

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

ИНФОРМАТИКА

Учебник для 8 класса общеобразовательной школы

8

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 учебника

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

Мухамбетжанова С. Т.

М92 Информатика: Учебник для 8 кл. общеобразоват. шк. / С. Т. Мухамбетжанова, А. С. Тен, Л. Г. Демидова. – Алматы: Атамұра, 2021. – 176 с.

ISBN 978-601-331-944-5

ISBN 978-601-331-944-5

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

Содержание

Введение	5
Раздел I	
Технические характеристики компьютера и сетей	
1.1. Измерение информации	7
1.2. Процессор и его характеристики	11
1.3. Компьютерные сети.....	18
Дополнительные задания к первому разделу.....	27
Тестовые задания к разделу I.....	27
Раздел II	
Здоровье и безопасность	
2.1. Негативные аспекты использования компьютера.....	30
2.2. Безопасность в сети	37
Дополнительные задания ко второму разделу.....	44
Тестовые задания к разделу II.....	45
Раздел III	
Обработка информации в электронных таблицах	
3.1. Статистические данные	47
3.2. Встроенные функции	57
3.3. Анализ данных на основе имеющейся информации	66
3.4. Решение прикладных задач	75
Дополнительные задания к третьему разделу.....	90
Тестовые задания к разделу III.....	92
Раздел IV	
Программирование алгоритмов на языке программирования Python	
4.1. Цикл while	100
4.2. Цикл for	106
4.3. Управление циклом: break	112
4.4. Управление циклом: continue	117

4.5. Управление циклом: else	120
4.6. Трассировка алгоритма.....	126
Тестовые задания к разделу IV.....	135

Раздел V

Практическое программирование

5.1. Постановка проблемы.....	139
5.2. Разработка алгоритма.....	145
5.3. Программирование алгоритма.....	150
5.4. Тестирование программы.....	157
Тестовые задания к разделу V.....	165
Глоссарий.....	169
Список использованной литературы.....	173
Интернет-ресурсы.....	174

ВВЕДЕНИЕ

Дорогие восьмиклассники!

Вы продолжаете изучать сложный, но интересный предмет «Информатика». Для вас разработаны учебник и электронное приложение. Электронное приложение – это учебный материал к учебнику и разнообразные интерактивные задания, при выполнении которых нужно перемещать объекты, вводить числа, выбирать ответы, устанавливать соответствия, вводить строки и т.д.

Условные обозначения, данные в начале учебника, позволяют 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 информатики для 8 класса написан в соответствии с Государственным общеобязательным стандартом образования. Основными целями изучения предмета информатики являются:

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

В процессе реализации целей обучения вы научитесь измерять информацию с помощью алфавитного подхода, рассчитывать пропускную способность сети, преобразовывать информацию посредством компьютеров и т.д.

Прикладной характер учебника информатики позволит формировать и развивать навыки эффективного и правильного использования информационных технологий (электронной таблицы) как в учебной деятельности, так и в повседневной жизни.

Основное внимание в учебнике уделяется не только изучению методов и процессов преобразования информации с помощью компьютеров, но и умению соблюдать этикет и безопасность при работе в Интернете, эффективно защищать себя от электромагнитного излучения компьютера и других электронных устройств.

Принцип «спиральности» учебника позволяет рассматривать некоторые темы и цели обучения снова и снова. Но это происходит с постепенным углублением, усложнением и наращиванием объема знаний и навыков. Так, например, в 8 классе вы будете продолжать изучать среду программирования Python, углубите свои знания по данной теме, а также будете решать более сложные, чем в 6–7 классах, задачи.

Вы должны помнить, что без программирования были бы невозможны все передовые технологии – от квантовых компьютеров до космических программ. В реальной жизни не все вы станете программистами. Но умение писать программы поднимет вас на более высокий уровень – творческий, интеллектуальный.

Весь учебный материал структурирован и разделен на разделы. Каждый раздел состоит из параграфов. К каждой теме приведены задания, ранжированные по шести уровням таксономии Б. Блума: знание, понимание, применение, анализ, синтез и оценивание. Разноуровневые задания в свою очередь распределены по уровням мышления и развития. **Низкий уровень** включает задания на «Знание. Понимание». Задания на «Применение» и «Анализ» относятся к **среднему уровню**. Соответственно, **высокий уровень** включает задания из разделов «Синтез» и «Оценивание».

Прикладные задания помогут вам развивать функциональную грамотность, то есть критически работать с информацией (проблемой), представленной в таблицах, графиках, инфографике, быстро понимать ее, извлекать, анализировать, осмысливать, интерпретировать, аргументировать и оценивать.

Кроме того, рекомендации для выполнения домашних заданий позволят облегчить работу дома. Рубрика «Для тех, кто хочет знать больше» познакомит с дополнительным и интересным материалом.

Таким образом, учебник реализует связь обучения с жизнью и практикой, актуализирует знания в учебной деятельности и жизненных ситуациях.

Желаем вам успехов в изучении самой интересной и современной науки – информатики!

Авторы

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ КОМПЬЮТЕРА И СЕТЕЙ

1.1. ИЗМЕРЕНИЕ ИНФОРМАЦИИ

ВЫ НАУЧИТЕСЬ

Применять алфавитный подход при определении количества информации.

КЛЮЧЕВЫЕ СЛОВА

Алфавитный подход	Алфавиттік тәсіл	Alphabetical approach
Количество информации	Ақпарат мөлшері	Quantity of information



Единица измерения информации была определена в науке, которая называется **теорией информации**. Для измерения различных величин применяют эталонные единицы измерения, как показано на рисунке 1.1.

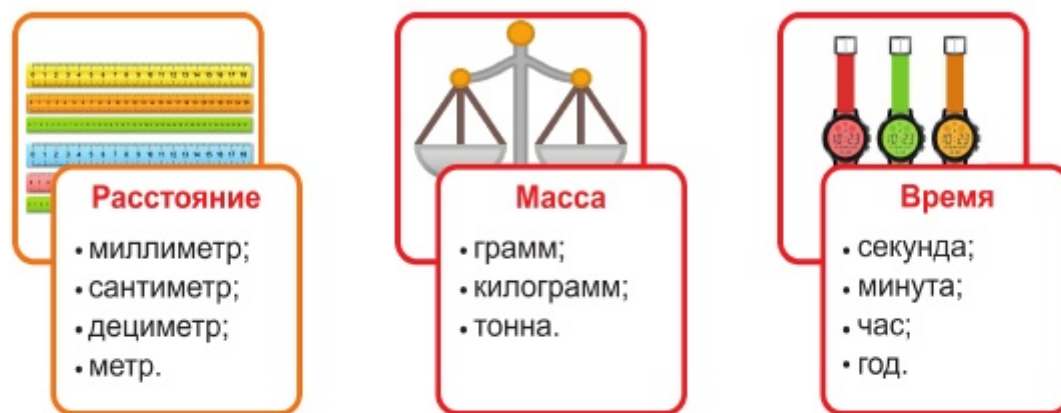


Рис. 1.1. Эталонные единицы измерения

Для измерения информации введена своя эталонная единица – **бит**.

Восемь последовательных битов составляют **байт**. В одном байте можно закодировать значение одного символа из 256 возможных ($256 = 2^8$).



Какие более крупные единицы измерения количества информации используются наряду с битами и байтами?

Определить количество информации можно по-разному. Существуют два подхода к измерению информации: содержательный и алфавитный (рис. 1.2).



Рис. 1.2. Подходы к измерению информации

Алфавитный подход является объективным способом измерения информации, в отличие от субъективного, содержательного подхода.

Рассмотрим подробнее алфавитный подход. В нем количество информации в сообщении зависит от объема текста и мощности алфавита, а не от содержания.



Что такое алфавит? Входят ли в алфавит, кроме букв, знаки препинания, цифры, скобки, пробелы и другие символы, используемые в тексте?



Множество используемых символов в языке называется **алфавитом**. Количество символов в алфавите (его размер) называется **мощностью**.

Наименьшую мощность имеет алфавит, используемый в компьютере (машинный язык). Его называют **двоичным алфавитом**, и он содержит только два знака: **0** и **1**.

При алфавитном подходе считают, что каждый символ текста имеет информационный вес. Информационный вес символа зависит от мощности алфавита. С увеличением мощности алфавита увеличивается **информационный вес** каждого символа. Для измерения объема информации необходимо определить, сколько раз информация, равная 1 бит, содержится в определяемом объеме информации.

Для примера придумаем четырехзначный алфавит (рис. 1.3):

Четырехзначный алфавит				
Символ	◆	⬠	⬡	⬢
Порядковый номер	1	2	3	4
Двоичный код	00	01	10	11

Рис. 1.3. Четырехзначный алфавит

Для кодирования символов алфавита, мощность которого равна четырем, понадобятся два символа двоичного кода. Следовательно, каждый символ четырехзначного алфавита весит 2 бита.

Так как каждый символ несет i бит информации, то количество информации в одном символе можно определить из уравнения: $2^i = N$ (рис. 1.4).

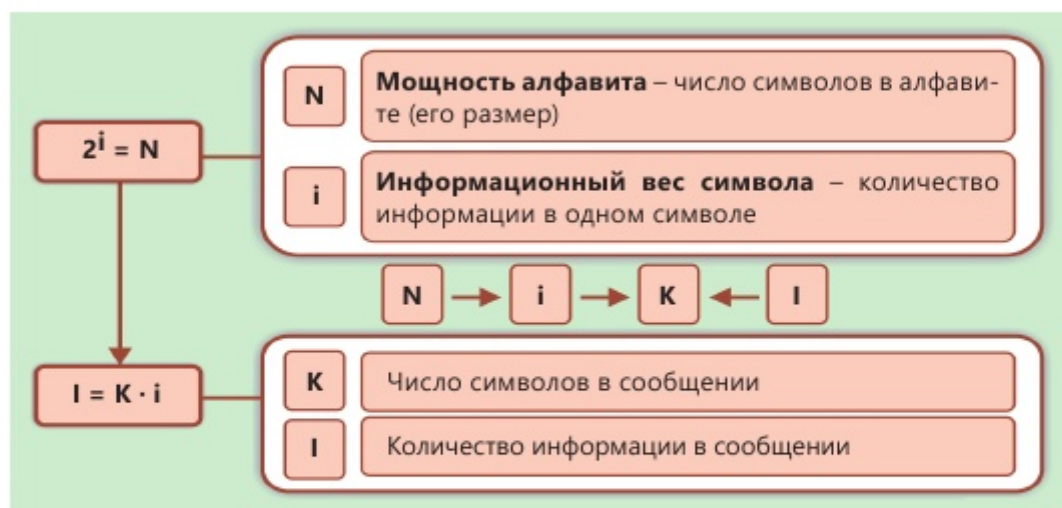


Рис. 1.4. Алфавитный подход к измерению информации



Рассмотрим пример расчета мощности алфавита. Текстовый файл содержит 15 Кбайт информации. Определите мощность алфавита.

Дано:

$I = 15$ Кбайт

Найти: $N - ?$

Решение.

Объем информации будет равен: $15 \cdot 1024 = 15 \cdot 2^{10}$ (байт) = $15 \cdot 2^{10} \cdot 2^3$ (бит).

Тогда на 1 символ приходится $(15 \cdot 2^{10} \cdot 2^3) : (15 \cdot 2^{10}) = 2^3$ (бит) = 8 (бит).

Мощность алфавита будет равна: $N = 2^8 = 256$.

Ответ: алфавит мощностью 256 символов несет в себе 8 бит информации (или 1 байт).

Применение алфавитного подхода удобно при использовании технических средств работы с информацией.



Знание. Понимание



1. Произнесите ключевые слова на трех языках.
2. Какие два подхода к измерению информации вы знаете?
3. Что называется алфавитным подходом?
4. Что означает понятие «мощность алфавита»?
5. Как определяется информационный объем текста при использовании алфавитного подхода?
6. Где удобно применение алфавитного подхода?



Применение

7. Ануару дали задание закодировать с помощью двоичного кода каждый символ алфавита, мощность которого равна 8 (табл. 1.1). Каков информационный объем каждого символа восьмизначного алфавита, закодированного Ануаром?

Таблица 1.1

Восьмизначный алфавит								
Порядковый номер	1	2	3	4	5	6	7	8
Двоичный код	000							

8. Закодируйте с помощью двоичного кода каждый символ алфавита, мощность которого равна 16. Какой вывод можно сделать?

9. Алфавит содержит 32 буквы. Какое количество информации несет одна буква?

Анализ

10. Айдана получила секретное сообщение, записанное буквами 32-символьного алфавита, содержащее 140 символов. Какое количество информации несет это сообщение?

**Синтез**

11. В статье, созданной с помощью текстового процессора Word, содержится 30 страниц, на каждой странице – 40 строк, в каждой строке – 50 символов. Какой объем информации содержит статья?

**Рекомендации для выполнения домашнего задания**

1. Узнайте, сколько на каждой странице символов.
2. Узнайте, сколько символов во всей статье.
3. Вес каждого символа компьютерного алфавита равен 8 бит.
4. Вычислите информационный объем всей статьи.

**Оценивание**

12. Вы получили два письма. Первое письмо состоит из 50 символов 32-символьного алфавита, а второе – из 40 символов 64-символьного алфавита. Сравните объемы информации, содержащиеся в двух письмах.



Эксабайт – единица измерения количества информации, равная 10^{18} байт. Аналитики из Калифорнийского университета утверждают, что человечеству потребовалось 300 тысяч лет, чтобы создать первые 12 эксабайт информации. Вторые 12 эксабайт были созданы всего за два года.

1.2. ПРОЦЕССОР И ЕГО ХАРАКТЕРИСТИКИ**ВЫ НАУЧИТЕСЬ**

Объяснять функции процессора и его основные характеристики.

КЛЮЧЕВЫЕ СЛОВА

Разрядность	Разрядтылығы	Bit depth
Тактовая частота	Тактілік жиілігі	Clock frequency
Регистры памяти	Жад регистрлері	Memory registers



Современные процессоры – это сложнейшие устройства, которые изготавливает человек. Процессор является основной микросхемой компьютера и представляет собой небольшую плоскую полупроводниковую пластину сложной структуры (рис. 1.5).

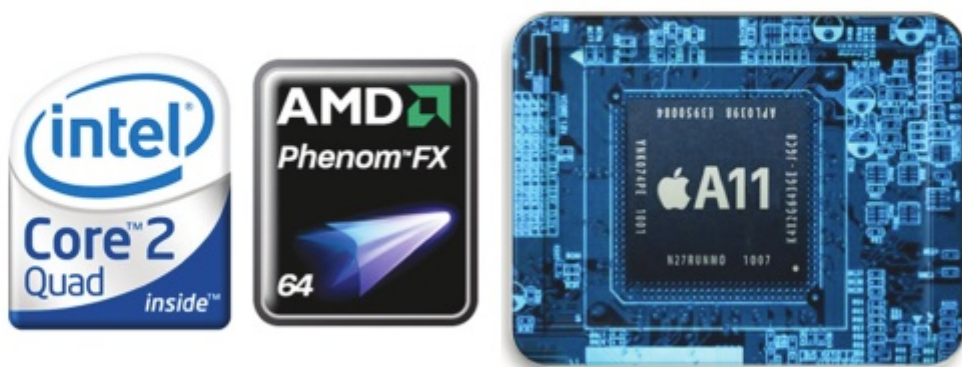


Рис. 1.5. Процессоры Intel, AMD, Apple

Для любого компьютера, ноутбука, моноблока, смартфона, планшета основным источником системы является процессор. Именно процессор отвечает за выполнение задач, и от него зависит большая часть работы устройства. Современные компьютеры, состоящие из нескольких ядер центрального процессора, выполняют разные задачи независимо друг от друга.

В настоящее время лидерами по выпуску процессоров являются фирмы **Intel** (INTEgrated ELEctronics) и **AMD** (Advanced Micro Devices). Известная компания **Apple** ведет разработку собственных графических процессоров, которые используются в мобильных устройствах: смартфонах iPhone и планшетах iPad.

Кроме перечисленных известных компаний, существуют другие, такие как **Cyrix**, **Texas Instruments**, конкурирующие с корпорацией Intel.



Процессор (центральный процессор, *Central Processing Unit*) – это микросхема, обрабатывающая информацию и координирующая работу всех устройств компьютера.

Для краткого обозначения центрального процессора принята аббревиатура **ЦП**. На английском языке принято обозначение **Central Processing Unit (CPU)**, которое переводится как «центральное обрабатывающее устройство».

Рассмотрите рисунок 1.6. Какие модули не подписаны в функциональной схеме компьютера?

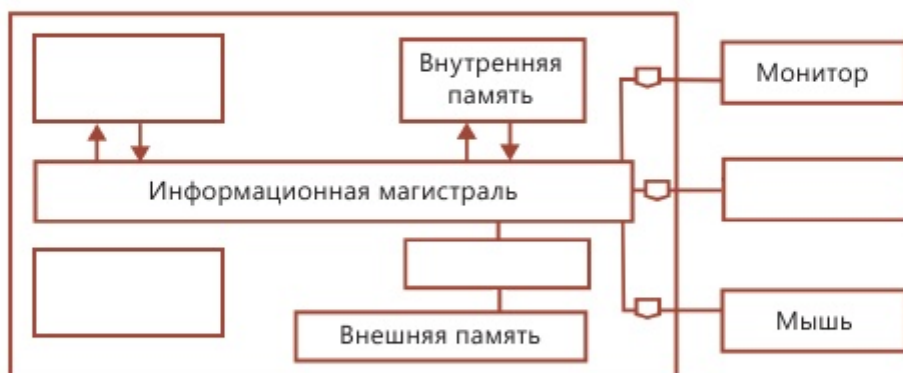


Рис. 1.6. Функциональная схема компьютера

Процессор состоит из десятков миллионов транзисторов. В основе внутренних схем процессора, образующих ядро процессора, лежат (рис. 1.7):

- устройство управления (координирует работу всех устройств компьютера);
- арифметико-логическое устройство (выполняет команды программы, находящиеся в оперативной памяти);
- внутренняя память (регистры памяти – ячейки, в которые по очереди помещаются команды программы для работы процессора);
- шины данных, команд и адресов (по этим магистралям происходит обмен данными между внутренними и внешними устройствами процессора);
- кэш-память (сверхоперативная память).

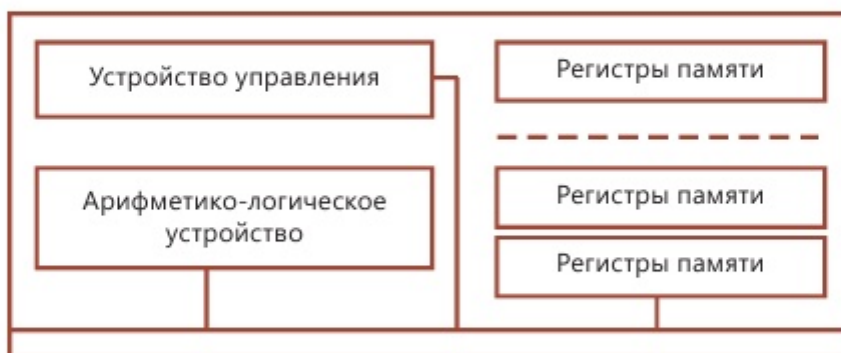


Рис. 1.7. Внутренняя схема процессора

Функционирование процессора происходит следующим образом: вся входящая информация, представленная в виде данных и команд, попадает из оперативной памяти в процессор через внешние шины. Обработка данных происходит в арифметико-логическом устройстве в соответствии с командами. Результат выводится с помощью устройств вывода.

Скорость работы процессора зависит от основных характеристик:

- регистров;
- разрядности;
- тактовой частоты;
- адресного пространства.



Регистр – это ячейка процессора, в которой хранится машинное слово. Машинное слово представляет собой некоторое число (или команду), записанное в двоичном коде.

Существует много разнообразных процессоров, и у каждого свои регистры. Регистры различаются по размеру и по назначению. Бывают регистры 8-, 16-, 32-, 64-разрядные. Это значит, что в регистр помещаются соответственно 8, 16, 32, 64 бит двоичной информации. Размер регистра определяет одну из характеристик процессора – **разрядность**.



Разрядность процессора – это число одновременно обрабатываемых процессором битов. Поэтому процессор может быть 8-, 16-, 32-, 64-разрядным.

От разрядности центрального процессора зависит производительность компьютерной системы. Поэтому чем больше разрядность процессора, тем большее количество информации обрабатывает процессор за единицу времени.

Первые модели процессоров были 16-разрядными и предназначались для персональных компьютеров IBM PC 80286. Следующими моделями процессора стали 32-разрядные. В настоящее время в производстве компьютеров используются 64-разрядные процессоры.

Кроме разрядности процессора важную роль играет **тактовая частота**. Единицей измерения тактовой частоты является **мегагерц (МГц)**. Один мегагерц – это миллион тактов в секунду. Соответственно 1000 мегагерц (1 гигагерц) – это миллиард тактов в секунду.

Тактовой частотой называют количество тактов (операций), выполненных процессором в секунду. Чем выше тактовая частота процессора, тем выше его производительность.



Процессор не только исполняет машинные функции, но и управляет обменом данных между внешней и оперативной памятью.

С какой тактовой частотой работали первые модели процессоров Intel (i8088)? Какова тактовая частота у современных компьютеров?



Процессор управляет не только работой всех устройств компьютера, но и организывает обмен данными между внешней и оперативной памятью. Как происходит этот обмен? Для того чтобы найти в оперативной памяти нужные данные, процессор должен знать их адрес. А адрес к нему передается по **адресной шине**.

Если шина является **n**-разрядной, то по ней можно передать 2^n двоичных чисел. Именно такое количество адресов можно передать по этой шине.

Величина 2^n – это объем адресного пространства процессора, то есть объем оперативной памяти, физически адресуемой процессором.



Зная разрядность шины, можно легко вычислить величину адресного пространства процессора. Например, если адресная шина 24-разрядная, то адресное пространство равно $2^{24} = 16\,777\,216$ байт = 16 Мбайт. То есть процессор имеет адресуемый доступ к 16 Мбайт оперативной памяти.

Знание. Понимание



1. Произнесите ключевые слова на трех языках.
2. Как устроен процессор? Каково назначение процессора?
3. Как процессор управляет работой всех устройств персонального компьютера?
4. Каковы основные характеристики процессора?
5. Каким образом тактовая частота влияет на скорость работы процессора?
6. Как вы понимаете понятие «адресное пространство»?
7. По какой формуле можно вычислить адресное пространство?





Применение



8. Представьте, что вы работаете консультантом в компьютерном магазине. Компьютер с каким процессором вы порекомендовали бы покупателю (табл. 1.2)?

Таблица 1.2

Вид деятельности	Процессор	Причина выбора
Распечатка текста, создание простейших рисунков, табличные расчеты		
Компьютерные игры		
Профессиональная работа с графикой и звуком (видеомонтаж)		

- Процессор Intel Core i7-6700K 4×4.0GHz (Skylake);
- Процессор Intel Core i5-6600K 3.5GHz LGA1151 OEM;
- Процессор Intel Core i3-7100 Kaby Lake 3.9 GHz LGA1151 OEM;
- Процессор Intel Celeron G1840 2.80GHz LGA1150 OEM;
- Процессор AMD FX-6300 BE Vishera 4.1;
- Процессор Intel Pentium MMX 233МГц;
- Процессор Intel Pentium II 400Гц;
- Процессор Intel Celeron 800МГц;
- Процессор Intel Pentium IV 3.5 МГц;
- Процессор Intel Pentium IV 2ГГц.

9. Найдите в прайс-листе (по возможности используйте поисковые системы Интернета) характеристики процессоров, указанных в задании 8, и расшифруйте их.



Анализ



10. **Проблемная ситуация:** «Как ускорить работу процессора?»

Допустим, вы – рабочая группа и должны работать в помещении 1. Исходные данные и условие задачи находятся в помещении 2.

Выдача необходимой информации происходит очень медленно. В помещении 2 за порцией данных может ходить только один человек. Он приносит данные в помещение 1.

Рабочая группа обрабатывает данные и посылает за следующей порцией информации. Во время ожидания группа бездействует.

- 1) За счет чего можно ускорить этот процесс? Можно ли посылать за информацией в помещение 2 не одного человека, а несколько?
- 2) Смоделируйте процесс ускорения работы процессора.

Для этого обозначьте: помещение 1 – процессор, помещение 2 – оперативная память.

Синтез

Дополните таблицу 1.3 «Поколения процессоров» с 1978–2019 гг. (По возможности используйте данные из Интернета или других источников.)

**Таблица 1.3****Поколения процессоров**

Год выпуска	Тип	Частота (МГц)	Шина данных	Адресное пространство
1978	8086	4–12	16	1 Мбайт
1982	80 286	8–20	16	16 Мбайт
1985				
1989	80 486	33–50	32	4 Гбайт
1993	Pentium	60–300	64	4 Гбайт
1997				
1999	Pentium III	450–1000	64	4 Гбайт
2000	Pentium 4	1000–2000	64	4 Гбайт
2008				
2011				
2014				
2015				
2017				
2018				
2019				

Оценивание

12. Постройте генеалогическое древо семейства персональных компьютеров в зависимости от характеристик процессора (используйте таблицу 1.3).

**Рекомендации для выполнения домашнего задания**

Генеалогическое древо – схематичное представление родственных связей в виде условно-символического «дерева». У «корней» указывается родоначальник. На «стволе» – представители основной (по старшинству) линии рода, на «ветвях» – различные линии родословия, известные его потомки – «листья».



2–6001



Современные процессоры имеют возможность многоканальной работы с оперативной памятью, благодаря которой повышается их функциональный уровень.

