс, направленный на достижение образовательных целей. В компьютерных играх основным принципом достижения успеха является «учиться на своих ошибках». Этот принцип учит концентрироваться на самих знаниях и содержании изучаемого материала.

В геймификации используются такие игровые элементы, как постановка задач, обратная связь, уровни, творчество. Игровые правила позволяют достичь реальных целей обучения. Кроме этого, используется система оценок, которая основана на следующих тезисах:

- система оценок в баллах;
- все ученики начинают с «нулевым баллом» с начала учебного года;
- каждое задание оценивается определенным количеством баллов.

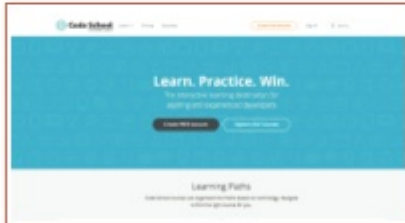
Очки и баллы являются стимулом для дальнейшего обучения и пути к хорошей оценке, успеху. Значимым фактором игрового формата обучения является командная работа. Командное взаимодействие вовлекает в процесс обучения и изучения материала. Поэтому, независимо от своих способностей, каждый находится в равных условиях с остальными.

Приведем примеры сервисов, которые используют геймификацию для образования.



Codecademy

Обучение программированию на JavaScript, HTML, Python, Ruby



Code School

Сервис для обучения программированию с элементами геймификации



Motion Math Games

Мобильные игры по математике делают обучение веселым и захватывающим



Mathletics

Программа для школ, направленная на привлечение детей к математике через игры и челленджи



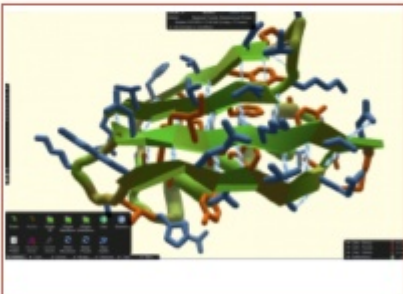
Khanacademy

Большое количество бесплатных видеокурсов, видеоресурсов по различным предметам



Spongelab

Платформа для персонализированного научного образования



Foldit

Решение научных задач как пазлов

В настоящее время много людей играет в MMORPG-игры (англ. *massively multiplayer online role-playing game*). Это многопользовательские ролевые онлайн-игры. Примером такой игры является **World of Warcraft**. Сеанс игры может заменять сразу несколько уроков – математику, русский и английский языки, литературу, этику. Игра учит работать в команде, развивает навыки общения с помощью компьютера и технологическую готовность, доброжелательность, открытость опыту, сознательность.

Игры в альтернативной реальности (*Alternative Reality Games*) – интерактивное повествование с игровыми элементами. В них используется в качестве платформы реальный мир. ARG-игры усиливают мотивацию к изучению нового и неизведанного. Примером ARG является игра **Year Zero**. Это тактическая приключенческая игра, сочетающая в себе пошаговое исследование мира в реальном времени с увлекательным сюжетом, скрытностью и стратегией.

Остановимся на примере сервиса **Академия Хана**. Геймификация здесь заключается в наличии инструмента, который показывает их прогресс, что именно они изучили, а что еще предстоит изучить. Также есть возможность устанавливать собственные цели в обучении для внутренней мотивации.

На платформе **Mozaweb** можно выбрать образовательные инструменты, игры, воспроизводить выбранные образовательные видео или 3D-сцены, такие как работа с компьютерной сетью, устройство компьютера, история развития носителей информации, устройство жесткого диска и др. (рис. 1).

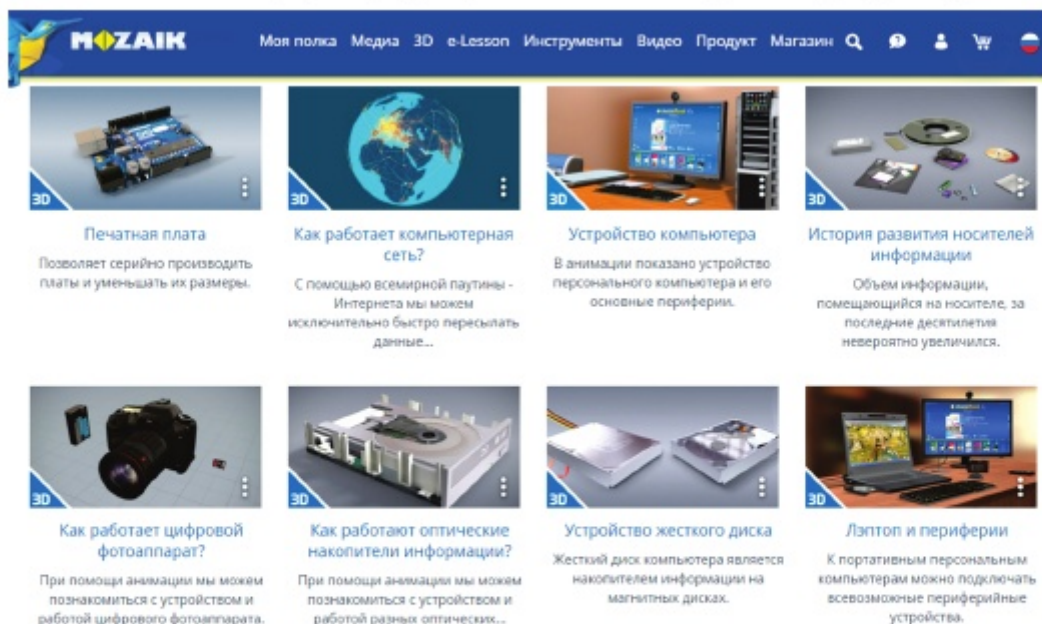


Рис. 1. Платформа Mozaweb

Онлайн-конструктор дидактических игр

Umaigra (UI) представляет собой интернет-проект дистанционного обучения. Онлайн-конструктор дидактических игр (*Umapalata.com*) предлагает новую онлайн-систему для создания, публикации и выполнения дидактических игр для учащихся (рис. 2).



Рис. 2. Umaigra

Решение базируется на представлении обучения через игру как процесса, состоящего из следующих этапов:

- А) подготовка материала, создание и публикация игры;
- В) представление игры ученикам в виде одного или более заданий;
- С) выполнение заданий учащимися через одну или несколько попыток;
- Д) контроль за ходом выполнения заданий отдельными учениками или группами учеников и вывод результатов;
- Е) анализ результатов, выявление проблем в выполнении заданий и возвращение к этапу А.

Использование электронных игр не заменяет традиционные уроки, а предоставляет дополнительную возможность в обучении. Основные характеристики, которыми должна обладать электронная игра для ее эффективного использования:

- простой и понятный интерфейс с графикой;
- занимательный игровой сценарий, ориентированный на четко определенные образовательные цели;
- возможность адаптировать игру к индивидуальным потребностям учащихся;
- возможность персонализировать содержание для различных предметных областей, возрастного диапазона и языка учащихся;
- показ во время игры параметров выполнения: прогресс игрока, счет, время и т.д.;
- возможность отслеживания, оценки и анализа результатов учащихся;
- возможность публикации и обмена игрой с сообществом учителей и учащихся и т.д.

UI может быть легко интегрирован в основной учебный процесс в качестве дополнительного обучающего инструмента. Конструктор можно использовать в школе и дома, индивидуально и для группы учеников.

UI предлагается в двух версиях: **Editor** и **Class**.

UI Editor дает возможность создавать игры на прототипах за несколько шагов, подготовленных для различных предметных областей, языков, разных возрастов и видов упражнений.

Приведем несколько готовых игр.

1. **«Футбол +»**. Данная игра позволяет подготовить:

- альтернативные тесты (Правильно – неправильно);
- тесты с выбором ответа.

2. **«Заколдованный замок»**. В этом прототипе активно используется технология Drag-and-drop (перетаскивание объектов). Это позволяет удалять слова и строить предложения.

3. **«Авиаперелет»**:

- конструирование альтернативных тестов (Правильно – неправильно);
- тесты с выбором ответа.