У каждого процессора своя система команд, поэтому один и тот же код для разных процессоров может обозначать разные команды. Примером семейства процессоров являются все процессоры Intel. Их родоначальником является процессор Intel 8086.

Все процессоры можно разделить на:

- 1) процессоры с расширенной системой команд (CISC-процессоры);
- 2) процессоры с сокращенной системой команд (RISC-процессоры).

Первый тип процессоров характеризуется небольшим числом регистров, большим числом команд и медленной работой. Они используются в универсальных вычислительных системах и являются стандартом для микрокомпьютеров.

Второй тип процессоров характеризуется меньшим количеством команд и высоким быстродействием. Используются в специализированных вычислительных системах, ориентированных на выполнение однообразных операций.

Компьютерные технологии развиваются с большой скоростью. Что будет с компьютерами в будущем? Допустим, если производство микропроцессоров будет развиваться по закону Мура, то вычислительная мощность компьютеров должна удваиваться каждые два года. Тогда через 100 лет компьютеры будут в 1 125 899 906 842 624 раза мощнее, чем сегодня.

1.3. КОМПЬЮТЕРНЫЕ СЕТИ



ВЫ НАУЧИТЕСЬ

Определять пропускную способность сети.

КЛЮЧЕВЫЕ СЛОВА

Пропускная способность сети	Желінің өткізу қабілеті	Network bandwidth
Отправитель информации	Ақпарат жіберуші	Sender of information
Получатель информации	Ақпарат қабылдаушы	Recipient of information
Канал связи	Байланыс арнасы	Communication channel
Пропускная способность	Өткізу қабілеті	Bandwidth
Скорость передачи	Тарату жылдамдығы	Transmission speed
Трафик	Трафик	Traffic

Создание глобальной компьютерной сети Интернет привело к формированию единого мирового информационного пространства. Для процесса обмена информацией между компьютерами в Интернете существует единая система адресации. Каждый компьютер имеет свой уникальный **IP-адрес** (англ. *Internet Protocol Address*). С помощью адреса (2^{32} -битного) различные устройства могут обмениваться информацией с другими объектами Интернета.

Общая схема передачи информации включает в себя три объекта (рис. 1.8):

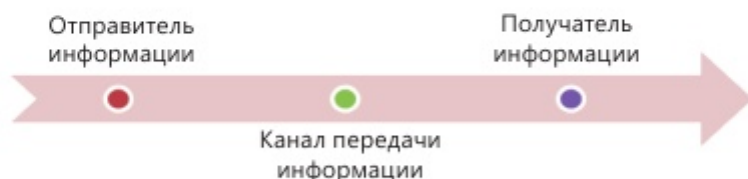


Рис. 1.8. Общая схема передачи информации

По **каналам передачи информации** происходит обмен данными. Каналы передачи информации могут использовать различные физические принципы. Например, при общении людей информация передается с помощью звуковых волн. При разговоре по телефону происходит распространение сигналов по линиям связи.

Канал связи (англ. *channel, data line*) – система технических средств и среда распространения сигналов для односторонней передачи данных (информации) от отправителя (источника) к получателю (приёмнику).



При использовании проводной линии связи средой распространения сигнала может являться оптическое волокно или витая пара.

Информация в сети передается с определенной скоростью. Что такое скорость передачи информации?

Скорость передачи информации – количество информации, передаваемое за единицу времени.



$$V = q \cdot t$$

где **V** – объем переданной информации, **q** – пропускная способность канала, **t** – время передачи информации.

Основной характеристикой каналов передачи информации является их **пропускная способность**.

Пропускная способность измеряется в битах в секунду (**бит/с**) и кратных единицах – **Кбит/с** и **Мбит/с**, а когда в качестве единицы измерения используется байт в секунду (**байт/с**), то – в кратных ему единицах – **Кбайт/с** и **Мбайт/с**.

Соотношения между единицами пропускной способности канала передачи информации такие же, как и между единицами измерения количества информации:

$$\begin{aligned} 1 \text{ байт/с} &= 2^3 \text{ бит/с} = 8 \text{ бит/с} \\ 1 \text{ Кбит/с} &= 2^{10} \text{ бит/с} = 1024 \text{ бит/с} \\ 1 \text{ Мбит/с} &= 2^{10} \text{ Кбит/с} = 1024 \text{ Кбит/с} \\ 1 \text{ Гбит/с} &= 2^{10} \text{ Мбит/с} = 1024 \text{ Мбит/с} \end{aligned}$$

Если типичная сеть спроектирована и реализована с полосой пропускания 100 Мбит/с, то это не означает, что каждый пользователь сможет за одну секунду передать данные объемом 100 Мбит/с. Такое возможно только в идеальной ситуации.

Под пропускной способностью понимается реальная полоса пропускания, измеренная в определенное время. Пропускная способность сети часто оказывается значительно меньше, чем максимально возможная цифровая полоса пропускания.

Ниже приведены основные факторы, определяющие реальную полосу пропускания, или пропускную способность сети (рис. 1.9).



Рис. 1.9. Факторы пропускной способности сети

Основным компонентом для измерения потока данных в Интернете, контроля компьютерных сетей и моделирования сетевых соединений является **трафик**.

Интернет-трафик (англ. *traffic* – движение) – это объем информации, передаваемой через компьютерную сеть за определенный период времени.



Трафик бывает входящим и исходящим.

Входящий трафик – это поток информации, получаемой из сети Интернет.

Исходящий трафик – это отправляемые данные (текстовая, графическая, видео и т.д.) с использованием сети Интернет.



Трафик может увеличиваться, если на компьютере появились вирусы. Объем интернет-трафика в мире распределен следующим образом (рис. 1.10):



Рис. 1.10. Распределение интернет-трафика в мире

Правильная организация трафика помогает в обеспечении качества обслуживания в данной сети и снижении затрат на ее поддержание.

Рассмотрим решение задач.

Шаг за шагом

Задача 1. Ануару нужно срочно отправить информацию. Какой объем информации в мегабайтах будет передан за 1 час при пропускной способности канала в 1 Мбит/с?

Дано:

$$q = 1 \text{ Мбит/с}$$

$$t = 1 \text{ час}$$

Найти: V (объем информации) = ? (Мбайт)

Решение.

1. Переведем 1 Мбит в биты:

$$1 \text{ Килобит} = 1024 \text{ бит.}$$

$$1 \text{ Мбит} = 1024 \cdot 1024 = 1\,048\,576 \text{ бит.}$$

2. Переведем биты в байты, Кбайты, Мбайты:

$$1\,048\,576 \text{ бит} / 8 = 131\,072 \text{ байт.}$$

$$131\,072 \text{ байт} / 1024 = 128 \text{ Кбайт.}$$

$$128 \text{ Кбайт} / 1024 = 0,125 \text{ Мбайт.}$$

$$1 \text{ Мбит} = 0,125 \text{ Мбайт.}$$

$$1 \text{ Мбит/с} = 0,125 \text{ Мбайт/с.}$$

3. Переведем 1 час в секунды:

$$1 \text{ час} = 60 \cdot 60 = 3600 \text{ с.}$$

4. Вычислим объем переданной информации за 1 час:

$$V = 0,125 \text{ Мбайт/с} \cdot 3600 \text{ с} = 450 \text{ Мбайт.}$$

Ответ: объем переданной информации равен 450 Мбайт.

Задача 2. Амиру нужно передать 500 страниц текста в 30 строк по 60 символов. Каждый символ закодирован одним байтом. Сколько минут потребуется ждать Амиру, если модем передает сообщения со скоростью 28 800 бит/с?

Дано:

$$q \text{ (пропускная способность)} = 28\,800 \text{ бит/с}$$

Весь текст – 500 страниц

На 1 странице – 30 строк

В строке – 60 символов

1 символ – 1 байт

Найти: t (время) – ? (мин)

Решение.

1. Вычислим объем файла в битах:

$$V = 500 \cdot 30 \cdot 60 = 900\,000 \cdot 8 = 7\,200\,000 \text{ бит.}$$

2. Вычислим время, которое потребуется модему для передачи сообщения:

$$t = V / q = 7\,200\,000 \text{ бит} / 28\,800 \text{ бит/с} = 250 \text{ с} \approx 4 \text{ мин}$$

Ответ: время, которое потребуется модему, составляет 250 секунд или приблизительно 4 минуты.

Задача 3.

Даурену дали задание посчитать пропускную способность канала, если за 30 секунд по каналу связи было передано 1500 байт информации.

Дано:

$$V = 1500 \text{ байт}$$

$$t = 30 \text{ с}$$

Найти: q – ? (байт)

Решение.

$$q = V / t = 1500 / 30 = 50 \text{ байт/с.}$$

Ответ: пропускная способность канала равна 50 байт/с.

Знание. Понимание



1. Какова общая схема передачи информации?
2. С помощью чего осуществляется передача данных на расстояние?
3. С помощью каких каналов связи происходит обмен информацией в компьютерной сети?
4. Что вы понимаете под пропускной способностью канала?
5. По какой формуле вычисляется объем передаваемой информации?
6. Каким образом можно измерить пропускную способность сети?
7. Назовите единицы измерения пропускной способности.
8. Перечислите основные факторы пропускной способности.



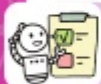


Применение



9. Скорость передачи данных через ADSL-соединение (вид модема) равна 128 000 бит/с. Через данное соединение передают файл размером 625 Кбайт. Определите время передачи файла в секундах.

10. Скорость передачи данных через ADSL-соединение равна 1 024 000 бит/с. Передача файла через данное соединение заняла 5 секунд. Определите размер файла в килобайтах.



Анализ



11. Текст подготовлен для передачи по сети и содержит 512 000 символов. Каждый символ кодируется 2 байтами и во избежание искажений передается трижды. Время передачи текста составило 64 секунды. Какова скорость передачи в байтах в секунду?



Синтез



12. Скорость передачи данных через модемное соединение равна 512 000 бит/с. Передача текстового файла через это соединение заняла 10 секунд. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 8-битной кодировке Unicode.



Оценивание



13. Нужно передать цветное растровое изображение размером 640×480 пикселей. Цвет каждого пикселя кодируется 3 байтами. Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с?

Рекомендации для выполнения задания

1. Подсчитайте информационный объем информации, применив формулу: $I = K \cdot i$.

2. Из формулы вычисления пропускной способности канала $q = V/t$ найдите время (t), необходимое модему для передачи сообщения.



Искусственный интеллект для организации сетей

Искусственный интеллект (ИИ) – программное обеспечение, которое выполняет задачи наравне с человеком-экспертом. Искусственный интеллект играет важную роль в управлении сетей. Без правильной стратегии в сфере

искусственного интеллекта информация не может соответствовать современным жестким сетевым требованиям. Опишем несколько технологических элементов, которые должна включать стратегия в сфере искусственного интеллекта.

1. **Данные:** любое значимое решение на основе ИИ начинается с огромного количества качественных данных. Со временем ИИ постоянно наращивает свой интеллект посредством сбора и анализа данных. Чем разнообразнее собираемые данные, тем умнее решение на основе ИИ. Например, для устройств Интернета вещей и мобильных устройств важно собирать данные с каждого устройства в режиме реального времени, а затем быстро обрабатывать их, используя алгоритмы ИИ.

2. **Знания, специфические для конкретной области:** помощь врачам в диагностировании рака или поддержка ИТ-администраторов при диагностике проблем беспроводных сетей. Для выполнения самых разных задач решениям на основе ИИ нужны маркированные данные.

3. **Инструменты для анализа и обработки данных** для использования в машинном обучении и в качестве больших данных. Для анализа данных и предоставления практических рекомендаций используют различные методы, такие как машинное обучение и нейронные сети.

4. **Виртуальный сетевой помощник.** Коллаборативная фильтрация – это метод машинного обучения, с которым сталкиваются многие, когда выбирают фильм в сервисе Netflix или покупают что-либо на Amazon и получают рекомендации похожих фильмов или товаров. Помимо рекомендаций, коллаборативная фильтрация может применяться для сортировки больших массивов данных, а также для выявления и корреляции тех из них, которые формируют решение конкретной задачи на основе ИИ.

В ИИ для организации сетей виртуальный сетевой помощник может функционировать в беспроводной среде как виртуальный беспроводной эксперт, помогающий решать сложные задачи. Виртуальный сетевой помощник сочетает в себе качественные данные, экспертные знания в предметных областях. Он дает прогнозные рекомендации о том, как избежать проблемы, и предлагает практические рекомендации по их устранению. Помощник может изучить нюансы беспроводной сети и ответить на такие вопросы, как «Что пошло не так?» и «Почему это случилось?». Именно такие автоматизированные улучшения обеспечивает ИИ.

В перспективе сети будущего смогут выполнять следующие задачи.

1. Сети на базе ИИ и машинного обучения смогут узнавать о намерениях поведения сети, обеспечивать аналитику, а также предоставлять рекомендации и информацию о внесении исправлений.

2. Автоматическое размещение и перемещение сервисов.

3. Использование передовых технологий проверки для активного мониторинга качества обслуживания и изменения потоков трафика на основе требований, предъявляемых к сервису.

4. Обеспечение конкретных обновлений на базе настроенных сервисов.

5. Автономная работа благодаря активному мониторингу и отчетности, которые обеспечивает сетевой оператор.

Проект 1

Название проекта: «Процессор и его характеристики».

Цель: развитие творческого мышления при систематизации знаний о процессоре и его характеристиках.

Задачи:

- систематизируйте информацию о работе процессора и его характеристиках;
- выявите особенности работы процессора и его характеристики;
- проведите презентацию проекта.

Этапы проведения работы

№	Содержание работы	Что делать?
1	Выявите тему и цель проекта	Обсудите тему с учителем и дополните
2	Найдите источники информации и обсудите	Закрепите план работы
3	Проанализируйте информацию и сделайте заключение	Проведите анализ указанной работы
4	Презентуйте результаты	Презентация выполненной работы в группе
5	Рефлексия	Оценивание

Рекомендации к выполнению проекта: в виде «генеалогического древа» начертите схему и характеристики процессора.

Требования к выполнению презентации

1. Создайте «генеалогическое древо» в программе Microsoft PowerPoint.
2. В корне «генеалогического древа» укажите «основу».
3. В стволе – «объем памяти».
4. В стеблях – «принципы работы».
5. В листьях – вид процессора.
6. Вставьте соответствующие графические изображения, которые можно фотографировать, нарисовать или применить готовые.
7. Оформите презентацию. Проявите творчество.

Дополнительные задания к первому разделу

Задания для проектной работы

1. Книга, набранная с помощью компьютера, содержит 150 страниц. На каждой странице – 40 строк, в каждой строке – 60 символов. Каков объем информации в книге в килобайтах и мегабайтах?
2. Сообщение, записанное буквами из 64-символьного алфавита, содержит 20 символов. Какой объем информации оно несет?
3. Сколько символов содержит сообщение, записанное с помощью 16-символьного алфавита, если его объем составил 1/16 часть мегабайта?
4. Объем сообщения, содержащего 2048 символов, составил 1/512 часть мегабайта. Каков размер алфавита, с помощью которого записано сообщение?

Тестовые задания к разделу I

1. Укажите наиболее полный перечень основных устройств персонального компьютера:

- A) микропроцессор, сопроцессор, монитор;
- B) арифметико-логическое устройство, устройство управления, регистр памяти;
- C) процессор, оперативная память, устройства ввода-вывода;
- D) процессор, оперативная память.

2. Тактовая частота процессора – это:

- A) число двоичных операций, совершаемых процессором в единицу времени;
- B) количество тактов, вырабатываемых за одну секунду;
- C) число возможных обращений процессора к оперативной памяти в единицу времени;
- D) скорость обмена информацией между процессором и устройствами ввода-вывода;
- E) обмен информацией между процессором и устройствами ввода-вывода.

3. Информационный объем информации при алфавитном подходе зависит от:

- A) суммы информационных весов составляющих его символов с учетом вероятности появления символа алфавита;
- B) суммы символов, составляющих текст;
- C) суммы информационных весов составляющих его символов;

- D) информационного объема;
- E) содержательного подхода к измерению информации.

4. Информационный вес одного символа компьютерного алфавита, мощность которого равна 256, составляет:

- A) 8 бит.
- B) 16 байт.
- C) 8 байт.
- D) 2 байт.
- E) 16 бит.

5. Что характерно для алфавитного подхода к измерению информации?

- A) Количество информации зависит от восприятия текста человеком.
- B) Количество информации зависит от информационного веса.
- C) Количество информации не зависит от восприятия текста человеком.
- D) Количество информации зависит от смыслового содержания текста.
- E) Количество информации зависит от позиции человека.

6. Нурдаулет отправил своему однокласснику SMS-сообщение объемом 20 480 бит. Позвонив однокласснику, Нурдаулет узнал, что свободного места для приема информации осталось только 3,125 Кбайт. Прочтет ли сообщение одноклассник?

- A) Да, так как 20 480 бит = 2,5 Кбайт, а свободного места – 3,125 Кбайт.
- B) Да, так как 2560 байт = 2,5 Кбайт, а свободного места – 256 000 байт.
- C) Нет, так как 20 480 бит = 2,5 Кбайт, а свободного места – 3200 байт.
- D) Нет, так как 20 480 бит = 2,5 Кбайт, а свободного места – 32 000 байт.
- E) Нет, так как 20 480 бит = 3,5 Кбайт, а свободного места – 3,125 байт.

7. Сообщение объемом 2,5 Мбит проходит за 1/3 минуты. Рассчитайте пропускную способность канала (в Кбит/с).

- A) 120 Кбит/с.
- B) 128 Кбит/с.
- C) 8 Кбит/с.
- D) 256 Кбит/с.

8. Рассчитайте время, за которое будет передано 900 Кбайт информации, если пропускная способность канала связи равна 512 байт/с.

- A) 10 мин.

- B) 100 с.
- C) 0,5 ч.
- D) 1/2.

9. Считая, что каждый символ кодируется одним байтом, оцените информационный объем стихотворения Абая Кунанбаева «Будь разборчив на пути своём...»: «Речи тех, кто учит детей, как зерно в земле прорастут». (Кавычки не учитывать.)

- A) 54 байт.
- B) 352 бит.
- C) 1024 байт.
- D) 8 бит.
- E) 1 байт.

10. Объем сообщения равен 11 Кбайт. Сообщение содержит 11 264 символа. Какова мощность алфавита?

- A) 33 символа.
- B) 24 символа.
- C) 256 символов.
- D) 11 символов.
- E) 16 символов.



2.1. НЕГАТИВНЫЕ АСПЕКТЫ ИСПОЛЬЗОВАНИЯ КОМПЬЮТЕРА



ВЫ НАУЧИТЕСЬ

Приводить примеры влияния различных электронных устройств на организм человека и эффективно использовать методы защиты.

КЛЮЧЕВЫЕ СЛОВА

Электронные устройства	Электрондық құрылғылар	Electronic devices
Электромагнитное излучение	Электрмагниттік сәулелену	Electromagnetic radiation



Как вы считаете, электромагнитное излучение есть только в электронных устройствах? Приведите примеры.

В настоящее время электромагнитное излучение преследует нас везде. Источниками электромагнитного излучения являются электрические сети, вся бытовая техника, транспорт (троллейбусы, трамваи), уличное освещение. К ним также относятся телевизоры, мобильные телефоны, гаджеты и другие электрические приборы (рис. 2.1). То есть любое электронное устройство, которое производит или потребляет электроэнергию, создает электромагнитное излучение.



Рис. 2.1. Электрические приборы

Электромагнитное излучение – это электромагнитные волны, которые возбуждаются разными излучающими объектами.



Компьютерная техника (ноутбук, планшет, смартфон и т.д.) является неотъемлемым элементом практически каждого дома.

В наше время компьютеризация в Казахстане принимает широкий размах. Сотни тысяч людей проводят большую часть свободного времени и рабочего дня перед монитором компьютера. Несмотря на пользу применения компьютерной техники, она может негативно воздействовать на здоровье человека.

Человек, находясь в непосредственной близости от компьютера, подвергается электромагнитному излучению. Это излучение в 500 раз превышает допустимое значение, если учитывать, что предельно допустимая доза электромагнитного излучения для человека – 0,2 мкТл (микротесла).

Воздействие компьютера на организм человека достаточно серьезное. Это связано с тем, что:

- 1) монитор и системный блок являются источниками электромагнитного излучения;
- 2) часто устройства размещаются на небезопасном расстоянии;
- 3) длительность работы за компьютером не регулируется;
- 4) большое количество компьютеров концентрируется в одном месте.

Электромагнитное излучение невидимо, поэтому многим людям его вредное воздействие кажется нереальным. К сожалению, негативное влияние на детский организм увеличивается в десятки раз, в отличие от взрослого.

На рисунке 2.2 представлены основные проблемы, связанные с применением электронных устройств и их негативным влиянием на детский организм.

Необходимо помнить, что компьютерные игры являются самыми опасными для зрения. Это связано с тем, что мелкие детали и постоянно сменяющееся динамичное изображение очень сильно утомляют глаза.

Каким образом компьютер может влиять на опорно-двигательный аппарат? Длительное пребывание за электронным устройством в однообразной, стесненной позе приводит к боли в кистях рук, плечах, онемению шеи и покалыванию в ногах, возникает искривление позвоночника и развивается мышечная слабость.

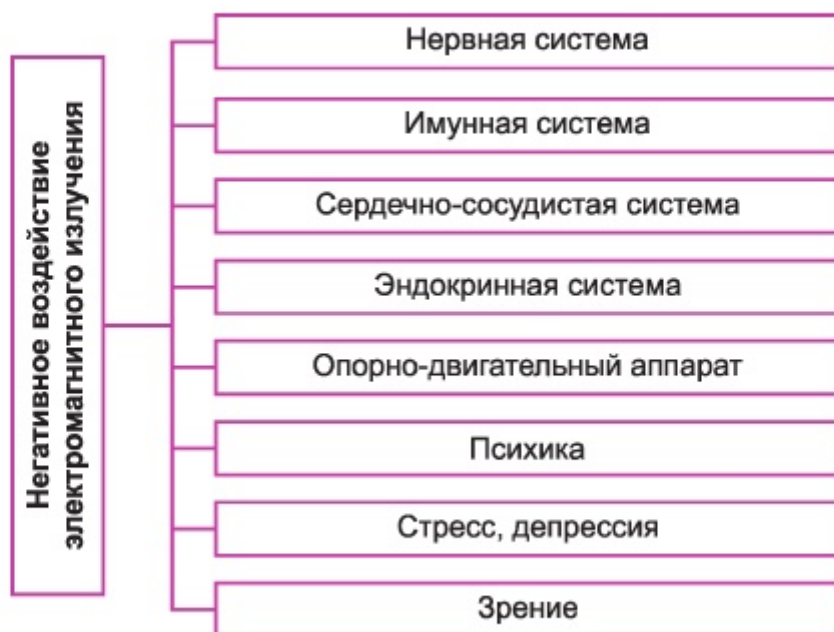


Рис. 2.2. Негативное воздействие электронных устройств на человека

Для минимизации вредного воздействия на организм человека от электронных устройств необходимо выполнять требования, указанные на рисунке 2.3.

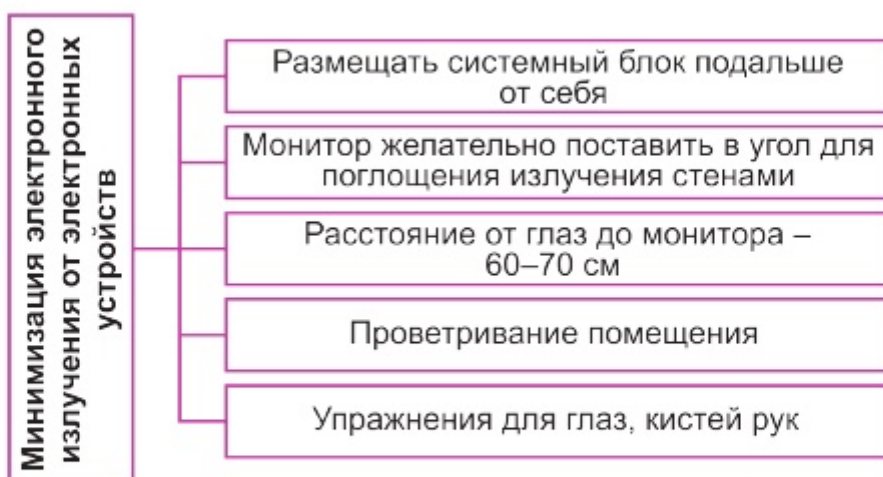


Рис. 2.3. Требования к минимизации электронного излучения

Какие правила техники безопасности при работе за компьютером и защиты от негативного влияния электронных устройств вы знаете?



Как известно, уровень электромагнитного фона Земли превышает естественный уровень в 200 000 раз. Это становится реальной угрозой всему живому. Электромагнитное загрязнение окружающей среды происходит из-за автомобильных, портативных и ручных радиостанций, компьютерной и бытовой электротехники и т.д. Чтобы избежать или минимизировать электромагнитное излучение при работе с электронными устройствами, необходимо соблюдать технику безопасности. Приведем некоторые способы защиты от негативного влияния электронных устройств (табл. 2.1).

Таблица 2.1

Электронные устройства	Защита человека от негативного воздействия
Кондиционеры, принтеры, проекторы, неоновые лампы, электропроводка, источники бесперебойного питания и т.д.	Если расстояние менее 1,5 метра, зона считается опасной. Постарайтесь находиться от них на расстоянии более чем 1,5 метра
При работе за компьютером	Самым уязвимым является место справа, под углом или сзади компьютера
Несколько компьютеров в компьютерном классе повышают общий электромагнитный фон помещения	Чтобы уменьшить источник электромагнитного излучения, необходимо: <ul style="list-style-type: none"> – делать на перемене влажную уборку в кабинете; – размещать компьютеры по периметру кабинета, оставляя центр помещения свободным; – выключать компьютер по окончании работы; – запрещается очищать от пыли и загрязнения электрооборудование, находящееся под напряжением
Процессор генерирует СВЧ-излучение	Молодой организм более подвержен влиянию сверхчастотного излучения. Поэтому необходимо придерживаться определенного времени работы за компьютером. Систематически выполнять упражнения для снятия напряжения с глаз и мышц тела

Электронные устройства	Защита человека от негативного воздействия
Мобильные устройства	Эксперты рекомендуют конкретные меры предосторожности: – не прижимать мобильный телефон к голове, а использовать гарнитуру (наушники) или громкую связь; – передавать большие объемы данных или смотреть видео при хорошей беспроводной связи устройства; – выключать точки доступа, когда они не используются, так как устройство всё равно посылает сигналы

Для беспроводного подключения к Интернету широко внедряется Wi-Fi-система. Опасны ли для здоровья человека сети Wi-Fi? Исследования показывают, что сети Wi-Fi действуют на той же частоте, что и сверхвысокочастотная печь (СВЧ-печь). Для человека такая частота вредна. Поэтому точку доступа к Wi-Fi нужно размещать не ближе, чем на 1 метр от места, где человек находится длительное время (кровать, стол, диван, место для игр и т.д.).



Какие методы защиты от вредного воздействия электромагнитного излучения при работе за компьютером вы еще можете предложить?



Знание. Понимание



1. Расскажите о вредных воздействиях электромагнитного излучения при работе с электронными устройствами.
2. Все ли электронные устройства дают электромагнитное излучение?
3. Какие основные проблемы, связанные с электромагнитным излучением, могут возникнуть у человека?
4. Какие возможности существуют для минимизации электромагнитного излучения?
5. Перечислите причины воздействия компьютера на организм человека.

Применение



6. Примените изученный материал в конкретных условиях, соедините теорию и практику. Создайте презентацию.

Для теоретиков: Изучите влияние негативных факторов на здоровье человека. На слайдах информацию можно представить в любой форме: в виде таблиц, рисунков, видео и т.д. В отчете представить как минимум шесть негативных факторов, влияющих на здоровье человека.

Для практиков: Изучите способы профилактики заболеваний, вызываемых вредными факторами. На слайдах информацию можно представлять в любой форме: в виде таблиц, рисунков, видео и т.д. В отчете представить как минимум шесть заболеваний и их профилактику.



Анализ



7. Проведите социологический опрос по вопросам из таблицы 2.2. По итогам опроса составьте диаграмму в электронной таблице Excel. Проанализируйте данные и составьте краткий отчет.



Таблица 2.2

№	Вопросы	Ответы		
		Да	Нет	Иногда
1	Сможет ли человек обойтись без компьютера?			
2	Дает ли компьютер электромагнитное излучение?			
3	Может ли компьютер влиять на ухудшение зрения, памяти, психики и т.д.?			
4	Можно ли за компьютером работать без перерыва?			
5	Возможно ли оградить себя от электромагнитного излучения при работе за компьютером?			
6	Делаете ли вы физкультминутки или динамические паузы во время работы за компьютером?			
7	Соблюдаете ли вы технику безопасности при работе с электронными устройствами?			



Синтез



8. Напишите эссе «Компьютер – друг мой, враг мой». В эссе должна быть новизна.



Рекомендации для выполнения домашнего задания

Эссе – литературный жанр, прозаическое сочинение небольшого объема и свободной композиции.



Оценивание



9. Систематизируйте изученный материал. Составьте рекомендации для эффективной защиты здоровья от электромагнитного излучения электронных устройств. Для представления информации примените инфографику.

Инфографика – это способ подачи информации, данных и знаний с помощью графического изображения.

Можно воспользоваться онлайн-сервисами для создания инфографики или нарисовать на бумаге.

Онлайн-сервисы для иконографии:

- Piktochart.com;
- Easel.ly;
- Creately.com;
- Infogr.am.



Дополнительные устройства, подключенные к компьютеру, также несут опасность для здоровья. Чтобы защитить себя от их вреда, соблюдайте меры безопасности, приведенные ниже.

Наушники, беспроводные гарнитуры представляют опасность, потому что всегда надеваются на голову.

Колонки создают вокруг себя большое электромагнитное поле. От них нужно находиться на расстоянии не менее 50 сантиметров.

Простой домашний **принтер** держите на расстоянии 50 сантиметров.

Магнитные поля **роутеров, модемов, маршрутизаторов** простираются вокруг на много метров. Располагайте их на расстоянии не менее 35 сантиметров от себя.

2.2. БЕЗОПАСНОСТЬ В СЕТИ

ВЫ НАУЧИТЕСЬ

Соблюдать правила обеспечения безопасности пользователя в сети (мошенничество и агрессия в Интернете).

КЛЮЧЕВЫЕ СЛОВА

Безопасность в сети	Желідері қауіпсіздік	Security in the network
Фишинг	Фишинг	Phishing
Спам	Спам	Spam
Кибербуллинг	Кибербуллинг	Cyberbullying
Троллинг	Троллинг	Trolling
Аккаунт	Аккаунт	Account
Информационная безопасность	Ақпараттық қауіпсіздік	Information security
Брандмауэр	Брандмауэр	Firewall
Ботнеты DoS	DoS ботнеттері	Dos Botnets
DDoS-атака	DDos шабуылы	Distributed Denial of Service



Какие достоинства и недостатки Интернета вы можете назвать?



Развитие средств информационных коммуникаций привело к возникновению угроз **информационной безопасности**.

Информационная безопасность – это сохранение и защита информации, в том числе системы и оборудования, которые предназначены для применения, сохранения и передачи этой информации.



К угрозам информационной и компьютерной безопасности относятся события, процессы или действия, которые могут нанести ущерб информационным и компьютерным системам.

На рисунке 2.4 показано, на какие основные подгруппы можно разделить угрозы информационной безопасности.

Вредоносные программы, ботнеты DoS и DDoS-атаки являются основными техническими угрозами. Они могут причинить ущерб компьютеру, серверу или компьютерной сети.

Часто вредоносные программы распространяются через файлы или ссылки, присланные по электронной почте, при скачивании файлов из Интернета, через съемные носители с информацией.



Рис. 2.4. Угрозы информационной безопасности

Как скрываются вредоносные программы? Они могут быть спрятаны в виде звуковых или графических сообщений, ссылок, которые могут появиться во всплывающих окнах, в рекламных объявлениях различных сайтов. К вредоносным программам относятся **вирусы, черви, троянские программы**.

Ботнет (англ. *botnet*) – это компьютерная сеть, состоящая из некоторого количества хостов с ботами. **Бот** – это автономное программное обеспечение, которое может состоять из вирусов, брандмауэров, программ для удаленного управления компьютером, а также инструментов для скрытия от операционной системы.

Брандмауэр – это программно-аппаратный комплекс, контролирующей сетевые пакеты или установленные программы. Применяется для защиты системы от компьютерных атак.

DoS-атакой называется атака на веб-сервер или другую компьютерную систему для вывода ее из строя. При этом создаются условия, затрудняющие доступ пользователей к сайту.

Одновременную атаку с большого количества компьютеров называют **DDoS** (*Distributed Denial of Service*), т.е. распределенной атакой на отказ в обслуживании.

Чтобы избежать опасности, которые подстерегают пользователей в сети Интернет, необходимо не только знать специальную терминологию различных видов угроз (рис. 2.5), но и избегать **нежелательный контент** (изображения, видео, аудио, тексты), запрещенный законодательством.



Рис. 2.5. Специальные термины угроз

Подростки легко интегрируются в мир интернет-технологий. Но они уязвимы к виртуальной агрессии сверстников, чаще подвергаются краже личных данных, взлому аккаунтов, агрессии и другим видам мошенничества.

Аккаунт – учетная запись пользователя, то есть набор личных данных хранящихся на каком-либо сайте или интернет-сервисе. Учетная запись является **конфиденциальной информацией** (секретной) и не подлежит огласке. Аккаунт необходим для общения в социальных сетях, при обучении с помощью дистанционных образовательных технологий, в сетевых компьютерных играх и т.д.



1. Не сообщайте пароли от своих персональных страниц и электронной почты.
2. Придумывайте сложные пароли, включающие комбинации букв и цифр. Не используйте в качестве пароля дату и год рождения.
3. Придумывайте разные логины и пароли для своей электронной почты, аккаунтов в социальных сетях и регистрации на различных сайтах. Один и тот же пароль позволит мошенникам взломать и другие ваши персональные страницы.
4. Закройте свой аккаунт, если использовали чужой компьютер для входа в него.
5. При взломе аккаунта смените пароль.

Чтобы свести опасность попадания в неприятные ситуации к минимуму, каждый пользователь должен знать и соблюдать правила безопасности в Интернете. На рисунке 2.6 приведены некоторые правила поведения во Всемирной информационной сети.

1

Не указывайте свою личную информацию: адрес, телефон, место жительства, номер школы и т.д.

2

Осторожно публикуйте личные фотографии. Не выкладывайте фотографии чужих людей без их согласия

3

Не вся информация в Интернете достоверная, поэтому не доверяйте незнакомым людям

4

Не переходите по сомнительным ссылкам, например, обещающим выигрыши. Не посещайте сомнительные сайты

- 5 Помните об авторских правах и незаконном использовании чужой интеллектуальной собственности
- 6 Хорошо подумайте, прежде чем встречаться в реальной жизни с людьми, с которыми познакомились в Интернете
- 7 Помните, что в виртуальном мире действуют те же правила этикета, что и в реальной жизни
- 8 Не отправляйте СМС, чтобы получить какую-либо услугу или выиграть приз
- 9 Обращайтесь за советом к взрослым при малейшем подозрении или сомнении

Рис. 2.6. Правила поведения в сети Интернет

При работе с электронной почтой необходимо соблюдать правила безопасной работы, приведенные в таблице 2.3.

Таблица 2.3

№	Правило безопасности
1	Закрывайте окна веб-браузера после окончания работы в библиотеках, интернет-кафе и т.п.
2	Не пользуйтесь электронной почтой для пересылки конфиденциальной информации
3	Архивируйте корреспонденцию, если она делового содержания
4	Письма могут быть восстановлены, так как они годами хранятся в архивных папках на удаленных серверах
5	Не открывайте письма с названием темы типа «наследство», «выигрыш в лотерею» и т.п.
6	Вирусами можно заразиться с адресов друзей, если их компьютер заражен
7	Спам-сообщения помещайте в черный список
8	Включите спам-фильтр
9	Пользуйтесь антивирусными программами с возможностью сканирования вложений электронной почты
10	Применяйте сложные буквенно-цифровые пароли
11	Шифруйте ваши самые важные сообщения

Интернет – это часть нашей жизни. В виртуальной жизни вы не должны вести себя плохо: грубить, проявлять агрессию в адрес других людей, нецензурно выражаться. Очень важно соблюдать **правила вежливости при общении**.



1. Если кто-то проявляет агрессию в ваш адрес, не отвечайте грубостью на грубость.
2. Не выкладывайте в Интернет информацию, оскорбляющую и унижающую достоинство других людей.
3. Регистрируясь на сайтах и форумах, обязательно ознакомьтесь с правилами общения.
4. Играя в онлайн-игры, уважительно относитесь к другим участникам игры.

Для безопасной работы в Интернете желательно устанавливать «детские» браузеры, программы родительского контроля и фильтры поисковых систем.



Пользуетесь ли вы «детским» браузером, стандартными фильтрами, специализированными контролирующими приложениями фильтрации сайтов и т.д.? Поделитесь опытом.



Знание. Понимание



1. Какие опасности подстерегают пользователей сети Интернет?
2. Что означают специальные термины «троллинг», «фишинг», «кибербуллинг», «спам»?
3. Перечислите основные правила поведения в сети Интернет.
4. Каковы правила защиты личных данных?
5. Какие правила вежливого общения в Интернете нужно соблюдать?



Применение



6. К каждой группе угроз приведите примеры:
 - 1) Угроза заражения вредоносным программным обеспечением (ПО).
 - 2) Доступ к нежелательному содержанию.
 - 3) Контакты с незнакомыми людьми с помощью чатов или электронной почты.
 - 4) Поиск развлечений (например, игр) в Интернете.
 - 5) Неконтролируемые покупки.

7. Существуют ли специальные программы для защиты от угроз сети Интернет?
8. Найдите в Интернете сайты, которые защитят компьютер от опасной информации.
9. Зарегистрируйтесь в чате и пообщайтесь, используя правила безопасности.

Анализ



10. Сочините сказку о «золотых правилах» безопасной работы в сети Интернет.

Синтез



11. Нарисуйте рисунки на тему «Безопасный Интернет в картинках».

Рекомендации для выполнения домашнего задания

Для создания рисунков используйте графические редакторы. По желанию можете выбрать растровый или векторный редактор.



Оценивание



12. Составьте сводную таблицу «Азбука безопасного Интернета». Напишите в первом столбце заголовок «Нужно», во втором столбце – заголовок «Нельзя». В строчках запишите семь самых важных правил безопасности поведения подростков в сети Интернет.



Особенности социальной сети Tiktok

В Интернете человек может делать только два действия:

- пассивное действие (наблюдать). Пользователь читает ленту, странички друзей;
- активное действие (постить). Пользователь пишет посты, выкладывает фотографии, видео, ставит лайки.

Социальная сеть – общественное пространство. Для позитивного самовыражения, социального вовлечения и личного прогресса можно применять сервис создания и просмотра коротких видео TikTok (<https://www.tiktok.com>). Сервис TikTok содержит большое число инструментов редактирования, возможность встраивания музыки, эффектов, создание интерактива и получение мгновенной обратной связи

Преимущества TikTok:

- наличие платформы для проявления своей креативности;
- широкий выбор музыки;
- отличная адаптация под различные устройства;

- большое количество эффектов и фильтров;
- возможность замедления и ускорения видео;
- публикация видео в социальные сети, в том числе и в Instagram;
- возможность быстрой коммуникации.

Встроенные эффекты TikTok – это маски или фильтры, которые накладываются поверх видео и делают его более интересным. Можно использовать какой-то один эффект в видео или комбинацию из нескольких. Одни можно добавить в процессе съемки, другие – наложить на готовый ролик.

Дополнительные задания ко второму разделу

Проект 2

Темы для проектных работ

1. «Негативное и положительное влияние компьютера на здоровье человека».
2. «Меры предосторожности и профилактика заболеваний при работе с электронными устройствами».
3. «Безопасность в сети Интернет».
4. «Виды угроз в Интернете».
5. «Правила безопасной работы в электронной почте».

При оформлении проектной работы примените шаблон, представленный в проекте 1: тема проекта, цель, задачи, этапы проведения, рекомендации.

Требования к оформлению проекта

1. Используйте текстовый процессор Microsoft Word.
2. Информацию и картинки можно нарисовать, сфотографировать или сохранить из Интернета.
3. Оформите проект в виде книжки, буклета (буклет – это издание, отпечатанное на листе формата А-3 или А-4 и сложенное один или несколько раз).
4. Напишите мини-эссе или создайте ролик.

Тестовые задания к разделу II

1. Электромагнитное излучение – это:

- А) вид энергии, представляющий волны, возбуждаемые различными излучающими объектами;
- В) это электромагнитные волны, которые возбуждаются разными излучающими объектами;
- С) вид энергии, представляющей волны, возбуждаемые различными неизлучающими объектами;
- Д) электромагнитные волны, невозбуждаемые различными излучающими объектами.

2. Устройства компьютера, которые дают наибольшее электромагнитное излучение.

- А) Монитор.
- В) Принтер.
- С) Системный блок.
- Д) Колонки.

3. Фишинг – это:

- А) сетевое мошенничество;
- В) общение;
- С) спам;
- Д) троллинг.

4. Травля, оскорбления или угрозы с помощью средств электронной коммуникации – это:

- А) фишинг;
- В) кибербуллинг;
- С) спам;
- Д) троллинг.

5. Рассылка навязчивой, нежелательной рекламной информации называется:

- А) фишинг;
- В) кибербуллинг;
- С) спам;
- Д) троллинг.

6. Отметьте вредные факторы, влияющие на состояние здоровья людей, работающих за компьютером.

- A) Проблемы, связанные с электромагнитным излучением.
- B) Проблемы зрения.
- C) Проблемы, связанные с мышцами и суставами.
- D) Стресс, депрессия и другие нервные расстройства, вызванные влиянием компьютера на психику человека.
- E) Негативное влияние на кругозор.

7. Отметьте опасные угрозы Сети.

- A) Вредоносные программы.
- B) Кража компьютера.
- C) Хакерские атаки.
- D) Спам.
- E) Финансовое мошенничество.

8. Это не дает хакерам проникать в компьютер и просматривать файлы и документы.

- A) Применение брандмауэра.
- B) Обновления операционной системы.
- C) Антивирусная программа.
- D) Блокирование спама.

9. Какую информацию нельзя разглашать в Интернете?

- A) Свои увлечения.
- B) Свой псевдоним.
- C) Домашний адрес.
- D) Свой электронный адрес.

10. Действуют ли правила этикета в Интернете?

- A) Интернет – пространство, свободное от правил.
- B) В особых случаях.
- C) Да, как и в реальной жизни.
- D) Нет, не действуют правила этикета.



ОБРАБОТКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

3.1. СТАТИСТИЧЕСКИЕ ДАННЫЕ

ВЫ НАУЧИТЕСЬ

- Использовать абсолютную и относительную ссылки;
- использовать различные форматы данных для решения задач в электронных таблицах.

КЛЮЧЕВЫЕ СЛОВА

Относительная ссылка	Салыстырмалы сілтеме	Relative cell reference
Абсолютная ссылка	Абсолюттік сілтеме	Absolute cell reference
Статистические данные	Статистикалық мәліметтер	Statistical data



Вспомните, для чего используется электронная таблица Excel. Что находится на пересечении строки и столбца? С чего начинается ввод формул?



Одной из функций электронной таблицы Excel является работа с формулами. В формулах указываются не значения, а адреса ячеек, в которых находятся значения. Для быстрого пересчета значений в ячейке со ссылками нужно нажать клавишу **F9**.

Для чего нужны абсолютные и относительные адреса ячеек?

Любая ячейка, которая содержит формулу, называется **зависимой**. От метода адресации ссылок зависит, что будет с ними происходить при копировании формулы из одной ячейки в другую.

Поэтому очень важно отличать **относительную ссылку** на адрес ячейки от **абсолютной ссылки**.

Ссылка – это указание на адрес ячейки.

По умолчанию ссылки на ячейки в формулах рассматриваются как относительные.

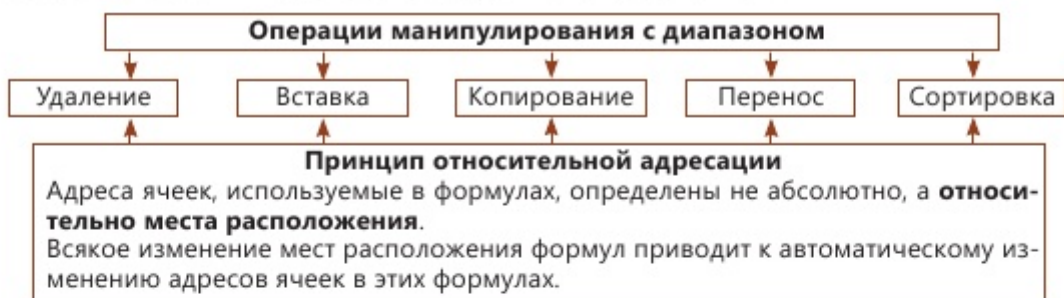


Если при копировании формулы адрес ячейки изменяется, то он называется **относительным**. Относительный адрес имеет вид: **A1, B1, C1** и т.д. При копировании такой формулы вправо или влево будет изменяться заголовок столбца в имени ячейки, а при копировании вверх или вниз – номер строки.

Абсолютный адрес применяется при использовании значения, которое не будет меняться в процессе вычислений. Абсолютный адрес записывают **\$C\$1**. Значение **=C\$1** всегда будет постоянным при копировании формулы в другие ячейки рабочего листа. Для быстрой установки в ячейке абсолютного адреса нужно нажать на клавишу **F4**. Кроме относительной и абсолютной адресации используется **смешанная адресация**. При задании смешанной адресации применяется комбинация относительной и абсолютной адресаций. Одна из составляющих имени ячейки не меняется при копировании. Примеры такой адресации: **\$A1, A\$1**.

При работе в электронной таблице вы часто будете выполнять такие операции с диапазоном ячеек, как удаление, вставка и т.д. (рис. 3.1).

Диапазон – это две или более ячеек листа, выделенных прямоугольной областью. Ссылки в диапазоне разделяются двоеточием.



Например:

	A	B	C	D	E
1					
2				Курс доллара	=321
3	Товар	Цена, тенге	Продажи, шт	Продажи, тенге	Продажи, доллар
4	товар1	1000	130	=B4*C4	=D4/\$E\$2
5	товар2	2000	150	=B5*C5	=D5/\$E\$2
6	товар3	3000	500	=B6*C6	=D6/\$E\$2
7	товар4	2500	200	=B7*C7	=D7/\$E\$2

Рис. 3.1. Пример операции манипулирования с диапазоном ячеек

Ссылки на ячейки **B4, B5, B6, B7** в примере на рисунке 3.1 являются относительными, так как при копировании формулы меняются адреса ячеек.

Ссылка на ячейку **\$E\$2** является **абсолютной**. Она применяется для указания фиксированного значения и адреса ячейки, которые при копировании и перемещении формулы не меняются.

Используя ссылки в формулах, можно быстро провести сложные расчеты для большого количества данных. В электронной таблице Excel средства статистической обработки данных используются для проведения статистического анализа данных.

Внешне **статистическая таблица** представляет собой систему, построенную из горизонтальных строк и вертикальных столбцов. В такой таблице имеются общий заголовок, заголовки столбцов и строк, на пересечении которых записываются **статистические данные**.

В зависимости от задачи применяются различные форматы представления данных. Например, для представления чисел может использоваться несколько различных форматов: **числовой** (100,72), **экспоненциальный** (для больших чисел: 500000000 – 5,00E + 08), **дробный** и **процентный**. Существуют специальные форматы для хранения **дат** (например, 01.09.2017) и **времени** (например, 13:30:55), а также **финансовый** и **денежный** форматы.

Постройте таблицу умножения Пифагора, используя относительные, абсолютные ссылки в формулах.



Шаг за шагом



Создайте таблицу умножения от 1 до 9, выполнив действия по шагам.

1. Введите данные в ячейках с **B1** по **J1** – **1, 2, ..., 9**. Для заполнения числами **1, 2, ...** примените **Маркер автозаполнения**. Установите **Формат ячеек – Числовые форматы – Общий формат** (рис. 3.2).

2. Выполните аналогичные действия для ячеек с **A2** по **A10**.

3. Выделите столбцы от **A** до **J**. Установите ширину столбца **5,00** (или **40** пикселей). Для строк установите высоту **30,00** (или **40** пикселей).

4. Выделите диапазон ячеек **A1:J10**. Установите **Границы – Все границы**.

5. А теперь попробуйте в ячейку **B2** ввести формулу: **=B\$1*\$A2**. Зафиксируйте **строку 1** и **столбец A**, поставив перед ними знак **\$**. Можно нажать функциональную клавишу **F4**, чтобы быстро вставить знак **\$** (абсолютная адресация).

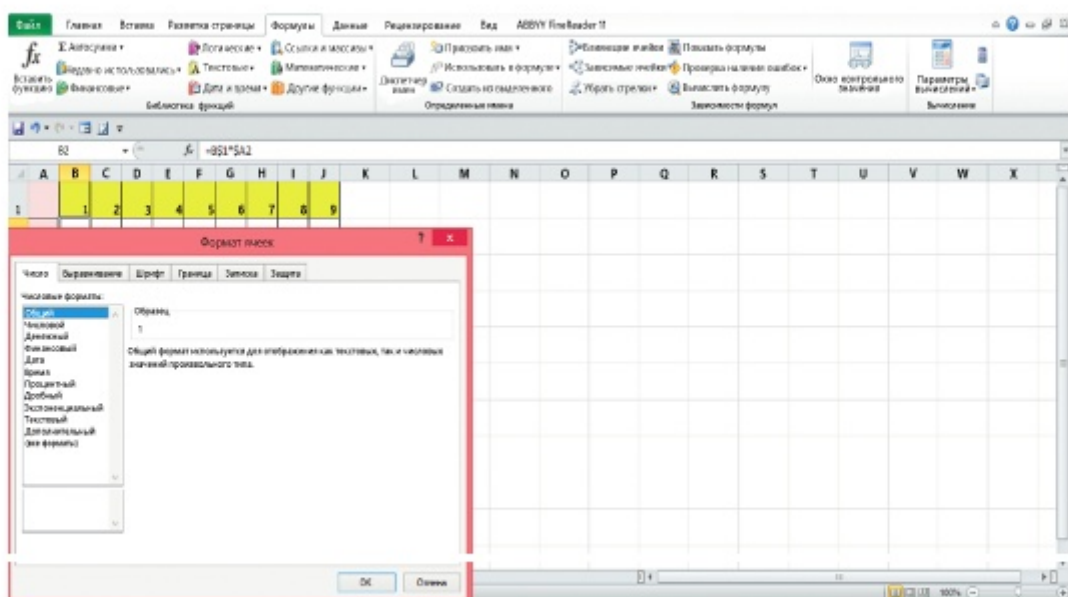


Рис. 3.2. Формат ячеек

	B2	C2	D2	E2	F2	G2	H2	I2	J2
1	1	2	3	4	5	6	7	8	9
2	1	2	3	4	5	6	7	8	9
3	2	4	6	8	10	12	14	16	18
4	3	6	9	12	15	18	21	24	27
5	4	8	12	16	20	24	28	32	36
6	5	10	15	20	25	30	35	40	45
7	6	12	18	24	30	36	42	48	54
8	7	14	21	28	35	42	49	56	63
9	8	16	24	32	40	48	56	64	72
10	9	18	27	36	45	54	63	72	81

Рис. 3.3. Таблица умножения от 1 до 9

6. Скопируйте формулу на всю таблицу. В итоге вы увидите таблицу умножения (рис. 3.3).