Конструктор позволяет включить/выключить звуковое сопровождение, определить количество заданий, скорость их решения. Многие из дидактических игр можно использовать не только для дистанционного обучения, но и в школе на интерактивной доске.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. *Банкрашков А. В.* Программирование для детей на языке Python. – М.: АСТ, 2018. – 288 с.
2. *Вордерман К.* Программирование на Python. Иллюстрированное руководство для детей / К. Вордерман, К. Стили, К. Квигли. – М.: Манн, Иванов и Фербер, 2017. – 346 с.
3. *Горячев А. В.* Практикум по информационным технологиям / А. В. Горячев, Ю. А. Шафрин. – М.: Бином, 2016. – 272 с.
4. Информатика: Энциклопедический словарь для начинающих / ред. Д. А. Поспелов. – М.: Педагогика-Пресс, 2013. – 352 с.
5. *Леонтьев В. П.* Новейшая энциклопедия персонального компьютера 2008. – М.: ОЛМА Медиа Групп, 2008. – 960 с.
6. *МакГрат М.* Программирование на Python для начинающих. – М.: Эксмо, 2015. – 192 с.
7. *Петроченков А. В.* Персональный компьютер – просто и ясно! – М.: Смоленск: Русич, 2013. – 400 с.
8. *Семакин И.* Информатика. Базовый курс. 7–9 классы / И. Семакин, Л. Залогова, С. Русаков. – М.: Бином. Лаборатория знаний, 2015. – 390 с.
9. *Саммерфилд М.* Программирование на Python 3. Подробное руководство. – М.: Символ, 2016. – 608 с.
10. *Семакин И. Г., Залогова Л. А., Русаков С. В., Шестакова Л. В.* Информатика и ИКТ. 7 класс. – М.: Бином. Лаборатория знаний, 2012. – 167 с.
11. *Угринович Н. Д.* Информатика и ИКТ. 7 класс. – М.: Бином. Лаборатория знаний, 2012.

ИНТЕРНЕТ-РЕСУРСЫ

1. <https://www.360totalsecurity.com/ru/antivirus-on-the-computer-download/>
2. <https://ru.wikipedia.org/wiki>.
3. Информатика. Теория (с задачами и решениями) [<http://book.kbsu.ru/theory/index.html>] дата обращ.: 10.11.2020.
4. Алексеев, Е. Г. Информатика. Мультимедийный электронный учебник / Е. Г. Алексеев, С. Д. Богатырев [<http://inf.e-alekseev.ru/text/toc.html>] дата обращ.: 10.11.2020.
5. Словарь терминов по информатике [http://uchu2008.narod.ru/razdely/informatika/slovar_terminov.html] дата обращ.: 10.11.2020.
6. Словарь основных понятий и терминов [<http://book.kbsu.ru/theory/definition.html>] дата обращ.: 10.11.2020.
7. Словарь [<http://pandia.ru/text/78/419/69850.php>] дата обращ.: 10.11.2020.
8. Словарь терминов по информатике [http://wiki.iteach.ru/images/5/5d/Словарь_терминов.pdf]
9. <https://kopilkaurokov.ru/informatika/uroki/modielirovaniie-v-excel> дата обращ.: 10.11.2020.

Учебное издание

**Мухамбетжанова Сауле Талапеденовна
Тен Алира Сунтаковна
Демидова Лариса Геннадьевна**

ИНФОРМАТИКА

Учебник для 7 класса общеобразовательной школы

Зав. редакцией *Н. Жиенгалиев*

Редактор *Л. Холина*

Художники *А. Лукманов, А. Айдархан*

Технический редактор *О. Рысалиева*

Компьютерная верстка *Э. Омаровой*

ИБ № 034

Сдано в набор 11.02.2021. Подписано в печать 09.07.2021. Формат 70x90 ¹/₁₆.
Бумага офсетная. Гарнитура «Школьная». Печать офсетная. Усл. печ. л. 15,21. Уч.-изд. л. 9,98.
Тираж 9700 экз. Заказ № 5997.

ТОО «Корпорация «Атамұра», 050000, г. Алматы, пр. Абылай хана, 75.

Полиграфкомбинат ТОО «Корпорация «Атамұра» Республики Казахстан,
050002, г. Алматы, ул. М. Макатаева, 41.