7. Проанализируйте полученную таблицу умножения (рис. 3.4). Каким образом меняются относительные и абсолютные ссылки в таблице?

8. Поменяйте данные в строке заголовка, например, **10, 11, ..., 19**. При этом изменятся данные в столбцах. Что произойдет с таблицей умножения?

9. Вставьте строки над таблицей и введите заголовок. Сохраните файл.



В ячейку **A10** ввели формулу $= (A5+A9)/\$D\1 . Затем эту формулу скопировали в ячейку **A15**. Какая формула содержится в ячейке **A15**? Проверьте на практике правильность выполнения.

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	=B\$1*\$A2	=C\$1*\$A2	=D\$1*\$A2	=E\$1*\$A2	=F\$1*\$A2	=G\$1*\$A2	=H\$1*\$A2	=I\$1*\$A2	=J\$1*\$A2
3	2	=B\$1*\$A3	=C\$1*\$A3	=D\$1*\$A3	=E\$1*\$A3	=F\$1*\$A3	=G\$1*\$A3	=H\$1*\$A3	=I\$1*\$A3	=J\$1*\$A3
4	3	=B\$1*\$A4	=C\$1*\$A4	=D\$1*\$A4	=E\$1*\$A4	=F\$1*\$A4	=G\$1*\$A4	=H\$1*\$A4	=I\$1*\$A4	=J\$1*\$A4
5	4	=B\$1*\$A5	=C\$1*\$A5	=D\$1*\$A5	=E\$1*\$A5	=F\$1*\$A5	=G\$1*\$A5	=H\$1*\$A5	=I\$1*\$A5	=J\$1*\$A5
6	5	=B\$1*\$A6	=C\$1*\$A6	=D\$1*\$A6	=E\$1*\$A6	=F\$1*\$A6	=G\$1*\$A6	=H\$1*\$A6	=I\$1*\$A6	=J\$1*\$A6
7	6	=B\$1*\$A7	=C\$1*\$A7	=D\$1*\$A7	=E\$1*\$A7	=F\$1*\$A7	=G\$1*\$A7	=H\$1*\$A7	=I\$1*\$A7	=J\$1*\$A7
8	7	=B\$1*\$A8	=C\$1*\$A8	=D\$1*\$A8	=E\$1*\$A8	=F\$1*\$A8	=G\$1*\$A8	=H\$1*\$A8	=I\$1*\$A8	=J\$1*\$A8
9	8	=B\$1*\$A9	=C\$1*\$A9	=D\$1*\$A9	=E\$1*\$A9	=F\$1*\$A9	=G\$1*\$A9	=H\$1*\$A9	=I\$1*\$A9	=J\$1*\$A9
10	9	=B\$1*\$A10	=C\$1*\$A10	=D\$1*\$A10	=E\$1*\$A10	=F\$1*\$A10	=G\$1*\$A10	=H\$1*\$A10	=I\$1*\$A10	=J\$1*\$A10

Рис. 3.4. Таблица умножения в формулах

Знание. Понимание



1. Где используются адреса ячеек?
2. Какие адреса ячеек встречаются в формулах?
3. В чем отличие относительной ссылки от абсолютной? Приведите примеры.
4. Могут ли в адресе ячейки формулы использоваться и относительная, и абсолютная адресации?
5. Какие форматы данных чаще всего используют в электронных таблицах?
6. Установите соответствие.



1	Относительный адрес	A	Перечисление имени столбца и номера строки, на пересечении которых находится ячейка
2	Абсолютный адрес	Б	Это адрес ячеек в формулах, который при копировании изменяется относительно места расположения формулы
3	Адрес ячейки	В	Это адрес ячеек в формулах, который при копировании меняется частично: или только номер строки, или только имя столбца
4	Смешанный адрес	Г	Это адрес ячеек в формулах, который при копировании не изменяется (фиксирование адреса)

7. Установите соответствие.

1	R5, G23, J122	А	Смешанный адрес ячейки в формуле
2	\$R5, G\$23, \$J122	Б	Абсолютный адрес ячейки в формуле
3	\$R\$5, \$G\$23, \$J\$122	В	Относительный адрес ячейки в формуле



Применение



8. Примените теоретические знания в практической ситуации.

1) Оформите на листе электронной таблицы Excel расписание школьных звонков. За исходные данные возьмите начало урока и продолжительность урока (рис. 3.5, а).

Установите курсор в ячейку **B8** и введите формулу **=C7+D7** (к концу урока прибавляем переменную).

Формулы из ячеек **B8** и **C7** скопируйте.

Получите таблицу «Расписание звонков» (рис. 3.5, б).

	A	B	C
1	Расписание звонков		
2			
3			Начало урока
4		Продолжительность урока	
5			
6	№ урока	Начало урока	Конец урока
7	1	=D3	=B7+\$D\$4
8	2	=C7+D7	=B8+\$D\$4
9	3	=C8+D8	=B9+\$D\$4
10	4	=C9+D9	=B10+\$D\$4
11	5	=C10+D10	=B11+\$D\$4
12	6	=C11+D11	=B12+\$D\$4
13	7	=C12+D12	=B13+\$D\$4
14	8	=C13+D13	=B14+\$D\$4
15	9	=C14+D14	=B15+\$D\$4
16	10	=C15+D15	=B16+\$D\$4

а)

	A	B	C	D
1	Расписание звонков			
2				
3			Начало урока	8:30
4		Продолжительность урока		0:40
5				
6	№ урока	Начало урока	Конец урока	Перемена
7	1	8:30	9:10	0:05
8	2	9:15	9:55	0:15
9	3	10:10	10:50	0:05
10	4	10:55	11:35	0:20
11	5	11:55	12:35	0:05
12	6	12:40	13:20	0:20
13	7	13:40	14:20	0:05
14	8	14:25	15:05	0:15
15	9	15:20	16:00	0:05
16	10	16:05	16:45	

б)

Рис. 3.5. Таблица **Расписание звонков** (в виде формул и данных)

2) На рисунках 3.6, 3.7 даны две таблицы «Расписание уроков». Заполните ячейки, залитые желтым цветом, формулами.

Расписание уроков

Прямая задача			
Урок	Начало урока	Конец урока	Перемена после урока
1	8:30	9:15	
2	9:20	10:05	
3	10:35	11:20	
4	11:45	12:30	
5	12:35	13:20	
6	13:40	14:25	
7	14:40	15:25	
8	15:30	16:15	

Рис. 3.6. Прямая задача

Расписание уроков

Обратная задача			
Продолжительность перемен после урока	Урок	Начало урока	Конец урока
0:10	1	8:30	
0:15	2		
0:05	3		
0:20	4		
0:15	5		
0:10	6		
0:05	7		
	8		

Рис. 3.7. Обратная задача

Анализ

9. В ячейке **B2** (табл. 3.1) записано количество информации в байтах. Получите в ячейке **B3:B5** количество информации в единицах измерения, указанных в соответствующих ячейках столбца **A**. Проанализируйте полученные данные.



Таблица 3.1

	А	В	С
1			
2	Количество информации в байтах:		
3	Количество информации в битах:		
4	Количество информации в килобайтах:		
5	Количество информации в мегабайтах:		



Синтез



10. Информация о распределении суши и воды на земном шаре приведена в таблице 3.2.

Таблица 3.2

Поверхность земного шара	Северное полушарие		Южное полушарие		Земля в целом	
	в млн кв. км	в %	в млн кв. км	в %	в млн кв. км	в %
Суша	100,41		48,43			
Вода	154,64		206,62			
Всего						

Используя электронную таблицу, заполните данными пустые клетки приведенной таблицы. Обобщите полученную таблицу. Предложите альтернативу формулам.



Оценивание



11. По таблице 3.3 рассчитайте сумму оплаты электроэнергии за каждый месяц. Стоимость 1 кВт/ч составляет 16,65 тенге. Оцените счет при изменении стоимости 1 кВт/ч.

Таблица 3.3

Счет на оплату электроэнергии

Стоимость 1 кВт/ч

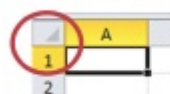
16,65 тенге

Месяц	Дата	Показания счетчика	Расход кВт/ч	Сумма (тенге)
Январь	30.01.2020	540	540	
Февраль	28.02.2020	1026	486	
Март	29.03.2020	1480	454	
Апрель	30.04.2020	2158	678	
Май	28.05.2020	2996	838	
Июнь	29.06.2020	3458	462	
Июль	28.07.2020	4021	563	
Август	30.08.2020	4684	663	
Сентябрь	25.09.2020			
Октябрь	26.10.2020			
Ноябрь	30.11.2020			
Декабрь	25.12.2020			



Выделение всех ячеек

Чтобы выделить все ячейки, нажмите **Ctrl + A** или специальную кнопку



в углу листа.

Быстрый ввод данных в несколько ячеек

Чтобы сэкономить время при вводе новых данных (даты, строки, числа, формулы) или при замене уже существующих, выделите ячейки (можно несмежные), введите данные и нажмите **Ctrl + Enter**.

Как восстановить файл, который вы не сохранили

Для восстановления несохраненного файла в Excel 2010 нажмите на **Файл**, выберите **Последние** (рис. 3.8). В правом нижнем углу экрана появится опция **Восстановить несохраненные книги**.

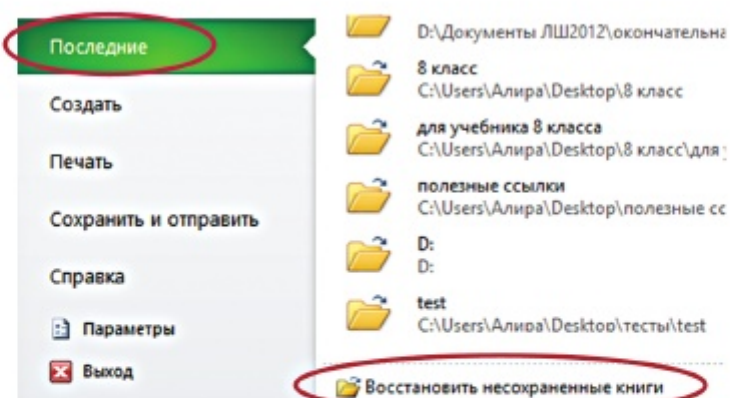
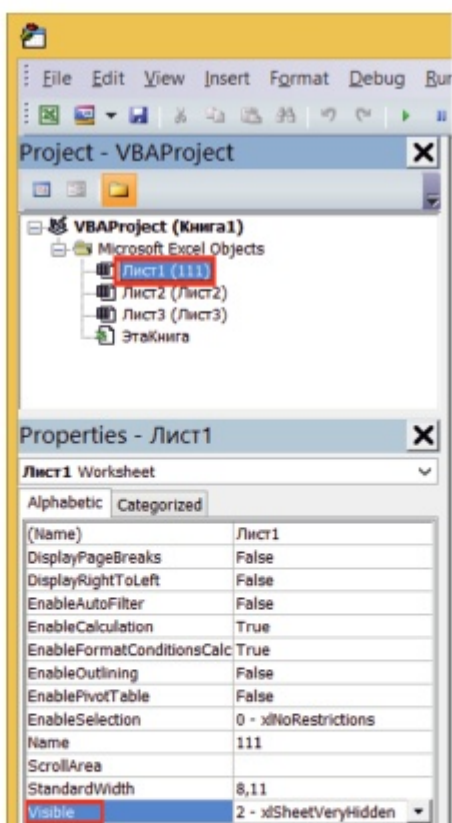


Рис. 3.8. Восстановление несохраненного файла в Excel 2010

В версии 2013 года путь такой: **Файл – Сведения – Управление версиями**.



Супертайный лист

Допустим, вы хотите скрыть часть листов в электронной таблице Excel от других пользователей, работающих над книгой. Это можно сделать классическим способом – щелкните по ярлычку листа правой кнопкой мыши и в контекстном меню выберите команду **Скрыть**. В этом случае имя скрытого листа всё равно будет видно другому человеку. Чтобы сделать его абсолютно невидимым, нужно действовать так:

1. Нажмите **Alt + F11**.
2. Выберите лист в **VBAProject** (рис. 3.9).
3. Слева появится окно **Microsoft Visual Basic for Applications**.
4. Найдите свойство **Visible** и сделайте его **xlSheetVery Hidden**.
5. Лист будет скрыт.

Рис. 3.9. Окно **Microsoft Visual Basic for Applications**

3.2. ВСТРОЕННЫЕ ФУНКЦИИ

ВЫ НАУЧИТЕСЬ

Использовать встроенные функции для решения задач с применением электронных таблиц.

КЛЮЧЕВЫЕ СЛОВА

Встроенные функции	Кірістірілген функциялар	Builtin functions
Мастер функций	Функция шебері	Function wizard
Логическое выражение	Логикалық өрнек	Boolean expression
Табулирование функции	Кестелік функциялар	Tabulating function



Кроме простых арифметических операций, электронная таблица Excel содержит достаточно большой набор **встроенных функций**. Функции в Excel упорядочены по категориям в зависимости от функциональной области. Функций несколько сотен.

Функция – встроенный инструмент Excel, способный вернуть значение в зависимости от переданного ему параметра. Она предназначена для расчетов, вычислений и анализа данных. Каждая функция может включать в себя **константу, оператор, ссылку, имя ячейки (диапазона ячеек)** и **формулы**.



Имя – это краткое обозначение, представляющее ячейку, диапазон ячеек, формулу, таблицу или значение константы. Можно создавать свои собственные определенные имена. С помощью поля **Имя** можно быстро находить и выделять конкретные ячейки или диапазоны. Для этого в поле, расположенное слева от строки формул нужно ввести имена или ссылки на ячейки.

Вы уже знакомы с понятием «диапазон». Диапазон рассматривается как единое целое. Прямоугольный блок ячеек называется **смежным диапазоном** и обозначается с помощью двух ссылок, разделенных двоеточием. Первая ссылка указывает на ячейку в левом верхнем углу диапазона, а вторая – на ячейку в правом нижнем углу.

Константа представляет собой число либо текст. Она не меняется в процессе расчетов.

Операторы бывают трех видов:

1. **Арифметический оператор** предназначен для выполнения арифметических действий. Он возвращает числовое значение.

2. **Оператор сравнения** предназначен для сравнения данных. Он возвращает логическое значение **ИСТИНА** или **ЛОЖЬ** (**TRUE** или **FALSE**).

3. **Текстовый оператор** применяется для объединения данных (в Excel им является **амперсанд** – &).

Каждая встроенная функция Excel имеет название или уникальное имя. Синтаксис встроенной функции:



Название функции (Аргументы)

где **Название функции** – уникальное имя функции. **Аргументы** – список аргументов функции или его входные данные.

Встроенные функции Excel работают по принципу:

- задать функции входным данным;
- функция вычисляет результат или возвращает выходные данные.

Имена функций могут быть набраны как строчными, так и заглавными буквами. В зависимости от функции ее аргументы могут представлять собой число, строку, логическое значение, ссылку и т.д. В качестве аргументов функции можно также использовать другие функции и формулы.

Чтобы вставить формулу с функцией, нужно щелкнуть по кнопке с изображением символа функции  слева от строки формул. Можно также воспользоваться специальной кнопкой  **Вставить функцию** на вкладке **Формулы**. В каждом из этих случаев открывается диалоговое окно **Мастер функций**, представленное на рисунке 3.10.

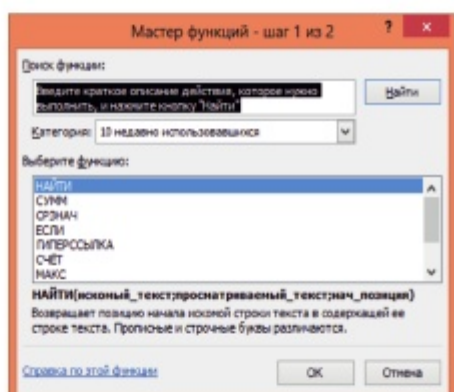


Рис. 3.10. Мастер функций



Рис. 3.11. Встроенные функции

В раскрывающемся списке **Категория** (рис. 3.11) можно выбрать ту категорию, к которой относится искомая функция. Функцию также можно найти с помощью поисковой строки **Поиск функции...**. Для этого введите в поле первые несколько букв ее названия или слово. Можно выбрать элемент из **Полного алфавитного перечня**. После того как функция выбрана, открывается диалоговое окно **Аргументы функции**, где в поля вводятся аргументы функции.

Рассмотрим простейший пример вычисления среднего значения данных в диапазоне ячеек **C1:C10**.



Для вычисления среднего арифметического значения можно использовать формулу:

$$=(C1 + C2 + C3 + C4 + C5 + C6 + C7 + C8 + C9 + C10)/10.$$

Но проще применить функцию из категории **Статистические** =СРЗНАЧ(C1:C10).

Применим на практике функции из категории **Текстовые**: СИМВОЛ и КОДСИМВ. Как вам известно, система ASCII (коды для представления цифр, букв и других символов) использует числа от 0 до 127 (в некоторых системах до 255). Функции СИМВОЛ и КОДСИМВ тесно связаны с системой ASCII. Например, функция СИМВОЛ возвращает символ, который соответствует числовому коду ASCII.

Синтаксис: =СИМВОЛ(число)

Функция КОДСИМВ возвращает код ASCII для первого символа ее аргумента.

Синтаксис: =КОДСИМВ(текст)

Например, формула =СИМВОЛ(83) возвратит букву **S**.

	A1				
	A	B	C	D	E
1	S				

Формула =КОДСИМВ(«S») возвратит значение **83**.

Аналогично, если первым символом в ячейке **A1** является буква **S**, формула =КОДСИМВ(A1) возвратит код **83**. Поскольку цифры также являются символами, аргумент КОДСИМВ может быть числовым. Например, формула =КОДСИМВ(«8») возвращает **56**, то есть код ASCII символа **8**.

1. С помощью функции **КОДСИМВ** узнайте коды, принадлежащие буквам A, D, Y, L, R, N, M.
2. С помощью функции **СИМВОЛ** определите символы, закодированные кодами 59, 67, 70, 78.





В одну из ячеек будет введено число. В другой ячейке нужно получить ответ на вопрос, является ли введенное число отрицательным. Если число отрицательное, то «Да», иначе «Нет».

Для решения данной задачи рассмотрим функцию **ЕСЛИ()** из категории **Логические**. Эта функция является одним из мощных инструментов в мире функций.

Синтаксис логической функции:

ЕСЛИ (лог_выражение; значение_если_истина)

где **лог_выражение** – логическое выражение, возвращающее значение **ИСТИНА** или **ЛОЖЬ**.

Логические выражения строятся с помощью операций отношения (<, >, <= (меньше или равно), >= (больше или равно), =, <> (не равно)) и логических операций (логическое **И**, логическое **ИЛИ**, логическое отрицание **НЕ**).

На рисунке 3.12 представлено решение задания об отрицательном числе. Проанализируйте введенную логическую функцию **ЕСЛИ()**.

ЕСЛИ		=ЕСЛИ(A2<0;"Да";"Нет")				
A	B	C	D	E	F	
1	Число	Введенное число отрицательное?				
2	-100	=ЕСЛИ(A2<0;"Да";"Нет")				

Аргументы функции

ЕСЛИ

Лог_выражение: = ИСТИНА

Значение_если_истина: = "Да"

Значение_если_ложь: = "Нет"

= "Да"

Проверяет, выполняется ли условие, и возвращает одно значение, если оно выполняется, и другое значение, если нет.

Лог_выражение любое значение или выражение, которое при вычислении дает значение ИСТИНА или ЛОЖЬ.

Значение: Да

[Справка по этой функции](#)

Рис. 3.12. Функция **ЕСЛИ()**

При проведении множественных логических проверок используется функция **ЕСЛИ()** внутри другого **ЕСЛИ()**, функции **И()** и функции **ИЛИ()**.

Существуют особенности записи логических операций. Сначала записывается имя логической операции (**И**, **ИЛИ**, **НЕ**), а затем в круглых скобках перечисляются логические операнды. Например:

= **ЕСЛИ(И(A5 > D1; B3 < 5); "Да"; "Нет")**

Содержимое ячейки **A5** больше содержимого ячейки **D1**, а содержимое ячейки **B3** меньше **5**. Если условие выполняется, то в выбранной клетке будет отображаться «Да», в противном случае – «Нет».

Нужно протабулировать функцию $y = f(x)$ с шагом **0,2**:

$$y = \begin{cases} 0, & \text{если } x < 0 \\ x, & \text{если } x \in [0,1] \\ 1, & \text{если } x > 1 \end{cases}$$



Табулирование функции – это вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения с определенным шагом.



На рисунке 3.13: **x** – аргумент, **y** – результат функции, шаг = **0,2**. Каким образом меняются значения в столбце **x**? Проанализируйте формулу логической функции вложенного **ЕСЛИ()**. Соответствует ли результат функции условиям функции $y = f(x)$?

	A	B	C	D	E	F	G
1	x	y	шаг				
2	-0,4	0	0,2				
3	-0,2	0					
4	0	0					
5	0,2	0,2					
6	0,4	0,4					
7	0,6	0,6					
8	0,8	0,8					
9	1	1					
10	1,2	1					
11	1,4	1					
12	1,6	1					

Рис. 3.13. Вложенное **ЕСЛИ()**



Знание. Понимание



1. Каким образом упорядочены встроенные функции?
2. Как вы понимаете понятие «встроенная функция»?
3. Каков синтаксис встроенной функции?
4. По какому принципу работают встроенные функции?
5. Можно ли имя функции набрать строчными буквами?
6. Каким образом можно вставить встроенную функцию?
7. Перечислите, какие категории входят во встроенные функции.
8. Каков синтаксис логической функции **ЕСЛИ()**?
9. С помощью каких логических операций строится логическое выражение?
10. Что означает вложенное **ЕСЛИ()**?
11. Каким образом происходит табулирование функции?



Применение



12. Чтобы получить допуск к экзамену, учащиеся должны успешно сдать зачет. Результаты класса занесите в таблицу с графами: **Фамилия, имя; Зачет; Экзамен** (рис. 3.14). Составьте функцию **ЕСЛИ()** в ячейке **C2** и скопируйте с помощью маркера заполнения.

	А	В	С
1	Фамилия, имя	Зачет	Экзамен
2	Дбенов В.	зачет	допущен
3	Ан М.	зачет	допущен
4	Алимов А.	незачет	не допущен
5	Крузе Н.	зачет	допущен
6	Сахарова Т.	незачет	не допущен

Рис. 3.14. Таблица зачетов



Анализ



13. Создайте таблицу для анализа успеваемости учащихся. Если ученик получит 5 баллов – «отлично», 4 – «хорошо», 3 – «удовлетворительно». Составьте вложенное условие **ЕСЛИ()** (рис. 3.15) в ячейке **C5** и скопируйте в остальные ячейки.

	А	В	С
4	Фамилия, имя	Баллы	Результат
5	Абенов В.	5	отлично
6	Ан М.	5	отлично
7	Алимов А.	2	неудовлетворительно

Рис. 3.15. Таблица Анализ успеваемости

Синтез



14. На рисунке 3.16 представлена таблица «Заказ на мягкую мебель». Проведите расчеты в столбцах с заголовками **ВСЕГО**, **ИТОГО**. Найдите минимальную, максимальную и среднюю стоимость заказа. Обобщите данные в таблице.

	A	B	C	D	E
1	Заказ на мягкую мебель				
2					
3	№	Тип салона мебели	Цена за 1 комплект	Количество комплектов	ВСЕГО
4	1	МЕРЕЙ	T408 000,0	2	
5	2	ЖАННА	T218 000,0	4	
6	3	БРАВО	T245 000,0	3	
7	4	БЕЛОРУССКАЯ МЕБЕЛЬ	T305 000,0	5	
8	5	RED APPLE	T230 800,0	4	
9	6	ARMADA	T439 000,0	6	
10	7	ДЕТСКАЯ	T110 870,0	10	
11	8	ZETA	T315 000,0		
12		ИТОГО:			
13					
14		Минимальная стоимость заказа			
15		Максимальная стоимость заказа			
16		Средняя стоимость заказа			
17					

Рис. 3.16. Заказ на мягкую мебель

Рекомендации к выполнению задания

1. Формат ячеек – Денежный – Казахский.
2. Для расчета **ВСЕГО** примените формулу «Цена за 1 комплект умножить на количество комплектов».
3. Для расчета **ИТОГО** используйте математическую функцию СУММ (число1; число2;...) – сумма аргументов.
4. Для расчета минимальной, максимальной и средней стоимости заказа используйте статистические функции МИН, МАКС, СРЗНАЧ.

Статистические функции:

МИН(число1;число2;...) возвращает минимальное значение из списка аргументов.

МАКС(число1;число2;...) возвращает максимальное значение из списка аргументов.

СРЗНАЧ(число1;число2;...) возвращает среднее арифметическое аргументов.



Оценивание



15. Проведите исследование, используя данные, приведенные в таблице 3.4. Для этого выбраны популярные национальные блюда:

- английское «Ростбиф с овощами»;
- казахское «Бешбармак»;
- русское «Пюре с котлетой»;
- японское «Ролл».

Известно, что суточная потребность калорий в день в белках, жирах, углеводах и энергии для подростков от 11 до 13 лет составляет:

Подростки	Белки, г	Жиры, г	Углеводы, г	Энергетическая ценность, ккал
Мальчики	93	93	370	2700
Девочки	85	85	340	2450

16. Какие расчеты можно выполнить в таблице с помощью встроенных функций?

17. Почему ингредиент «соль» отсутствует во всех блюдах в таблице?

18. Какие из перечисленных блюд по энергетической ценности в день наиболее подходят подросткам?

19. Какие проблемы могут возникнуть, если растущий организм не насыщается необходимыми питательными веществами?

20. Сделайте заключение, учитывая рекомендации для здорового питания подростков.

Таблица 3.4

Энергетическая ценность популярных национальных блюд

№	Национальная кухня	Белки	Жиры	Углеводы	Калории
1	Английская				
	<i>Ростбиф с овощами</i>				
	говядина	18,5	16	0	218
	цветная капуста	2,5	0,3	5,4	30
	морковь	1,3	0,1	9,3	34
2	Казахская				
	<i>Бешбармак</i>				
	говядина	18,5	16	0	218
	мука	10,3	1,1	69	334
	лук	1,4	0	10,4	41

№	Национальная кухня	Белки	Жиры	Углеводы	Калории
	картофель	2	0,4	18,1	80
	морковь	1,3	0,1	9,3	34
3	Русская				
	<i>Пюре с котлетой</i>				
	говядина	18,5	16	0	218
	лук	1,4	0	10,4	41
	сливочное масло	0,8	72,5	1,3	661
	чеснок	6,5	0	6	46
	ржаной хлеб	6	1,2	41,8	181
	яйцо	12,7	10,9	0,7	155
	картофель	2	0,4	18,1	80
4	Японская				
	<i>Ролл</i>				
	рис	0,6	7	73,7	330
	осетр	10,9	16,4	0	100
	огурец	0,8	0,1	3,8	14
	редька	1,9	0,2	8	35



Художник из Японии Тацуо Хориучи с помощью Microsoft Excel создает красивые произведения. Его картины – сложные и изысканные естественные японские пейзажи, усложненные культурными мотивами (рис. 3.17 а, б).

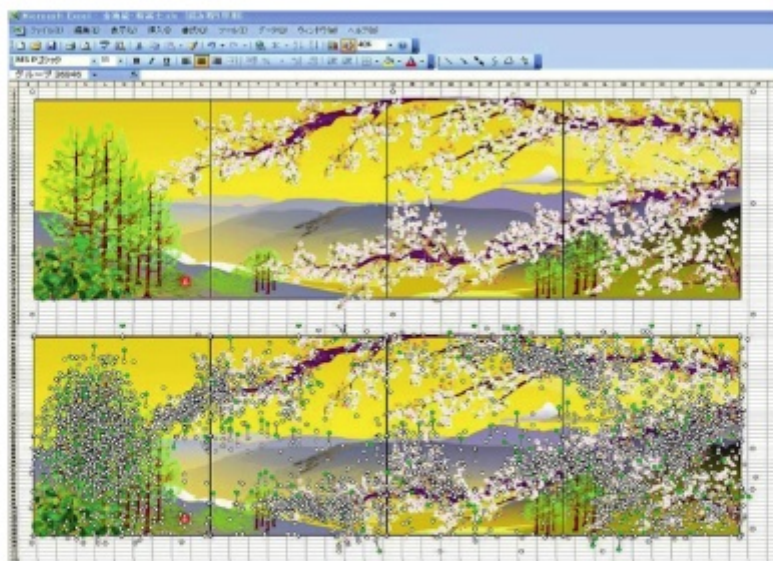


Рис. 3.17, а. Картины, созданные в Microsoft Excel

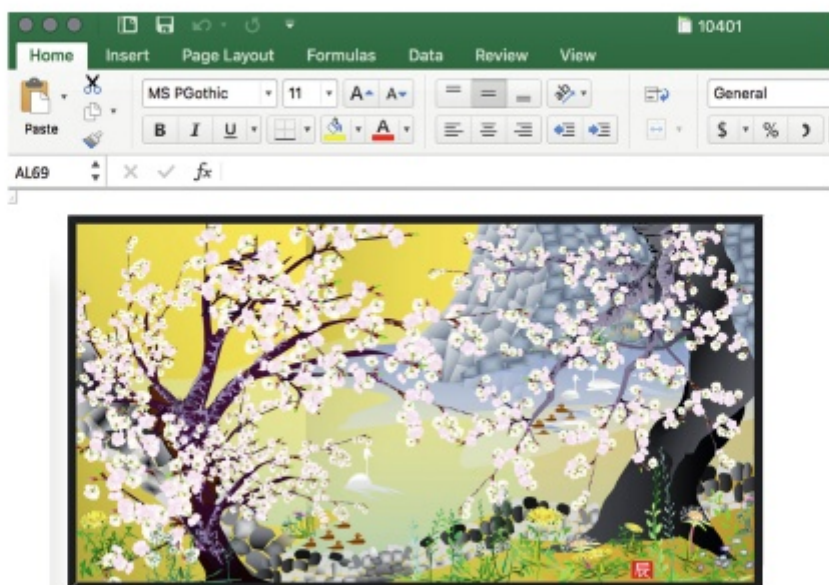


Рис. 3.17, 6. Картины, созданные в Microsoft Excel

3.3. АНАЛИЗ ДАННЫХ НА ОСНОВЕ ИМЕЮЩЕЙСЯ ИНФОРМАЦИИ

ВЫ НАУЧИТЕСЬ

Использовать встроенные функции для решения задач с применением электронных таблиц.

КЛЮЧЕВЫЕ СЛОВА

Анализ данных	Деректерді талдау	Data analysis
Спарклайн	Спарклайн	Sparkline
Сводная таблица	Жиынтық кестесі	Summary table
Итоги	Нәтижелері	Results

Электронная таблица Excel – эффективное средство для анализа больших объемов данных. Данные в электронной таблице можно вычислять, сортировать и фильтровать, создавать профессионально оформленные диаграммы. Кроме того, можно применять стиль таблицы, создавать сводные таблицы, быстро добавлять итоги, применять условное форматирование, средства сортировки и фильтрации с помощью встроенных функций (математических, финансовых, логических, статистических) и т.д. Мы рассмотрим некоторые из них.

Наиболее подходящий и быстрый способ анализа данных в электронной таблице Excel – это построение диаграмм и графиков, миниатюрных

графиков – **спарклайнов**. От качества их построения зависит точность анализа данных. Рассмотрим некоторые способы анализа.

Предприятие по производству мебели выпускает несколько видов продукции: шкафы (трех видов), столы (пяти видов) и стулья (двух видов). В таблице 3.5 приведены данные об объемах продаж за первый квартал текущего года. На основе этих данных нужно проанализировать, какие виды продукции пользуются большим спросом, и принять решение об увеличении/уменьшении объемов производства. Помните, как создавать гистограмму.



Таблица 3.5

Данные из условия задачи

	Продукция	Объем продаж (шт.)	Суммарный объем
Шкафы	шкаф 1	135	533
	шкаф 2	80	
	шкаф 3	318	
Стол	стол 1	200	481
	стол 2	55	
	стол 3	73	
	стол 4	90	
	стол 5	63	
Стулья	стул 1	505	805
	стул 2	300	

Чтобы решить данную проблему, нужно представить данные по всем видам продукции с помощью **гистограммы** (рис. 3.18).



Рис. 3.18. Данные о продажах, представленные с помощью гистограммы



Рассмотрите вторую альтернативу – создайте и рассмотрите данные по всем видам продукции с помощью **линейчатой диаграммы**. Она похожа на гистограмму, только столбцы расположены горизонтально.

Проанализируйте диаграммы, полученные в первом и во втором решениях. Какой из видов продукции пользуется наибольшим спросом, а какой – наименьшим? Какие решения необходимо принять об увеличении/уменьшении объемов производства?



Рассмотрите третью альтернативу – создайте и рассмотрите данные по всем видам продукции с помощью **круговой диаграммы**.

Что показывает третья решение – круговая диаграмма? Какую часть от общего объема продаж составляет объем продаж каждого из видов продукции?

Какие функции анализа можно еще использовать? Рассмотрим **спарклайны**.

Спарклайны – это отслеживание тенденции в рядах данных (колебание цен, объем продаж). Спарклайны располагаются внутри ячеек и представляют собой мини-графики. Они наглядно отображают поведение данных, максимальные и минимальные значения и другие особые точки.

Отличие спарклайнов от диаграмм в мгновенной скорости их создания. Изучите внимательно, как создаются спарклайны.

1. Выберите вкладку **Вставка** – группа **Спарклайны** – кнопка **График** (рис. 3.19).

	A	B	C	D	E	F	G	F	G
		Продукция	Объем продаж (шт.) 1 квартал	Объем продаж (шт.) 2 квартал	Объем продаж (шт.) 3 квартал	Объем продаж (шт.) 4 квартал		Объем продаж (шт.) 4 квартал	
2		шкаф 1	135	137	139	141		141	
3	Шкафы	шкаф 2	80	81	82	83		83	
4		шкаф 3						410	
5		стол 1						307	
6		стол 2						70	
7	Столы	стол 3						100	
8		стол 4						107	
9		стол 5						90	
10	Стулья	стул 1						707	
11		стул 2						555	

Создание спарклайнов ? x

Выберите нужные данные:

Диапазон данных:

Выберите место для размещения спарклайнов:

Диапазон размещения:

Рис. 3.19. Создание спарклайнов

2. В появившемся окне **Создание спарклайнов** выберите место для размещения спарклайнов. Обычно диапазон размещения указывают рядом с данными.

3. Нажмите **ОК**.

Если выделить ячейку со спарклайном, появится вкладка для работы с ним.

Поэкспериментируйте: измените минимальные и максимальные точки, стиль, цвет спарклайна, цвет маркера, тип спарклайна и т.д.



Альтернативой спарклайнам является **Условное форматирование** с правилом **Цветовые шкалы** (рис. 3.20).

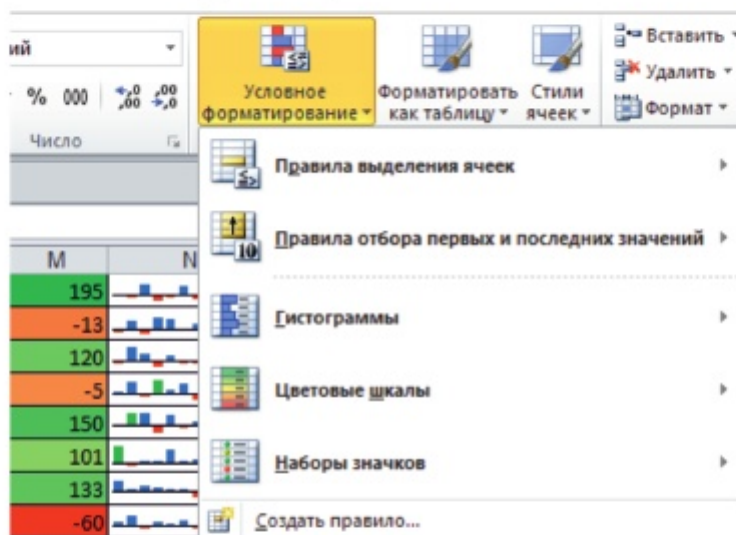


Рис. 3.20. Условное форматирование

Анализировать данные в Excel можно также с помощью встроенных функций и других инструментов для анализа данных, представленных на рисунке 3.21.

<p>Данные Рецензирование Вид Разработчик Office Tab Настройки</p> <p>Группировать (Shift+Alt+стрелка вправо)</p> <table border="1"> <tr><td>1</td><td></td><td></td></tr> <tr><td>2</td><td>A</td><td>2</td></tr> <tr><td>3</td><td>B</td><td>5</td></tr> <tr><td>4</td><td>C</td><td>7</td></tr> <tr><td>5</td><td>A+B+C</td><td>14</td></tr> </table> <p>Объединение диапазона ячеек таким образом, что все эти ячейки можно одновременно свернуть или расширить.</p> <p>Группировать</p> <p>Разгруппировать</p> <p>Промежуточный Структура</p>	1			2	A	2	3	B	5	4	C	7	5	A+B+C	14	<p>Группировка данных – объединение диапазона ячеек</p>
1																
2	A	2														
3	B	5														
4	C	7														
5	A+B+C	14														

	<p>Консолидация данных – объединение нескольких диапазонов данных в один диапазон</p>
	<p>Сортировка и фильтрация изменяют порядок строк</p>
	<p>Промежуточные итоги используются при работе со списками</p>
	<p>Сводная таблица – анализ больших данных</p>

Рис. 3.21. Инструменты для анализа данных



При анализе большого объема данных в таблице применяется сводная таблица. **Сводная таблица** – это отчет, построенный на основе табличных данных.



Создадим сводную таблицу для анализа больших данных в таблице.

Шаг за шагом



1. Постройте простую таблицу «Продажа канцелярских товаров».
2. Щелкните на ячейке **A1** для определения данных, с которыми нужно будет работать.
3. На вкладке **Вставка** из группы **Таблицы** выберите **Сводная таблица**. Откроется окно **Создание сводной таблицы** (рис. 3.22).
4. Укажите вариант **Выбрать таблицу** или **диапазон**.
5. Укажите диапазон и место расположения на текущем листе, например **F1**.
6. Появится форма сводной таблицы и **Список полей сводной таблицы**. На панели находятся:

1) **Выберите поля для добавления в отчет.** Для того чтобы добавить/убрать поле в сводную таблицу, поставьте/снимите галочку напротив названия необходимого поля или перетащите поля данного списка.

2) **Фильтр отчета.** В этот список надо перемещать поля, по которым будут фильтроваться данные отчета. За счет этого поля создается многомерность отчета.

3) **Названия столбцов.** В данный список перемещают поля, значения которых будут располагаться в названиях столбцов сводной таблицы.

4) **Названия строк.** В данный список перемещают поля, значения которых будут располагаться в названиях строк сводной таблицы.

5) **Значения.** Сюда помещают поля, значения которых будут отображаться на пересечении строк и столбцов сводной таблицы. Для отображения данных указывают формулу – сумму, среднее, количество и т.д.

На рисунке 3.22 выбраны только поля для отчета. Попробуйте поэкспериментировать со строками и столбцами.

№	A	B	C	D	E	F	G
1	Фамилия	Канцтовары	Дата продажи	Сумма		Названия строк	Сумма по полю Сумма
2	Алиева А.	тетради	02.06.2017	5000		Алиева А.	23000
3	Сатыбалдиева К.	ручки	03.06.2017	3000		02.06.2017	5000
4	Петров Л.	дневники	04.06.2017	4000		06.07.2017	8000
5	Алиева А.	тетради	06.07.2017	8000		01.08.2017	10000
6	Сатыбалдиева К.	ручки	07.07.2017	10000		Петров Л.	32000
7	Петров Л.	дневники	08.07.2017	15000		04.06.2017	4000
8	Алиева А.	тетради	01.08.2017	10000		08.07.2017	15000
9	Сатыбалдиева К.	ручки	02.08.2017	18000		03.08.2017	13000
10	Петров Л.	дневники	03.08.2017	13000		Сатыбалдиева К.	31000
11						03.06.2017	3000
12						07.07.2017	10000
13						02.08.2017	18000
14						Общий итог	86000

Рис. 3.22. Создание сводной таблицы

В качестве исходных данных для построения отчета мы взяли небольшую таблицу продаж. Но она могла бы состоять из большого количества строк и столбцов. В электронной таблице можно также быстро посчитать и мгновенно подготовить отчет.



Знание. Понимание



1. Какие инструменты анализа данных вы знаете?
2. Как с помощью гистограмм и диаграмм можно анализировать данные?
3. Для чего применяются спарклайны?
4. Что является альтернативой для спарклайнов?
5. С какой целью применяются сводные таблицы?
6. Какие еще встроенные функции применяются для анализа данных в таблице?
7. Как с помощью встроенных функций можно проанализировать данные?



Применение



8. Сравните данные о доле потребления некоторых видов продуктов детьми и взрослыми (рис. 3.23). Данные в таблице указаны в процентах. На рисунке 3.24 представлена диаграмма.

	А	В	С
1	Продукты питания	Дети	Взрослые
2	Конфеты	73	-25
3	Мороженое	68	-30
4	Мясо	50	-50
5	Картофель	43	-57
6	Сок	80	-20
7	Кумыс	0	-100

Рис. 3.23. Доля потребления некоторых видов продуктов детьми и взрослыми

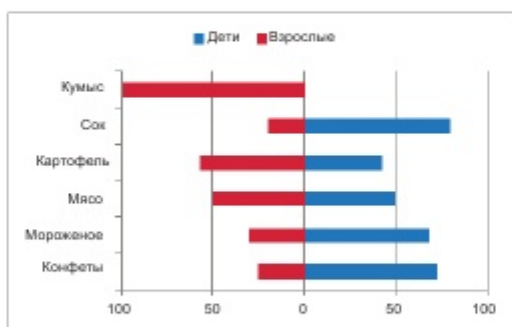


Рис. 3.24. Диаграмма сравнения данных о потреблении продуктов детьми и взрослыми

Шаг за шагом



- 1) Наберите таблицу (рис. 3.23).
- 2) Выберите на вкладке **Вставка – Диаграммы – Линейчатая – Линейчатая с накоплением**.
- 3) Щелкните правой кнопкой мыши по **Области диаграммы**. В контекстном меню выберите **Выбрать данные**.
- 4) Появится диалоговое окно **Выбор источника данных**. Нажмите кнопку **Добавить**. В поле **Имя ряда**: укажите адрес ячейки с названием **Дети**, щелкнув по ней левой кнопкой мыши. В поле **Значения**: укажите диапазон всех значений столбца **Дети**, выделив их. Нажмите кнопку **OK**.
- 5) В окне **Выбор источника данных** нажмите кнопку **Добавить**. Напротив поля **Имя ряда**: щелкните левой кнопкой мыши по ячейке с названием столбца **Взрослые**. Напротив поля **Значения**: укажите диапазон всех значений в столбце **Взрослые**. Нажмите кнопку **OK** (рис. 3.25).

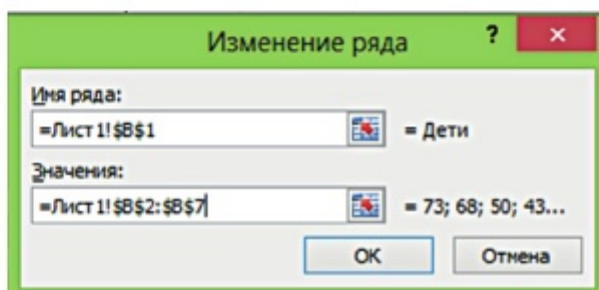


Рис. 3.25. Изменение ряда

- 6) Напротив **Подписи к горизонтальной оси** нажмите кнопку **Изменить**. Выделите названия продуктов в столбце **Продукты питания** и нажмите кнопку **OK**. Нажмите кнопку **OK** в окне **Выбор источников данных**.
- 7) На вкладке **Макет** выполните **Легенда – Добавить легенду сверху**.
- 8) Выделите горизонтальную ось, щелкнув по ней. Чтобы открыть окно настроек, нажмите одновременно клавиши **Ctrl + 1**, появится окно **Формат оси**. В разделе настроек **Параметры оси** для параметра **минимальное значение** выберите вариант **фиксированное** и укажите значение – **100**. Для параметра **Вертикальная ось пересекает** укажите вариант **Значение оси** и введите значение – **100**. Нажмите **Закрыть**.
- 9) Чтобы на горизонтальной оси слева и справа от нуля были положительные числа, зайдите в окно настроек горизонтальной оси. В разделе **Число** для параметра **Числовые форматы** укажите значение **Все форматы**, в поле **Код формата** укажите **0;0** и нажмите **Добавить**. Затем выберите добавленное значение из списка **Тип** и нажмите кнопку **Закрыть**. Вы должны получить диаграмму как на рисунке 3.24.



Анализ



9. На рисунке 3.26 представлены данные о 17 озерах.

	A	B	C
1	Название	Площадь, кв. км	Наибольшая глубина, м
2	Алаколь	2650	54
3	Аральское море	51 000	68
4	Байкал	31 500	1620
5	Балкаш	18 300	26
6	Ильмень	982	10
7	Имандра	876	67
8	Иссык-Куль	6280	702
9	Каспийское море	371 000	1025
10	Ладожское	17 700	230
11	Онежское	9720	127
12	Севан	1360	86
13	Таймыр	4560	26
14	Телецкое	223	325
15	Топозеро	986	56
16	Ханка	4190	11
17	Чаны	1990	9
18	Чудское с Псковским	3550	15

Рис. 3.26. Данные о 17 озерах

Подготовьте лист для получения ответа на вопрос: «Верно ли, что площадь самого большого озера превышает площадь самого маленького озера более чем в n раз?» (Значение n задается в отдельной ячейке.) Постройте гистограмму и проанализируйте данные. Выделите максимальную и минимальную площади озера.



Синтез



10. Для расчета коммунальных затрат на обслуживание жилых домов электроэнергией, водой и газом подготовлен следующий лист (рис. 3.27):

	A	B	C	D	E	F
1	Вид обслуживания		Январь	Февраль	Март	Апрель
2	Электричество	Расход				
3		Тариф				
4	Вода	Расход				
5		Тариф				
6	Газ	Расход				
7		Тариф				

Рис. 3.27. Расчет коммунальных услуг

Расход будет указываться в соответствующих единицах измерения (электроэнергия – кВт/час, вода – литр, газ – куб. м). Тариф – это стоимость единицы расхода в тенге.

Подготовьте лист для определения:

- среднесрочного расхода электроэнергии, воды и газа;
- общих за 4 месяца затрат на каждый из трех видов обслуживания;
- общих затрат на все виды обслуживания за каждый месяц;
- общих затрат на обслуживание домов за 4 месяца.

Сравните данный лист с квитанцией по оплате коммунальных услуг в реальности. Сделайте выводы.

Оценивание



11. Оцените возможности анализа данных в электронной таблице Excel и напишите эссе, проанализировав получившуюся таблицу.



Рекомендации для выполнения домашнего задания

Сформулируйте свое мнение по проведению мгновенного анализа в электронной таблице Excel.



С помощью функций **ПРОПИСН** и **СТРОЧН** можно менять регистр текстовой информации в Excel. Функция **ПРОПИСН** делает все буквы прописными, а **СТРОЧН** – строчными. **ПРОПНАЧ** делает прописной только первую букву в каждом слове.

При внесении информации с нулями в начале, например, число 00785, Excel автоматически превратит его в 785. Для того чтобы оставить нули в начале, поставьте перед числом апостроф «'».

3.4. РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ

ВЫ НАУЧИТЕСЬ

- Использовать встроенные функции для решения задач с применением электронных таблиц;
- использовать различные типы и форматы данных для решения задач в электронных таблицах;
- строить графики функций, заданных в таблице.

КЛЮЧЕВЫЕ СЛОВА

Прикладная задача	Қолданбалы есеп	Application task
Таблица	Кесте	Table



Усиление прикладной направленности курса информатики – это осуществление связи теории с практикой. Поэтому прикладные задачи позволяют осознать сущность будущей профессиональной деятельности.



Прикладная задача – задача, поставленная вне информатики, но решаемая методами использования информационно-коммуникационных технологий. В прикладной задаче прослеживается взаимосвязь других знаний (математики, физики, техники и т.д.) и видов деятельности, связанных с информатикой. Приемы решения таких задач приближены к профессиональной деятельности.



По метеорологическим наблюдениям (измерениям прогнозов погоды за три года) произведите расчеты (см. табл. 3.6–3.9).



Шаг за шагом

1. Заполните формулами пустые ячейки. Для всех данных установлен формат ячеек:
 - числовой формат – числовой;
 - число десятичных знаков – 1.
2. Для расчета количества выпавших осадков, которые меньше 15 мм – «засуха», воспользуйтесь статистической функцией **ЕСЛИСЧЕТ**.
3. Заполните столбец **Прогноз**, применив вложенную функцию **ЕСЛИ()**:
 - **засуха**, если количество осадков < 15 мм;
 - **дождливо**, если количество осадков > 70 мм;
 - **нормально** в остальных случаях.
4. Представьте данные таблицы **Количество осадков** (мм) графически, расположив диаграмму на Листе 2. Выберите тип диаграммы и элементы оформления по своему усмотрению.
5. Переименуйте Лист 1 в **Метео**, Лист 2 – в **Диаграмма**. Удалите лишние листы рабочей книги.
6. Установите ориентацию листа – **альбомная**.
7. Сохраните таблицу под именем **Метео**.

Таблица 3.6

Количество осадков

Название месяца	2018	2019	2020
январь	37,2	38,3	
февраль	11,4	67,8	
март	16,5	20,7	
апрель	19,5	26,8	

Таблица 3.6. Продолжение

Название месяца	2018	2019	2020
май	12,8	45,5	
июнь	130,1	71,8	
июль	57,8	161	
август	57	98,8	
сентябрь	83,5	78,3	
октябрь	87	15,9	
ноябрь	15,4	23	
декабрь	27,5	25,8	

Таблица 3.7

Данные за 2018–2020 годы

Максимальное количество осадков за 3 года (мм)	
Минимальное количество осадков за 3 года (мм)	
Суммарное количество осадков за 3 года (мм)	
Среднемесячное количество осадков за 3 года (мм)	
Количество засушливых месяцев за 3 года	

Таблица 3.8

Данные за 2020 год

№	Название месяца	2020	Прогноз
1	январь	10,7	засуха
2	февраль	2,5	засуха
3	март	14,1	засуха
4	апрель	11,8	засуха
5	май	68,8	нормально
6	июнь	70,7	дождливо
7	июль	100,3	дождливо
8	август	155,8	дождливо
9	сентябрь	78,5	дождливо
10	октябрь	71,3	дождливо
11	ноябрь	53,2	нормально
12	декабрь	13,7	засуха

Таблица 3.9

Данные за 2018–2020 годы

Минимальное количество осадков (мм)	2,5
Суммарное количество осадков (мм)	651,4
Среднемесячное количество осадков (мм)	54,28
Количество засушливых месяцев	5

Следующая задача применяется при анализе данных, когда нужно их отнести к тому или иному относительному диапазону. Такая задача возникает, например, при количественной оценке качества продукции, при распределении премии, при выставлении оценок и т.д.



Представим, что учащиеся на уроках художественного труда изготовили некоторое количество поделок. Нужно оценить их труд в зависимости от объема поделок:



- учащиеся, попавшие в верхний 20%-й участок диапазона изготовленных поделок, получают оценку «отлично»;
- учащиеся, попавшие в последние 10%, получают оценку «удовлетворительно»;
- остальные – оценку «хорошо».



Шаг за шагом

Построим функцию, реализующую цели задания. Выработку числа поделок обозначим буквой X . Иллюстрация техники вычисления $X_{\text{хор}}$ и $X_{\text{отл}}$, являющиеся рубежами диапазонов «хорошо» и «отлично», представлена на рисунке 3.28.

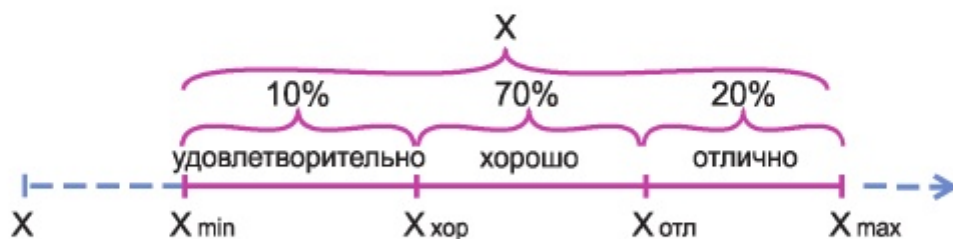


Рис. 3.28. Разбиение диапазона данных

Отсюда функция для ячейки **C2** будет выглядеть так: $C2=ЕСЛИ(B2>>(МАКС(B\$2:B\$10)-МИН(B\$2:B\$10))*90\%; «отлично»;$ $ЕСЛИ(B2>МИН(B2:B10)+20\%*(МАКС(B\$2:B\$10)-МИН(B\$2:B\$10)); «хорошо»;$ «удовлетворительно»)).

The screenshot shows the Microsoft Excel interface. The active window is titled 'Тимур'. The ribbon includes 'Файл', 'Главная', 'Вставка', 'Разметка страницы', 'Формулы', 'Данные', 'Рецензирование', 'Вид', and 'ABBYY FineReader 11'. The 'Данные' ribbon is active, showing options for 'Сортировка' and 'Фильтр'. A dialog box titled 'Сортировка от А до Я' is open, indicating that the selected range is sorted by the 'Имя' column. The table below shows the data being sorted.

	A	B	C	D
1	Имя	Поделки	Оценка	
2	Тимур	25	хорошо	
3	Станислав	30	хорошо	
4	Мурат	35	хорошо	
5	Максим	50	отлично	
6	Костя	45	отлично	
7	Кайрат	20	хорошо	
8	Ертостик	15	удовлетворительно	
9	Артем	40	отлично	
10	Айдар	10	удовлетворительно	

Рис. 3.29. Таблицы оценок

Выполните сортировку по возрастанию по полю **Имя** (рис. 3.29). Что произошло? Поменялся ли список имен? Какой тип диаграммы лучше применить для таблицы?



Знание. Понимание



1. Какие задачи называют прикладными?
2. Приведите примеры прикладных задач.
3. Что такое разбиение диапазона данных?
4. Рассмотрите таблицу расчета заработной платы (табл. 3.10). Заполните в ячейках используемые функции для расчета заработной платы.



Таблица 3.10

Курс доллара:

Расчет заработной платы

№	ФИО	Начислено (тенге)	Налоги		К выдаче в тенге	Эквива- лент в \$
			подход. 5%	пенс. 10%		
1	Майтбасов А.Т.	208 000				
2	Ержанов А.В.	205 000				
3	Закирова Е.Н.	235 000				
	Итого					
	Максимум					
	Минимум					
	Средняя					



Применение



5. Создайте кроссворд, где задание указывается в примечаниях, а правильно вставленная буква приводит к окрашиванию ячейки. Если слово вставлено правильно, то это отмечается каким-либо символом и появляется сообщение.
Примечание. При переходе от одного уровня к другому процесс создания кроссворда будет усложняться. Примените творчество.



Шаг за шагом

- 1) Запустите Excel.
- 2) Переименуйте лист:
 - Откройте контекстное меню, щелкнув правой кнопкой мыши по имени листа. Выберите команду **Переименовать** и введите имя **Кроссворд**.
- 3) Удалите остальные листы:
 - В контекстном меню выберите пункт **Удалить**.
- 4) Введите все слова кроссворда. Каждая буква вводится в отдельную ячейку таблицы.
- 5) Выделите все буквы:
 - Сначала выделите мышкой одно слово, затем, удерживая клавишу **Ctrl**, выделите остальные слова кроссворда.
- 6) Установите границы:
 - Командой **Границы** из группы **Шрифт** вкладки **Главная** выберите из списка границ пункт **Все границы** (рис. 3.30).

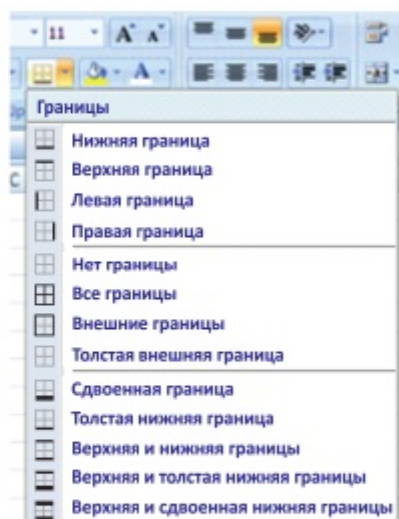


Рис. 3.30. Установка границы кроссворда

7) Сделайте ячейки кроссворда в виде квадратиков. Для этого выделите столбцы, где находятся слова кроссворда:

– Установите курсор на имя столбца **D**. Нажмите на левую кнопку мыши и выделите область до столбца **K**.

8) Поставьте курсор между любыми буквами **D** и **E** (можно взять любые другие буквы выделенного диапазона) так, чтобы появился крестик \leftrightarrow .

9) С нажатой левой кнопкой мыши передвиньте крестик, чтобы уменьшить ширину столбца на 40 пикселей.

10) Увеличьте высоту строк аналогично уменьшению ширины столбцов, чтобы получилась ячейка в виде квадрата.

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3				И	Н	Т	Е	Р	Н	Е	Т
4						Е					
5						Х					
6						Н					
7				Ш	К	О	Л	А			
8						Л					
9						О					
10						Г					
11						И					
12						Я					

Рис. 3.31. Отформатированный кроссворд

11) Выровняйте буквы по центру ячеек:

- Выделите весь кроссворд.

- Выполните команду **Выровнять по центру**  командой из группы

Выравнивание вкладки **Главная** (рис. 3.31).

12) Вставьте придуманные задания в примечания:

- Активируйте ячейку с буквой **И** (**D3**).

- В контекстном меню ячейки выберите пункт

Вставить примечание (рис. 3.32).

- В появившемся окне с красным треугольником замените текст на задание (например, введите определение Интернета – это *Глобальная компьютерная сеть*).

- Так же вставьте примечания в ячейки с буква-

ми **Ш** (**D7**) и **Т** (**F3**) – *Организация образования*.

13) Задайте условное форматирование ячейке для ввода нужной буквы.

- Выделите букву **И** (**D3**).

- Выполните команду **Условное форматирование – Правила выделения ячеек – Равно** из группы **Стили** вкладки **Главная** (рис. 3.33).

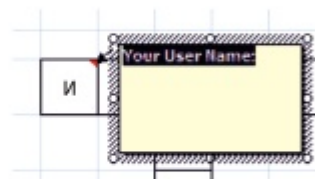


Рис. 3.32. Вставить примечание

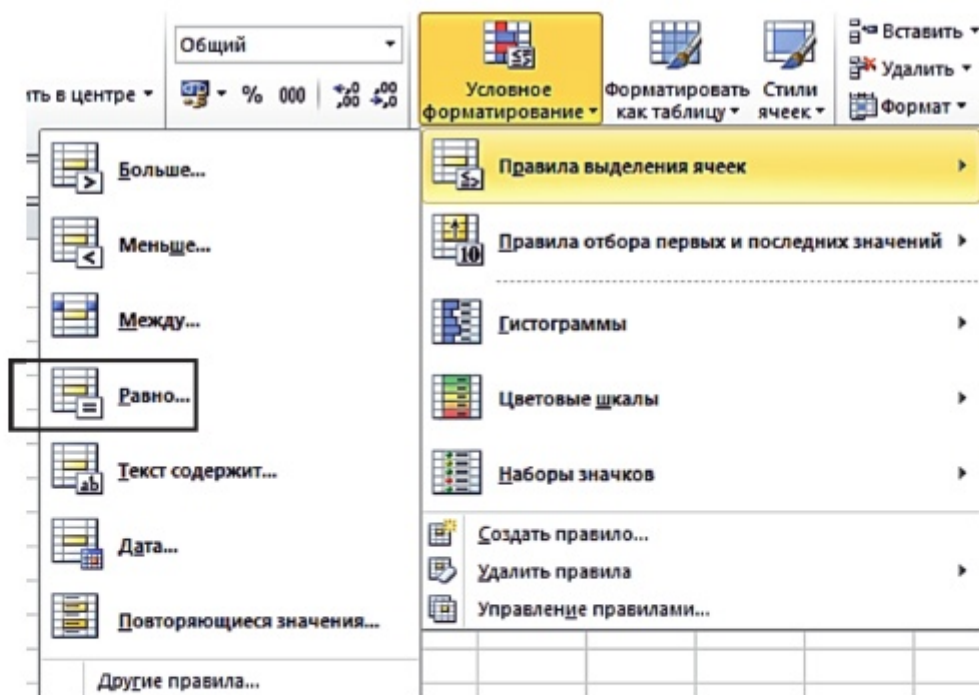


Рис. 3.33. Условное форматирование

- В появившемся окне **Равно** (рис. 3.34) вставьте букву **И**, затем выберите формат и нажмите **ОК**.
- То же сделайте с остальными буквами кроссворда.

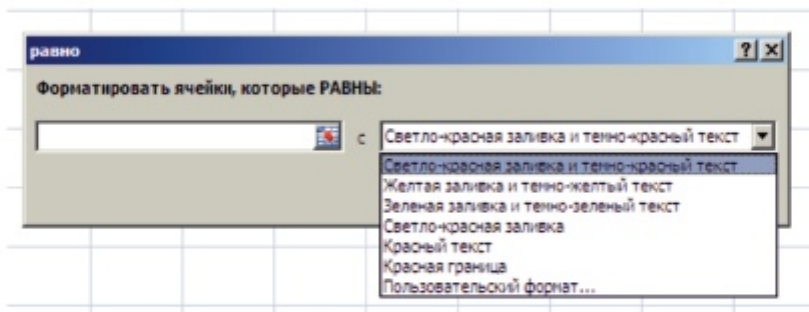



Рис. 3.34. Условное форматирование **Равно**

- 14) В ячейки слева или выше слова поставьте какой-нибудь символ, который будет отображаться при правильном наборе слова.
- Например, поставьте курсор в ячейку **М3**. Выберите функцию **ЕСЛИ** на вкладке **Формулы**, нажав на команду **Вставить функцию**  (или **SHIFT + F3**).
 - В появившемся списке выберите **Другие функции...**
 - В окне **Мастер функций** выберите из категории **Логические** функцию **ЕСЛИ**.
 - В окне **Аргументы функции** (рис. 3.35) поставьте курсор в область **Значение_если_истина** и наберите любой символ, например, **g**. Данный символ будет выводиться, если ответ верный.

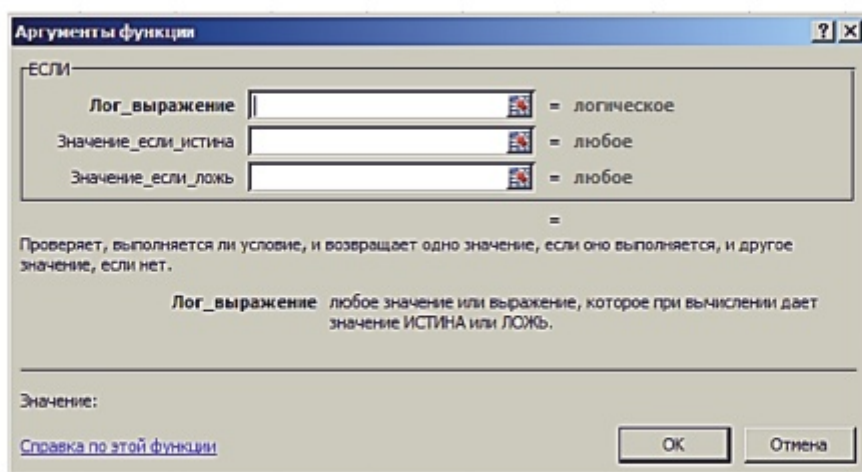


Рис. 3.35. Аргументы функции

- Поставьте курсор в область **Значение_если_ложь** и наберите кавычки «».
- Поставьте курсор в область **Лог_выражение** и выберите функцию **И** (рис. 3.36).

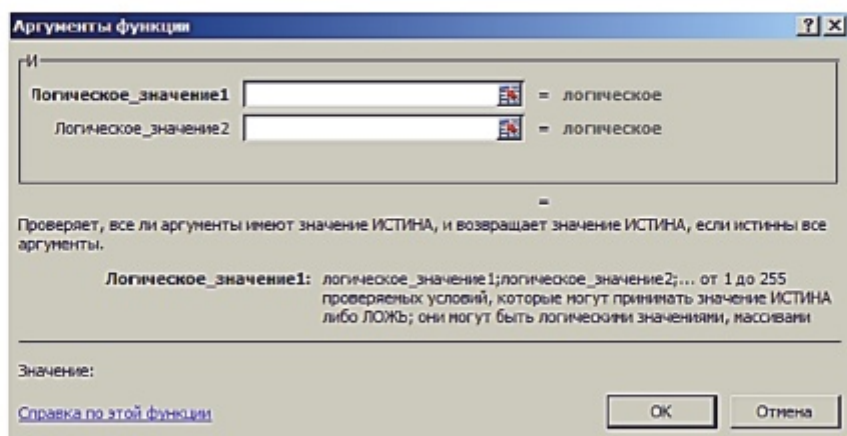


Рис. 3.36. Аргументы функции

- Слева в командной строке около функции **ЕСЛИ** откройте список, нажав на черную стрелку в столбце **М**, выберите **Другие функции**.
- В окне **Мастер функций** выберите логическую функцию **И**.
- Поставьте курсор в область **Логическое_значение1**, выделите ячейку с буквой **И**. В области появится адрес ячейки, допишите «=И».
- Перенесите курсор в следующую область **Логическое_значение2**, выделите ячейку с буквой **Н**. В области появится адрес ячейки, допишите «=И» и т.д. (рис. 3.37).

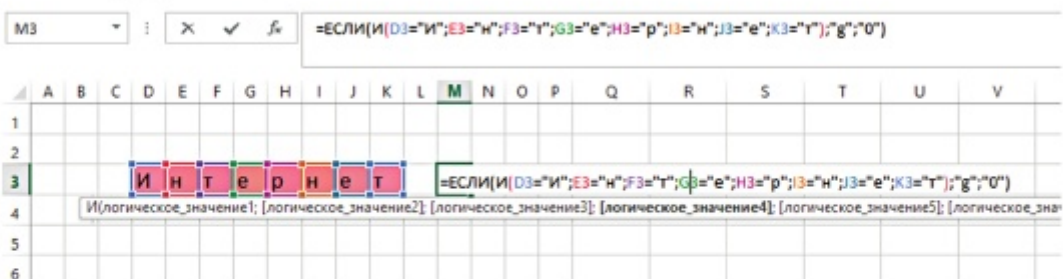


Рис. 3.37. Ввод формулы в ячейке М3

Рекомендации для выполнения задания

Составьте кроссворд в бумажном варианте. Например, в замечание введите вопрос кроссворда: 1) *Организация образования* – школа; 2) *Глобальная компьютерная сеть* – Интернет и т.д. Количество вопросов в кроссворде – от 5 до 7.

Анализ



6. Проанализируйте данные в кроссворде. Запишите для них функцию **ЕСЛИ**.

– В ячейках с символом измените шрифт – **Symbol**. В формуле будет отображаться символ **g**, в ячейке – **γ**.



ЕСЛИ(И(D3=«И»;E3=«н»;F3=«т»;G3=«е»;H3=р»;I3=«н»;J3=«е»;K3=«т»);«g»;«0»)

15) Выведите информацию о работе над кроссвордом (рис. 3.38).

=ЕСЛИ(И(M3="g";M7="g";F14="g");"Молодец, все верно!";"Решите кроссворд.")															
D	E	F	G	H	I	J	K	L	M	N	O	P	Q		
И	н	т	е	р	н	е	т		γ						
	е														
	х														
	н														
ш	к	о	л	а					γ						
	л														
	о														
	г														
	и														
	я														
	γ														

Молодец, все верно!

=ЕСЛИ(И(O7="ш";E7="х";F7="о";G7="л";H7="а");"g";"0")															
D	E	F	G	H	I	J	K	L	M	N	O	P			
И	н	т	е	р	н	е	т		γ						
	е														
	х														
	н														
ш	к	о	л	а					0						
	л														
	о														
	г														
	и														
	я														
	γ														

Верный ответ

Неверный ответ

Решите кроссворд.

Рис. 3.38. Результаты решения кроссворда



Синтез



7. Для очистки кроссворда создайте макрос. Макрос – автоматизация повторяющихся задач.

- Выделите область ячеек **M11:S14**.
- Введите формулу аналогично описанному выше.
- =ЕСЛИ(И(M3=«г»;M7=«г»;F14=«г»);«Молодец, все верно!»;«Решите кроссворд.»)
- Выполните условное форматирование области ячеек **M11:S14**.

16) Уберите сетку. Для этого нужно убрать галочку с **Сетка** в группе **Показать или скрыть** на вкладке **Вид**.

17) Очистите кроссворд (рис. 3.39):

- Выделите область ячеек **A1:R14**.
- Нажмите на клавишу **Delete**.



Рис. 3.39. Очистка кроссворда

18) Создайте кнопку для автоматической очистки кроссворда:

- Создайте **Макрос** для очистки кроссворда. Выполните команду **Макросы – Запись макроса** из группы **Макросы** вкладки **Вид** (рис. 3.40).

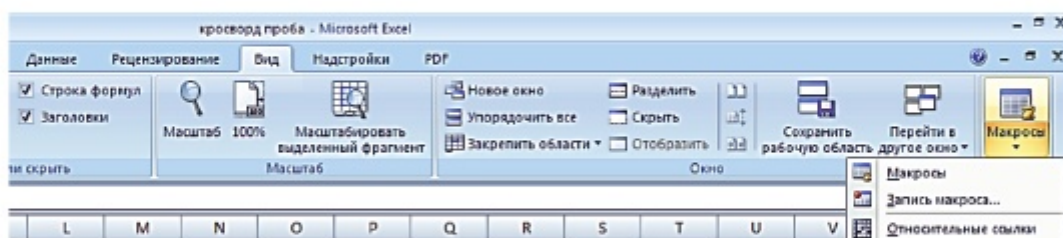


Рис. 3.40. Окно Макросы

- В появившемся окне **Запись макроса** впишите **Имя макроса – Очистка**, нажмите **OK** (рис. 3.41).

- На листе **Кроссворд** выделите область кроссворда (D3:K12) и клавишей **Delete** очистите область.
- Выполните команду **Макросы – Остановить запись** из группы **Макросы** вкладки **Вид**.

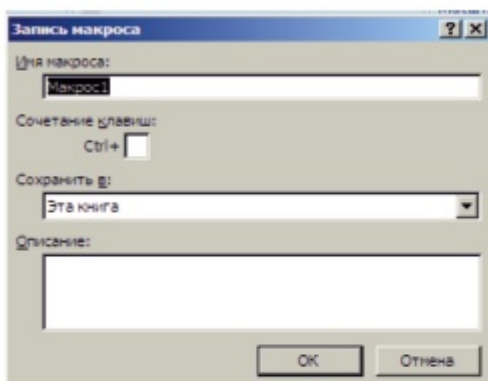


Рис. 3.41. Окно записи макроса

Оценивание



8. Как нарисовать кнопку на листе, чтобы очистить кроссворд?

- 19) Добавьте на **Ленту** вкладку **Разработчик**. Меню **Файл**, пункт **Параметры – Дополнительно**. Поставьте галочку **Разработчик**. Нажмите кнопку **OK** (рис. 3.42).

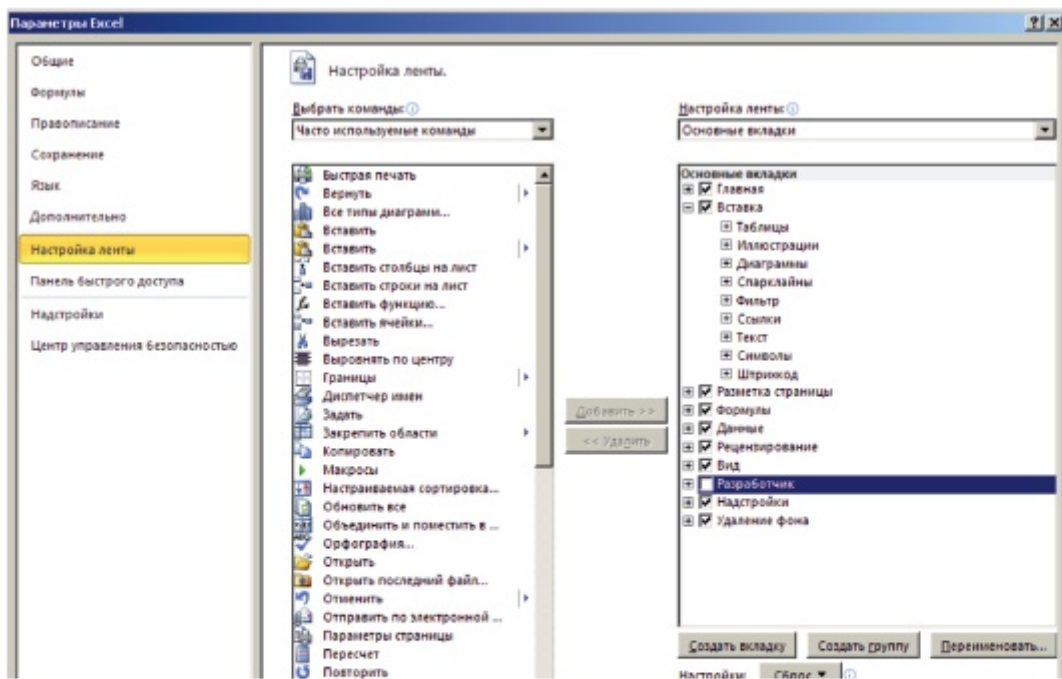


Рис. 3.42. Параметры Excel

- 20) Нарисуйте кнопку для очистки кроссворда:
- Щелкните по **Кнопке** (элемент управления формы), вкладка **Разработчик**, группа **Элемент управления**, команда **Вставить** (рис. 3.43).
 - Переведите курсор на лист и нажмите на левую кнопку мыши. Нарисуйте кнопку.
 - В появившемся окне **Назначить макрос объекту** выберите нужный **Макрос – Очистка – ОК** (рис. 3.44).
 - Выделите кнопку и измените надпись на ней.

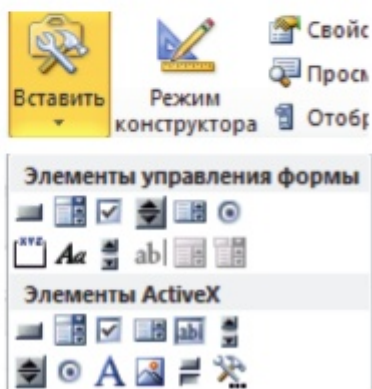


Рис. 3.43. Элементы управления

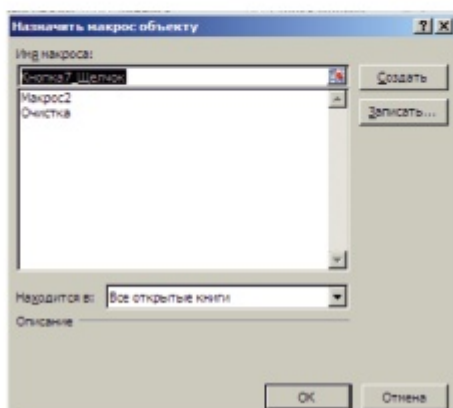


Рис. 3.44. Назначение макроса

- 21) Сохраните документ: команда **Сохранить как...** из меню **Файл**.
 22) Рассмотрите пример готового кроссворда (рис. 3.45).



Рис. 3.45. Готовый кроссворд

- 23) Оцените кроссворд.
 24) Рекомендуйте кроссворд своим одноклассникам.

Инструмент Камера или как перевернуть текст в ячейке?



Инструмент **Камера** по умолчанию недоступен во всех версиях Excel. Рассмотрим алгоритм вывода **Камеры** на **Панель быстрого доступа** в электронной таблице Excel (версии после 2003).

Выберите **Настройка панели быстрого доступа** из команды **Все команды – Камера – Добавить – ОК**.

Выделите ячейку с данными, которую необходимо скопировать.левой кнопкой мыши кликните на значок инструмента **Камера**. Появится окантовка выделенной ячейки, то есть ячейка станет активной как при обычном копировании.

левой кнопкой мыши укажите место на листе, куда необходимо вставить снимок. Снимок автоматически вставляется в ячейку. Переверните его на 180 градусов (рис. 3.46).



Рис. 3.46. Поворот рисунка

С получившимся снимком можно работать как с обычным рисунком, внедренным на лист. Можно изменять его размеры, свойства, форму, устанавливать прозрачность, рамки и т.д.

Поэкспериментируйте. Можно ли одновременно «фотографировать» несколько ячеек/диапазонов или несмежные диапазоны?

Задание для проектной работы

Проект 3

Создайте таблицу по теме «Статистические данные по областям Республики Казахстан» и проанализируйте ее. На основе статистических данных составьте краткий отчет. (Отчет – это сообщение или документ, который предоставляет информацию и описывает суть исследования применительно к определенной ситуации. Отчет основывается на привлечении документальных данных.)

Требования к оформлению проекта

1. Используйте для обработки информации электронную таблицу Microsoft Excel.
2. Создайте таблицу, в которой отобразите статистические данные о численности населения, проживающего в областях Казахстана (казахи, русские и другие национальности).
3. Вставьте соответствующие рисунки людей в национальных костюмах.
4. Задайте таблице различное оформление.
5. Отобразите данные на различных типах диаграмм. Задайте диаграмме заголовки, названия осей, подписи данных.
6. Добавьте краткий отчет.

Дополнительные задания к третьему разделу

1. Решите в электронной таблице Excel квадратные уравнения: $y = x^2$; $y = x^2 + 1$; $y = x^2 + x + 1$ (рис. 3.47). При этом значения x меняются в интервале от -5 до 5 . Постройте в одной координатной плоскости три графика функции.

x	$y = x^2$	$y = x^2 + 1$	$y = x^2 + x + 1$
-5	25	26	21
-4	16	17	13
-3	9	10	7
-2	4	5	3
-1	1	2	1
0	0	1	1
1	1	2	3
2	4	5	7
3	9	10	13
4	16	17	21
5	25	26	31

Рис. 3.47. Квадратные уравнения

2. Числа Фибоначчи – это ряд чисел, в котором каждое последующее число равно сумме двух предыдущих: 1, 1, 2, 3, 5, 8, 13 и т.д.

Формула:

$$F_0 = 1$$

$$F_1 = 1$$

$$F_2 = F_1 + F_0 = 1 + 1 = 2$$

$$F_3 = F_2 + F_1 = 2 + 1 = 3$$

$$F_4 = F_3 + F_2 = 3 + 2 = 5$$

$$F_5 = F_4 + F_3 = 5 + 3 = 8$$

.....

$$F_n = F_{n-1} + F_{n-2}$$

В электронной таблице Excel проведите расчет десяти чисел Фибоначчи. Постройте два типа диаграмм: гистограмму и график с маркерами. Сравните их. Какой тип диаграммы лучше отражает последовательность чисел Фибоначчи?

3. Составьте в электронной таблице Excel журнал класса по информатике за одну четверть, в котором выведены баллы от 1 до 10. Диапазон баллов дан в десятичной системе оценивания: если количество баллов меньше 5, то выводится оценка «2», от 5–6 баллов – оценка «3», от 7–8 – оценка «4», от 9–10 – оценка «5». Для вывода оценок используйте вложенную функцию **ЕСЛИ**. Установите для данных: **формат данных – числовой формат – общий**.

4. На стыке двух континентов Европы и Азии расположена Республика Казахстан. Казахстан занимает площадь, равную 2724,9 тысячам квадратных километров (тыс. км²). Протяженность его территории с запада на восток превышает 3000 км, с юга на север – 1700 км. Казахстан граничит на севере с Россией – 6467 км, на востоке с Китаем – 1460 км, на юге с Кыргызстаном – 980 км, Узбекистаном – 2300 км и Туркменистаном – 380 км, на западе омывается Каспийским морем. Общая протяженность границ – 12 187 км. Заполните представленные данные (взяты с сайта Комитета индустрии туризма Министерства туризма и спорта Республики Казахстан) в электронной таблице Excel. Посчитайте границы Казахстана по Каспию в километрах. Какую диаграмму по этим данным лучше построить?

Тестовые задания к разделу III

Вариант 1

1. Табличные процессоры – это:

- A) прикладные программы, предназначенные для работы с электронными таблицами;
 B) прикладные программы, предназначенные для работы с базами данных;
 C) прикладные программы, предназначенные для работы с текстом;
 D) таблицы в Word.

2. В ячейке D7 записана формула (C3+C5)/D6. Как она изменится при переносе этой формулы в ячейку E8?

- A) (C3+C5)/D6.
 B) (C3+C5)/E6.
 C) (D4+D6)/E7.
 D) (C4+C6)/D7.

3. Следствием принципа абсолютной адресации является правило:

- A) при перемещении формулы в другую ячейку адреса ячеек не изменяются;
 B) при перемещении формулы в другую ячейку изменяется только номер строки в адресах ячеек;
 C) при перемещении формулы в другую ячейку изменяются адреса ячеек (номер строки и буква столбца);
 D) при перемещении формулы в другую ячейку изменяется только имя столбца (буква) в адресах ячеек.

4. Дана электронная таблица:

	A	B	C	D
1	5	3	4	
2	6	1	5	=A1+\$B\$2*C\$1
3	8	3	-4	

В ячейку D2 введена формула (A1+*\$B\$2*C\$1) и перемещена в ячейку D3. Какое значение появится в ячейке D3?

- A) 9.
 B) 21.
 C) 10.
 D) 20.

5. Фрагмент или блок таблицы – это:

- A) часть строки;
- B) часть столбца;
- C) любая ее прямоугольная часть;
- D) ячейка.

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

- A) чертежом;
- B) картой;
- C) блок-схемой;
- D) диаграммой.

7. График функции можно создать в Excel с помощью:

- A) Строки формул;
- B) Мастера Функций;
- C) Мастера Шаблонов;
- D) Мастера Диаграмм.

8. Диаграмма – это:

- A) карта местности;
- B) форма графического представления числовых значений, которая позволяет облегчить интерпретацию числовых данных;
- C) красиво оформленная таблица;
- D) обычный график.

9. Круговая диаграмма – это диаграмма:

- A) которая представлена в виде круга, разбитого на секторы, и в которой допускается только один ряд данных;
- B) в которой отдельные ряды данных представлены в виде закрашенных разными цветами областей;
- C) в которой используется система координат с тремя координатными осями, что позволяет получить эффект пространственного представления рядов данных;
- D) в которой отдельные значения представлены точками в декартовой системе координат.

10. На основе чего строится любая диаграмма?

- A) Книга Excel.
- B) Графический файл.

- C) Текстовый файл.
- D) Данные таблицы.

11. При построении диаграммы она получилась пустой. Почему это произошло?

- A) Не хватило оперативной памяти. Следует закрыть ненужную программу.
- B) Выбран неверный путь диаграммы.
- C) Не выделен блок ячеек с данными.
- D) Диаграмма не помещается на экране, так как слишком большая.

12. Абсолютные ссылки – это такие ссылки, которые при копировании формулы из одной ячейки в другую:

- A) меняются, если только эти ячейки расположены рядом;
- B) не меняются;
- C) меняются, причем ссылки в новой формуле имеют такое же взаимное расположение ячеек, участвующих в формуле, как и в формуле-оригинале;
- D) не копируются.

13. В ячейке F6 записана формула =МИН(B2:D5). Эту формулу скопировали в ячейку G6. Укажите формулу, которая запишется в ячейку G6.

- A) =МИН(C2:D5).
- B) =МИН(C2:E5).
- C) =МИН(E2:F5).
- D) =МИН(B2:D5).

14. В ячейках записаны числа: в A1 – 5, в A2 – 5, в A3 – 5, в A4 – 5, в B1 – 100, в B2 – 1000, в B3 – 10000, в C1 – 5000. В ячейке A6 записана формула =МИН(A1:B2)+\$C\$1. Какой результат в ней будет?

- A) 5120.
- B) 1120.
- C) 5005.
- D) 5000.

15. В ячейках записаны числа: в A1 – 5, в A2 – 5, в A3 – 5, в A4 – 5, в B1 – 100, в B2 – 1000, в B3 – 10000. В ячейке A6 записана формула =СУММ(A1:B2). Какой результат в ней будет?

- A) 1020.
- B) 1110.

- C) 1120.
D) 1010.

16. Круговая диаграмма построена для трех ячеек с числами 30, 30, 40. На ней возле секторов указаны значения 30%, 30%, 40%. Какие значения были бы указаны у секторов, если бы в этих ячейках содержались числа 60, 60, 80?

- A) 60%, 60%, 80%.
B) 30%, 30%, 40%.
C) 25%, 25%, 50%.
D) значений не будет, так как диаграмма не будет построена, поскольку $60 + 60 + 80 > 100$.

17. Какой отчет называется сводной таблицей?

- A) Сводная таблица – это отчет, построенный на основе символьных данных.
B) Сводная таблица – это отчет, построенный на основе табличных данных.
C) Сводная таблица – это отчет, построенный на основе формул.
D) Сводная таблица – это отчет, построенный на основе статистических функций.

18. Табулирование функции – это:

- A) вычисление значений функции от некоторого начального значения до некоторого конечного значения;
B) вычисление значений функции при изменении аргумента;
C) вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения;
D) вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения с определенным шагом.

Вариант 2

1. В электронной таблице выделили группу из 4 ячеек. Какие это могут быть ячейки?

- A) A1:B4.
B) A1:C2.
C) A1:B2.
D) A2:C4.

2. Дана электронная таблица:

	A	B	C	D
1	5	2	4	
2	10	1	6	=A2*B1+C1

В ячейку D2 введена формула: (A2*B1+C1). Какое значение появится в ячейке D2?

- A) 6.
- B) 14.
- C) 16.
- D) 24.

3. В ячейке B1 записана формула =2*\$A1. Какой вид приобретет формула, после того как ячейку B1 скопируют в ячейку C2?

- A) =2*\$B1.
- B) =2*\$A2.
- C) =3*\$A2.
- D) =3*\$B2.

4. Деловая графика представляет собой:

- A) совокупность графиков функций;
- B) графические иллюстрации;
- C) график совещания;
- D) совокупность программных средств, позволяющих представить в графическом виде закономерности изменения числовых данных.

5. Гистограмма наиболее пригодна для:

- A) отображения вклада каждого значения в общую сумму;
- B) отображения динамики изменения данных;
- C) сравнения различных членов группы;
- D) отображения удельных соотношений различных признаков.

6. Диаграмма в электронных таблицах – это:

- A) качественно оформленная числовая таблица;
- B) график, отображающий зависимость между всеми числами таблицы;
- C) средство наглядного графического изображения информации, предназначенное для сравнения одной или нескольких величин или нескольких значений одной величины;
- D) зависимость между числовыми значениями.

7. В каком разделе меню табличного процессора Excel находится Мастер Диаграмм?

- A) Правка.
- B) Вставка.
- C) Сервис.
- D) Данные.

8. Гистограмма – это диаграмма, в которой:

- A) отдельные значения представлены полосами различной длины, расположенными горизонтально вдоль оси OX;
- B) отдельные значения представлены вертикальными столбцами различной высоты;
- C) используется система координат с тремя координатными осями;
- D) используются параллелепипеды, размещенные вдоль оси OX.

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

- A) линейчатой;
- B) точечной;
- C) круговой;
- D) гистограммой.

10. Линейчатая диаграмма – это диаграмма:

- A) в которой отдельные значения представлены вертикальными столбцами различной высоты;
- B) в которой отдельные значения представлены точками в декартовой системе координат;
- C) в которой отдельные значения представлены полосами различной длины, расположенными горизонтально вдоль оси OX;
- D) которая представлена в виде круга, разбитого на секторы, и в которой допускается только один ряд данных.

11. Чтобы отформатировать какой-либо элемент диаграммы, необходимо:

- A) дважды щелкнуть по нему курсором мыши;
- B) выполнить команды **Диаграмма, Исходные данные**;
- C) выполнить команды **Диаграмма, Добавить данные**;
- D) все вышеперечисленные ответы верны.

12. Укажите неверное высказывание относительно следующей фразы. Вместо формулы $=A2+A3+A4+A5+A6+A7+A8+A9+A10$ предпочтительнее пользоваться формулой $=СУММ(A2:A11)$.

- A) Вторая формула более удобна для чтения и записи.
- B) При вставке новой ячейки в диапазон A2:A10 формула будет откорректирована автоматически и получится $= СУММ(A2:A11)$.
- C) Сумма может быть подсчитана только по первой формуле.
- D) Первую формулу можно написать мышкой.

13. Относительные ссылки – это такие ссылки, которые при копировании формулы из одной ячейки в другую:

- A) меняются, если только эти ячейки расположены рядом;
- B) не меняются;
- C) меняются, причем ссылки в новой формуле имеют такое же взаимное расположение ячеек;
- D) не копируются.

14. В ячейке E8 записана формула $=МИН(B2:D5)+\$A\2 . Эту формулу скопировали в ячейку F8. Укажите формулу, которая запишется в ячейку F8.

- A) $=МИН(C2:D5)$.
- B) $=МИН(C2:E5)+\$A\2 .
- C) $=МИН(E2:F5)+\$B\2 .
- D) $=МИН(B2:D5)+\$A\2 .

15. В ячейках записаны числа: в A1 – 5, в A2 – 5, в A3 – 5, в B1 – 100, в B2 – 1000, в C1 – 0, в C2 – 0, в D1 – 0. В ячейке A6 записана формула $=МИН(A1:B2)+\$D\1 . Эту формулу скопировали в ячейку B6. Какой результат в ней будет?

- A) 5005.
- B) 0.
- C) 6000.
- D) 5100.

16. Круговая диаграмма построена для трех ячеек с числами 40, 60, 100. Какие значения будут указаны у секторов на диаграмме?

- A) 40%, 60%, 100%.
- B) 30%, 30%, 40%.

- С) 20%, 30%, 50%.
- Д) значений не будет, так как диаграмма не будет построена, поскольку $40 + 60 + 100 > 100$.
- Е) 20%, 20%, 60%.

17. Какие функции в Excel называются встроенными?

- А) Это функции, которые предназначены для расчетов, вычислений и анализа данных.
- В) Это функции, которые способны вернуть значение.
- С) Это функции, которые способны вернуть значение в зависимости от переданного им параметра и предназначены для расчетов, вычислений и анализа данных.
- Д) Это функции, которые не способны вернуть значение в зависимости от переданного им параметра и предназначены для расчетов, вычислений и анализа данных.

18. Прикладная задача – задача, поставленная вне информатики, но решаемая методами использования:

- А) математики;
- В) ИКТ;
- С) анализа данных;
- Д) физики.

РАЗДЕЛ
IV**ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ
НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON****4.1. ЦИКЛ WHILE****ВЫ НАУЧИТЕСЬ**

Использовать оператор цикла `while`.

КЛЮЧЕВЫЕ СЛОВА

Цикл	Цикл	Loop
Циклический процесс	Циклдік процесс	Loop process
Тело цикла	Цикл денесі	Loop body
Параметр цикла	Цикл параметрі	Loop parameter

В обычной жизни очень часто приходится выполнять одни и те же действия несколько раз. Например, при подготовке к соревнованию по бегу спортсмен многократно бежит по дорожке, пока не улучшит результат. Или цикл уроков, который повторяется в течение недели, четверти до тех пор, пока не завершится учебный год. Многие посещали либо слышали о противоселевой плотине «Медеу», к которой ведет бетонная лестница, состоящая из 842 ступеней. В этом случае, пока не преодолеете все ступени, вы не сможете добраться на самый верхний уровень плотины.



Каким образом многократное повторение одних и тех действий можно записать в программном коде?

Важнейшим достоинством языков программирования является применение алгоритмов ветвления или циклов. Основной смысл составления программ с циклом состоит в том, что команда, написанная один раз, будет выполняться многократно. Поэтому хорошим стилем программирования будет являться выбор программистом подходящей конструкции цикла. В итоге программа, выполняя весь циклический процесс, должна быть самой короткой, понятной и эффективной.



Цикл – это любая многократно исполняемая последовательность инструкций, пока действует заданное условие.

Циклический процесс – это вычислительный процесс, предусматривающий многократное повторение одного и того же действия или одних и тех же операций при различных исходных значениях.

Программы, реализующие циклический процесс, называются **циклическими программами**. Для реализации циклических программ в Python применяется цикл **while**.

Чаще цикл **while** применяют тогда, когда неизвестно изначально, сколько будет повторений.

Синтаксис цикла while

```
while <условие>:
инструкции тела цикла

while <a логический оператор b>
действие (-я) } Инструкции тела
изменение a   } цикла
```



Рис. 4.1. Блок-схема цикла **while**

Синтаксис цикла **while** показан на рисунке 4.1. Условие записывается после слова **while**. Оно может быть произвольным арифметическим выражением, в котором должен быть хотя бы один из операторов $=$, $!=$, $<$, $>$, $<=$, $>=$. Также в нем могут использоваться логические операторы **and**, **or**, **not**. После условия ставится **двоеточие**.



Параметр цикла – это переменная, которая принимает новые значения при каждом повторении цикла.

Тело цикла состоит из одной или нескольких инструкций, записанных с отступом одинаковой величины.

Условие определяет, будет ли выполняться тело цикла. Если проверяемое условие принимает **истинное** значение, то выполняется тело цикла до тех пор, пока условие не станет **ложным**. При ложном значении условия происходит выход из цикла. Если условие будет ложным при первой проверке, то тело цикла может ни разу не выполниться. Рассмотрим простой пример.



Напишите программу вывода всех четных чисел от 2 до 20.



Шаг за шагом

1. Присвоим параметру цикла **i=2**.
2. Для проверки условия запишем **while i<=20**:
Если параметр цикла **i меньше** или **равен 20**, то есть **условие истинно**, то выполняется тело цикла команда **print(i)** – происходит вывод на печать четного числа с новой строки.
3. Затем происходит наращивание параметра цикла на число 2: **i+=2**.
4. Как только условие становится **ложным**, происходит выход из цикла: в этом случае число будет больше **20**.
5. Код программы показан на рисунке 4.2.

```
File Edit Format Run ----- RESTART:
i=2
while i<=20:
    print(i)
    i+=2 #i=i+2
2
4
6
8
10
12
14
16
18
20
>>> |
```

Рис. 4.2. Код программы «Четные числа»

Если из кода программы удалить оператор увеличения значения **i**, то получится **бесконечный цикл**, в котором условие выхода цикла никогда не выполнится. Бесконечные циклы – логические ошибки, которые необходимо избегать. Для прерывания работы программы с бесконечным циклом используется комбинация клавиш **Ctrl + C**.

Часто в цикле **while** используют не одно условие, а несколько. Пример кода такой программы можно увидеть на рисунке 4.3.

```

File Edit Format Run Options >>>
x=7
y=15
while x < 10 and y < 20:
    x = x + 1
    y = y + 1
    print(x, y)
>>> ----- RESTART:
8 16
9 17
10 18
>>> |

```

Рис. 4.3. Составное условие

В данном примере оператор цикла **while** проверяет два условия:

Первое условие: «Верно ли, что **x** меньше 10?»

Второе условие: «Верно ли, что **y** меньше 20?»

Проверка происходит до тех пор, пока оба условия **истинны**. При этом значение каждой из двух переменных **x** и **y** увеличивается на **1**, затем выводится на экран.

Цикл будет выполняться до тех пор, пока **x < 10**, а **y < 20**. Будет выполнено 3 таких повтора, так значение **x** достигнет **10**. Так как первое условие **x < 10** стало **ложным**, то программа завершит цикл.

Знание. Понимание



1. Что такое цикл?
2. В каких случаях чаще всего используется цикл **while**?
3. До каких пор будет выполняться цикл **while**?
4. В каком случае в цикле **while** тело цикла может ни разу не выполниться?
5. Почему цикл может быть бесконечным?
6. Напишите верный синтаксис цикла с **while**.
7. Произнесите ключевые слова на разных языках.



Применение



8. Дана программа. Исправьте ошибки и напишите, что получится в результате ее выполнения.

```

i = 0
while i < 5
print (i)
i = i + 1

```

Результат: _____



9. Напишите программу, выводящую ряд четных чисел от **0** до **20**.



Анализ

10. Дан код программы, проанализируйте его. Напишите условие задачи. Какой будет результат выполнения программы?

```

fib1 = 0
fib2 = 1
print (fib1)
print (fib2)
n = 10
i = 0
while i < n:
    fib_sum = fib1 + fib2
    print (fib_sum)
    fib1 = fib2
    fib2 = fib_sum
    i = i + 1

```



Синтез



11. Напишите программу, которая выведет на экран все натуральные делители данного натурального числа **n**.

Рекомендации. Для решения задачи используйте цикл, в котором переменная **i** будет меняться от **1** до **n**. Внутри цикла используйте ветвление с проверкой условия, если остаток от деления **n** на **i** равен **0**, то напечатайте значение **i**.

12. Оцените, в каких задачах может использоваться цикл **while**. Придумайте задачу, в которой будет использоваться цикл **while**. Напишите код программы на языке программирования Python. Поделитесь своей задачей с одноклассниками.



Вложенные циклы while

Инструкция одного цикла **while** может находиться внутри инструкции другого цикла. Такая структура называется **структурой вложенных циклов**. Работа такой структуры совершается следующим образом: на одну итерацию внешнего цикла вложенный цикл совершает все возможные итерации.

Блок-схема вложенных циклов **while** показана на рисунке 4.4.



Рис. 4.4. Блок-схема вложенных циклов **while**

Пример. Дано целое число n . Разложите его на простые множители (рис. 4.5).

File	Edit	Format	Run	Options
<pre> n = int(input('n = ')) i = 2 while i < n // 2: while not n % i: print(i, end=' ') n //= i i += 1 </pre>				
<pre> n = 27 3 3 3 n = 1000 2 2 2 5 5 5 </pre>				

Рис. 4.5. Программа разложения числа на множители

Идея алгоритма в следующем. Во внешнем цикле перебираем возможные делители числа. Во внутреннем цикле мы делим число на делитель, пока это возможно.

4.2. ЦИКЛ FOR



ВЫ НАУЧИТЕСЬ

Использовать оператор цикла `for`.

КЛЮЧЕВЫЕ СЛОВА

Для	Үшін	For
Итерация	Итерация	Iteration
Диапазон	Диапазон	Range
Параметр цикла	Цикл параметрі	Loop parameter



Как вам известно, цикл с условием **while** используется при неизвестном количестве повторов. Какой цикл лучше применить, если количество повторов будет известно заранее? Есть ли другой способ организации повторения группы выражений?

Допустим, необходимо выполнить часто повторяющиеся действия с известным количеством повторов. Например, сделать два круга вокруг футбольного поля либо пройти 842 ступени для подъема на плотину «Медеу», просуммировать 100 чисел с шагом 5 и т.д. Для описания всех этих действий необходимо большое количество времени. Как сэкономить время, написав короче программу и улучшив ее структуру?

Для повторения какого-либо действия цикла (итераций) на заданное **n** число раз можно использовать цикл **for** вместе с функцией **range**. Цикл **for** иногда называют **циклом с параметром**.



Цикл **for** используется для повторения какой-либо последовательности действий заданное число раз или при изменении значения переменной в цикле от некоторого начального до конечного значений.

Синтаксис цикла **for** выглядит так:

```
for i in range(n):
    тело цикла
```

где **i** – **индексная переменная**, или **параметр цикла**.

В цикле **for** указываются переменная и множество значений. Переменная **i** будет последовательно принимать эти значения. В качестве параметра **n** функции **range** используются **числовая константа**, **переменная** или **произвольное арифметическое выражение**.

Множество значений может быть задано диапазоном, списком, кортежем или строкой. Мы будем применять диапазон, который считывает количество повторений, начиная с первого значения, равного нулю (0), и останавливается на предпоследнем.

Рассмотрим три варианта задания диапазона функции **range**.

1) Если указан один параметр, то индексная переменная **i** принимает начальное значение равно **нулю (0)**.

2) Если указаны два параметра, то индексная переменная **i** будет принимать значения:

- первый параметр – начальное значение;
- второй параметр – значение, находящееся на предпоследней позиции и равно **n**;
- последнее значение будет находиться в позиции **n + 1**, которое не будет выведено на печать.

3) Если указаны три параметра, то третий указывает на **шаг цикла**.

Необходимо запомнить, что **тело цикла ни разу не выполнится** при условии, что первый параметр **n** больше второго параметра или они равны между собой.

Приведем несколько примеров.

Пример 1. Вывести на печать все числа от **0** до **20** в одной строке (рис. 4.6).

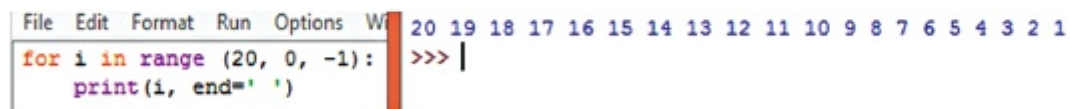


```
File Edit Format Run Options | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
for i in range (20): >>> |
    print(i, end=' ')
```

Рис. 4.6. Печать чисел от 0 до 20

Так как в диапазоне цикла не указано начальное значение, то по умолчанию значение будет равно **нулю**. Цикл выполняется 20 раз, он останавливается перед последним числом **20**. На экран будут выведены числа от **0** до **19**.

Пример 2. Вывести на печать все числа от **20** до **1** в обратном порядке на одной строке (рис. 4.7).

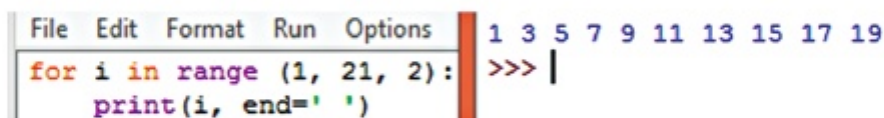


```
File Edit Format Run Options W | 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
for i in range (20, 0, -1): >>> |
    print(i, end=' ')
```

Рис. 4.7. Печать чисел в обратном порядке

Программа выводит числа от **20** до **1** с шагом цикла **1**. Предпоследнее число будет **1**.

Пример 3. Вывести на печать все нечетные числа от **1** до **21** на одной строке (рис. 4.8).



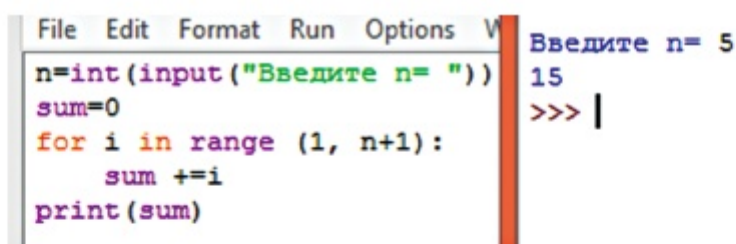
```
File Edit Format Run Options 1 3 5 7 9 11 13 15 17 19
for i in range (1, 21, 2): >>> |
    print(i, end=' ')
```

Рис. 4.8. Печать нечетных чисел

При использовании в диапазоне трех параметров:

- первый параметр (число **1**) задает начальное значение индексной переменной;
- второй параметр (число **21**) – значение, до которого будет меняться индексная переменная, не включая его;
- третий параметр (число **2**) – величина изменения индексной переменной (шаг цикла).

Пример 4. Найти сумму чисел от **1** до **n** (рис. 4.9).



```
File Edit Format Run Options V Введите n= 5
n=int(input("Введите n= ")) 15
sum=0 >>> |
for i in range (1, n+1):
    sum +=i
print(sum)
```

Рис. 4.9. Сумма чисел

Параметр цикла **i** принимает значения **1, 2, ..., n**. Поэтому значение переменной **sum** последовательно увеличивается на указанные значения. Происходит накопление суммы при начальном присвоении переменной **sum = 0**.

Таким образом, если синтаксис цикла **for** задать тремя параметрами, то при шаге цикла **d > 0** значения индексной переменной вычисляются в виде:

- i = a,**
- i = a + d,**
- i = a + 2 * d, ..., для всех значений i < b.**

for i in range(a, b, d):
тело цикла

Если шаг цикла $d < 0$, то переменная цикла принимает все значения $i > b$.

Если синтаксис цикла **for** задать с двумя параметрами, то индексная переменная i будет принимать значения от a до $b - 1$ включительно.

Первый параметр функции **range** задает начальное значение индексной переменной, а второй параметр – значение, которое оно не будет принимать.

Если один цикл помещен внутри другого, то такие циклы называются **вложенными**.



Внешняя часть вложенного цикла выполняется только после того, как завершит работу внутренняя часть.

Приведем пример, когда один цикл **for** вложен в другой цикл **for**.

Напишите программу, которая выведет на печать таблицу умножения при $n = 3$ с числами **1, 2, 3**.



	$\left. \begin{array}{l} 1 \times 1 = 1 \\ 2 \times 1 = 2 \\ 3 \times 1 = 3 \end{array} \right\}$	
	$\left. \begin{array}{l} 1 \times 2 = 2 \\ 2 \times 2 = 4 \\ 3 \times 2 = 6 \end{array} \right\}$	
	$\left. \begin{array}{l} 1 \times 3 = 3 \\ 2 \times 3 = 6 \\ 3 \times 3 = 9 \end{array} \right\}$	



Шаг за шагом



Для реализации данной программы применим вложенный цикл, состоящий из внешнего цикла и внутреннего. Если внешний цикл выполняется один раз, то внутренний цикл – три раза. В итоге внешний выполняется три раза, а внутренний – девять раз.

Например, если переменной n присвоим значение **3**, то внутренний цикл будет повторяться 9 раз. При $n = 4$ внутренний цикл выполняется 16 раз, а при $n = 7$ цикл повторяется 49 раз и т.д. Для того чтобы цикл повторялся n раз, в диапазоне функции **range** нужно поставить второй параметр, равный $n + 1$.

```
File Edit Format Run Options Window Help
n=3
for x in range (1, n+1): #значение переменной x меняется во внешнем цикле, внешний цикл
    for y in range (1, n+1):#значение переменной y для каждого значения x - от 1 до 3, внутренний цикл
        print (y, 'x',x,'=',x*y)#8 пробелов перед блоком инструкций

1 x 1 = 1
2 x 1 = 2
3 x 1 = 3
1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
1 x 3 = 3
2 x 3 = 6
3 x 3 = 9
>>> |
```



Знание. Понимание



1. Произнесите ключевые слова темы на разных языках.
2. Опишите составляющие синтаксиса оператора цикла **for**.
3. Установите соответствие.



1	Итерация	A	Не включает в себя последнее число
2	Шаг цикла	B	Числовая константа, переменная или произвольное арифметическое выражение
3	Параметр n функции range	C	Количество повторов цикла
4	Функция вызова диапазона чисел	D	Значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе

4. Что означают три параметра функции в синтаксисе **for i in range(a, b, d)**?



Применение



5. Что будет выведено на печать (рис. 4.10)? Объясните 3 способа вызова **range**.

- 1) **for i in range (2, 7, 3)** 2) **for i in range (3, 6)** 3) **for i in range (3)**
print (i) **print (i)** **print (i)**

>>> range (2, 7, 3)	>>> range (3, 6)	>>> range (3)
---------------------	------------------	---------------

Рис. 4.10. Задание для вывода на печать

6. Напишите программу с циклом **for** и использованием функции **range()**, выводящей на экран ряд натуральных чисел от минимума до максимума

с указанным шагом. Например, если **минимум 7, максимум 77, шаг 7**, то вывод должен быть таким: **7 14 21 28 35 42 49 56 63 70 77**. Минимум, максимум и шаг указываются пользователем (считываются с клавиатуры).

Анализ



7. Дан код программы, решающий следующую задачу: «Найдите три натуральных числа x , y , z , удовлетворяющих условию $15x + 20y + 30z = 270$ » (рис. 4.11). Проанализируйте код программы. Как работает цикл с параметром **for**? Можно ли его назвать вложенным? На основании данной программы попробуйте предложить собственный вариант программы.

```
File Edit Format Run Options Window Help
for x in range(1,19):
    for y in range(1,14):
        for z in range(1,10):
            if (15*x+20*y+30*z)==270:
                print("x=",x,";y=",y,";z=",z)
```

Рис. 4.11. Код программы поиска натуральных чисел x , y , z

Синтез. Оценивание



8. Приведены несколько примеров заголовков цикла. Установите соответствие, как будет меняться переменная i в приведенных примерах.

Например:

for i in range(1,101) – изменение параметра цикла от **1** до **100** включительно с шагом **1**.

- 1) **for i in range(10)**
- 2) **for i in range(10,1,-1)**
- 3) **for i in range(50,100,2)**
- 4) **for i in range(30,40,-1)**
- 5) **for i in range(16,66+2,3)**

9. Амиру дали задание написать программу, которая будет вводить десять чисел, и вывести на печать количество тех чисел, которые одновременно являются четными и положительными.

Рекомендации. Для решения задачи синтезируйте знания по циклу **for** и ветвлению.

10. Дан код программы, решающий задачу «Найти все совершенные числа до **10 000**. Совершенное число – это такое число, которое равно сумме всех своих делителей, кроме себя самого. Например, число **6** является совершенным, так как кроме себя самого делится на числа **1**, **2** и **3**, которые в сумме дают **6**». Оцените решение, предложите свое.



```

for i in range(2,10000):
    s = 0
    for j in range(1,i):
        if i%j == 0:
            s = s+j
    if s == i:
        print(i)

```



Задачи с циклом for

1. Напишите программу, вычисляющую сумму всех четных чисел в диапазоне от 1 до 90 включительно.
2. Напишите программу, вычисляющую сумму всех нечетных чисел в диапазоне от 1 до 90 включительно.
3. Напечатайте таблицу умножения на 5. Желательно печатать в виде:

$$1 \times 5 = 5$$

$$2 \times 5 = 10$$

$$\dots$$

$$9 \times 5 = 45$$

Вместо знака умножения \times можно использовать строчную латинскую букву x .

4. Найти сумму всех целых чисел от 10 до 100.
5. Найти произведение всех целых чисел от 10 до 100 включительно. Python может работать с целыми числами неограниченного размера.
6. Алия 1 марта открыла счет в банке, вложив 100 000 тенге. Через каждый месяц размер вклада увеличивается на 12% от имеющейся суммы. Определите сумму вклада, которая будет через 6 месяцев.
7. Начав тренировки, лыжник в первый день пробежал 10 километров. Каждый следующий день он увеличивал пробег на 10% от пробега предыдущего дня. Определите:
 - в какой день он пробежит больше 20 километров;
 - в какой день суммарный пробег за все дни превысит 100 километров.

4.3. УПРАВЛЕНИЕ ЦИКЛОМ: BREAK

ВЫ НАУЧИТЕСЬ

Использовать инструкции управления циклом (continue, break, else).

КЛЮЧЕВЫЕ СЛОВА

Прерывать	Үзу	Break
Управление циклом	Циклды басқару	Loop management
Прерывание цикла	Циклдың үзілуі	Loop interruption



Что надо делать, если в цикле образуются условия, которые несовместимы с его дальнейшим выполнением?

В языке программирования Python есть ключевое слово **break**, с помощью которого заранее можно остановить итерацию в циклах **while** и **for**. Инструкция **break** управляет выходом из цикла даже в том случае, если условие принимает истинное значение (рис. 4.12). И все команды, находящиеся в цикле после **break**, не будут выполняться (рис. 4.13).

Синтаксис break

Управление циклом
break
немедленный выход

Рис. 4.12. Управление циклом **break**

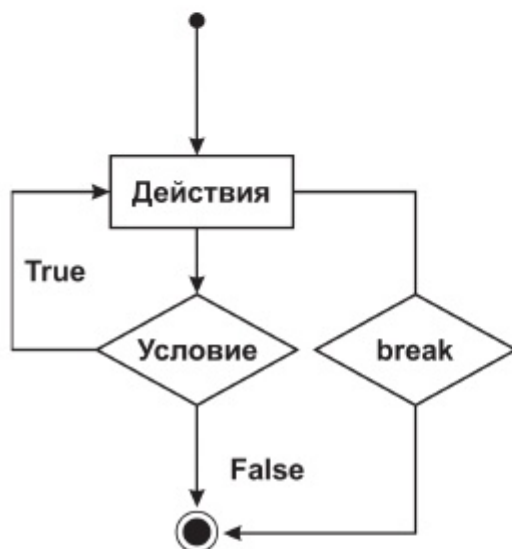


Рис. 4.13. Блок-схема **break**

Пример 1. Управление будет передано в конец цикла, если значение переменной **a** равно пяти, произойдет остановка итерации (рис. 4.14).

```
File Edit Format 2
a = 1 3
while a < 10: 4
    a += 1 >>>
    if a == 5:
        break
    print(a)
```

Рис. 4.14. Выход из цикла **while**

Цикл разрывается несмотря на то, что в условии **while** указано продолжать цикл, пока переменная не достигнет значения **10**. Встроенный **break** выполнит прерывание цикла.

Пример 2. Написать программу, в которой будет происходить перевод тенге в доллары. Пользователь вводит количество денег в тенге для обмена в доллары. Если в условие цикла будет введена цифра, равная нулю (0), то оператор **break** прервет выполнение цикла (рис. 4.15) и программа прекратит работу. Иначе происходит деление введенной суммы на обменный курс. С помощью функции **round** происходит округление результата и вывод его на печать. Цикл повторяется бесконечно, пока пользователь не захочет выйти из программы, нажав на клавишу **0**.

```
File Edit Format Run Options Window Help
# Обмен тенге в доллары
print('Нажмите на 0 для завершения обмена.')
while True:
    a = input('Введите сумму в тенге для обмена в доллары = ')
    if a == '0':
        break # выход из цикла
    tenge = int(a)
    obmen = round(tenge / 420, 2)
    print('Переведены тенге', obmen, 'в доллары.')
print('Обмен тенге в доллары завершен!')
```

```
Нажмите на 0 для завершения обмена.
Введите сумму в тенге для обмена в доллары = 100000
Переведены тенге 238.1 в доллары.
Введите сумму в тенге для обмена в доллары = 1000
Переведены тенге 2.38 в доллары.
Введите сумму в тенге для обмена в доллары = 200000
Переведены тенге 476.19 в доллары.
Введите сумму в тенге для обмена в доллары = 0
Обмен тенге в доллары завершен!
>>>
```

Рис. 4.15. Программа «Обмен тенге в доллары»

Данная программа является бесконечным циклом, так как условие **while True** имеет всегда **истинное** значение. Этот прием популярен при создании программ, которые должны выполняться так долго, пока не возникнет прерывание со стороны пользователя.

Как работает команда **break** в Python в цикле **for**? Команда **break** в цикле **for** прекращает выполнение цикла и переводит выполнение программы на следующую строку после цикла. Рассмотрим цикл с параметром **for** и прерывание цикла с помощью **break** (рис. 4.16).

```
File Edit Format Run Options Window Help
number = 0
for number in range(15):
    if number == 7:
        break # break
    print('Номер ' + str(number))
print('Выход из цикла') # завершение цикла for
```

```
Номер 0
Номер 1
Номер 2
Номер 3
Номер 4
Номер 5
Номер 6
Выход из цикла
```

Рис. 4.16. Цикл с параметрами **for** и **break**

Проанализировав код программы (рис. 4.16), ответьте на вопросы:

- 1) В какое число инициализирована переменная **number**?
- 2) До каких пор оператор **for** создает цикл?
- 3) При каком условии **if** будет прерван цикл?
- 4) Что будет выведено оператором **print** на печать?
- 5) В каком случае будет прерван вывод данных на печать?
- 6) Что делает оператор **break**?



Знание. Понимание



1. Произнесите ключевые слова на трех языках.
2. Вставьте пропущенные слова.
 - 1) Оператор **break** досрочно ...
 - 2) Инструкция **break** вызывается только ..., то есть она должна выполняться только при выполнении какого-то особенного условия.
 - 3) Для избегания бесконечного цикла используют ...



Применение. Анализ



3. Дан код программы, которая считывает числа до тех пор, пока не встретит отрицательное число. При появлении отрицательного числа программа завершается. Составьте аналогичную программу, используя цикл **for**.
4. Дан код программы. Прокомментируйте действия.



```

i = 0
while i < 10:
    if i == 3:
        i += 1
        continue
    print(i)
    if i == 5:
        break
    i += 1
  
```

5. Запрашиваются 10 чисел (целые значения от 0 до 1000). Опишите и проанализируйте алгоритм, позволяющий найти и вывести минимальное значение среди введенных чисел, которые имеют четное значение и не делятся на три.
6. Составьте программу для вычисления среднего арифметического **n** произвольно вводимых чисел.



Синтез. Оценивание



7. Определите, что получится в результате выполнения программы. Найдите альтернативное решение.

```

a = 0
n = 10
for i in range(1, n + 1, 2):
    a = a + 1
    if a == 5:
        continue
    print (a)
    if a == 7:
        break

```

8. Напишите эссе о сделанных вами выводах при использовании оператора `break`.



Решение одной задачи двумя способами

Напишите программу вывода на экран таблицы умножения всех однозначных чисел от 1 до 5 разными способами. Сравните и оцените их решения. Какое решение вы считаете лучшим? Возможно ли в программе прервать выполнение цикла?

Рассмотрим первый вариант.

Применим цикл **while** и два цикла. Во внешнем цикле переменная **i** меняется от **1** до **5**, а во внутреннем цикле переменная **j** также меняется от **1** до **5**. В блоке вложенного цикла на экран печатаются значения переменных **i**, **j** и их произведение в виде таблицы (рис. 4.17).

File	Edit	Format	Run	Options	Window	Help
<pre>i=1</pre>						
<pre>while i<5: # Условие внешнего цикла по i</pre>						
<pre> j=1</pre>						
<pre> while j<5: # Условие внутреннего цикла по j</pre>						
<pre> print (i, "+", j, "=", i*j)</pre>						
<pre> j=j+1 # Итерация j внутреннего цикла</pre>						
<pre> i=i+1 # Итерация i внешнего цикла</pre>						

1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16

Рис. 4.17. Цикл с условием **while**

Второй вариант решения задачи с использованием цикла **for** показан на рисунке 4.18.

```
File Edit Format Run Options Window Help
for i in range(1,5):
    for j in range(1,5):
        print(i, "*", j, "=", i*j)
>>> |
```

1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
>>> |

Рис. 4.18. Цикл с условием **for**

4.4. УПРАВЛЕНИЕ ЦИКЛОМ: CONTINUE

ВЫ НАУЧИТЕСЬ

Использовать инструкции управления циклом (**continue**, **break**, **else**).

КЛЮЧЕВЫЕ СЛОВА

Продолжить	Жалғастыру	Continue
Прерывать	Үзу	Break
Диапазон	Диапазон	Range



Как можно в программе пропустить часть цикла с условием, а затем продолжить выполнение до конца цикла?



В языке программирования Python для управления циклом, кроме оператора **break**, используется оператор **continue**. Конструкция управления циклом обычно представляется в виде последовательности операторов: **break** (немедленный выход), а затем **continue** (рис. 4.19). С помощью **continue** можно прервать текущую итерацию

Управление циклом
break
немедленный выход
continue
следующая итерация

Рис. 4.19. Управление циклом

цикла, и управление будет передано в начало цикла. После этого снова проверяется условие цикла. Если оно будет истинным, то продолжается выполнение следующей итерации.

В программе на рисунке 4.20 оператор **continue** прерывает текущую итерацию цикла, если в условии цикла **a = 5**. После прерывания продолжается выполнение операторов до конца цикла. Происходит переход на начало цикла **while** и проверка условия. Если оно истинно, то цикл продолжает выполняться.

Рассмотрим код программы (рис. 4.21), в которой необходимо перебрать в цикле все натуральные числа меньше **n**, заданные в диапазоне **range** и не делящиеся на **3**. В программе применяется оператор **continue** для управления циклом. Он продолжает выполнение следующей итерации цикла без выхода из нее.

```
File Edit Format Run 1
2
a=0
while a < 10:
3
4
    a += 1
5
6
    if a == 5:
7
8
        continue
9
10
    print(a)
```

Рис. 4.20. Оператор **continue**

```
File Edit Format Run 1
2
i=0
for i in range(10):
3
4
    if i%3==0:
5
6
        continue
7
8
    print(i)
9
10
>>>
```

Рис. 4.21. Перебор натуральных чисел

Оператор **continue** находится внутри блока кода под оператором цикла, обычно он размещается после условного оператора **if** (рис. 4.22).

```
File Edit Format Run Options Window Help
number = 0
for number in range(10):
    if number == 8:
        continue # прерывание итерации
    print('Номер ' + str(number)) # str функция перевода параметра в строку
print('Выход')
```

Номер 0
Номер 1
Номер 2
Номер 3
Номер 4
Номер 5
Номер 6
Номер 7
Номер 8
Номер 9
Выход

Рис. 4.22. Прерывание цикла с оператором **continue**

Оператор **continue** в программе выполняется до тех пор, пока не произойдет прерывание итерации цикла. Если в условии цикла **number = 8**, то этот номер не будет выведен на печать. Управление будет передано на начало цикла, снова проверяется условие и продолжается ее выполнение до конца цикла.

В каких случаях необходимо применять оператор **continue**? Оператор **continue** применяется:

- если необходимо прервать вложенный условный код;
- для оптимизации цикла, если необходимо отклонить часто встречающиеся действия.

Оператор **continue** нельзя применять за пределами цикла. Он применяется исключительно в циклах.

Преимущества использования операторов **continue** и **break** в программах:

- если нужный результат достигнут либо уже найдено искомое значение при выполнении циклического процесса, то можно прервать лишние итерации, что повысит скорость работы программы;
- структура программы не меняется в случае преждевременного прекращения циклического процесса.

Знание. Понимание



1. Произнесите ключевые слова на трех языках.
2. Что такое оператор **continue**?
3. Можно ли использовать **continue** с циклом **for**?
4. Когда инструкция **continue** влияет только на выполнение внутреннего цикла?



Применение. Анализ



5. Определите, что выведет данная программа.

```
a = 0
while a! = 7:
    a = a + 1
    if a == 5:
        continue
    print (a)
```



6. Проанализируйте код программы (рис. 4.23). Составьте к ней условие, определите результат. Составьте аналогичную программу без использования оператора **continue**.

```
File Edit Format Run Options Window Help
for num in range(2, 10):
    if num % 2 == 0:
        print ('Четное число ', num)
        continue
    print ('Нечетное число' , num)
```

Рис. 4.23. Код программы

**Синтез. Оценивание**

7. На рисунке 4.24 дан результат программы. Синтезируйте полученные знания о циклах. Составьте условие задачи и код программы, выводящий на печать аналогичный результат.

8. Оцените циклы **while**, **for** и проведите их сравнительный анализ. Напишите заключение.



В Python цикл с постусловием будет выглядеть таким образом:

```
while True:
    тело цикла
if not условие:
    break;
```

Выполняется тело цикла, изменяется параметр цикла, прерывается условие продолжения выполнения цикла.

```
0
прерывание
1
прерывание
2
прерывание
3
прерывание
4
прерывание
5
6
7
8
9
>>>
```

Рис. 4.24. Результат выполнения программы

4.5. УПРАВЛЕНИЕ ЦИКЛОМ: ELSE**ВЫ НАУЧИТЕСЬ**

Использовать инструкции управления циклом (`continue`, `break`).

КЛЮЧЕВЫЕ СЛОВА

Иначе	Эйтпесе	Else
Пока	Эзире	While



В ветвлении применяется блок **else**. Есть ли необходимость применения блока **иначе (else)** в цикле **for** и **while**?

В циклах **for** и **while** также используется блок **else**. Рассмотрим цикл **while** с использованием блока **else**.



Если используется оператор **else** в цикле **while**, то блок **else** выполняется в том случае, если цикл завершил итерацию, и не выполняется, если в цикле был выполнен **break**.

Приведем примеры.

Пример 1. Цикл **while** с оператором **else** (рис. 4.25).

```
File Edit Format Run
i=0
while i<7:
    print(i)
    i=i+1
else:
    print('Выход')
```

0
1
2
3
4
5
6
Выход

Рис. 4.25. Код программы цикла **while/else**

Блок **else** выполняется после завершения цикла **while**. После вывода всех чисел от **0** до **7** выводится на экран слово **'Выход'**.

Пример 2. Цикл **while** с **else** и **break** (рис. 4.26).

```
File Edit Format Run Op
i=0
while i<7:
    if i==5:
        break
    else:
        print(i)
        i=i+1
else:
    print('Выход')
```

0
1
2
3
4
>>>

Рис. 4.26. Код программы цикла **while/else/break**

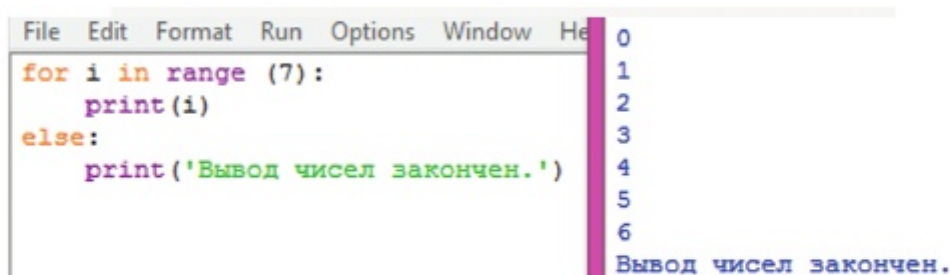
Блок **else** не выполняется в цикле из-за оператора **break**. При выполнении программы, если в условии параметр цикла **i** будет равен **5**, происходит прерывание итерации. Не будет выведено на печать слово **'Выход'**.

Цикл **for** также может иметь ветвление с **else** и содержать инструкции **break**, **continue** внутри себя.

Если используется цикл **for** с оператором **else**, то блок **else** выполняется тогда, когда цикл завершает итерацию, и не выполняется **else**, если в цикле был выполнен **break**.



Пример 3. Рассмотрим демонстрацию цикла **for** с блоком **else** (рис. 4.27).

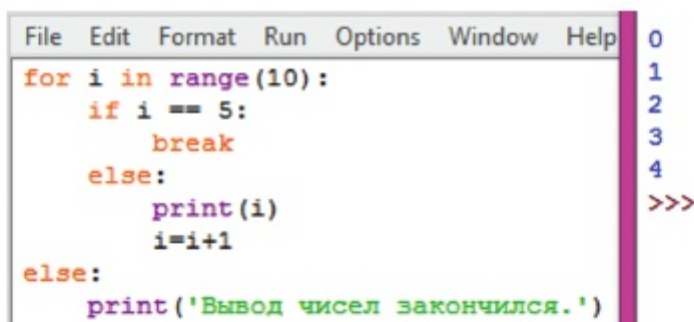


```
File Edit Format Run Options Window Help 0
for i in range (7):
    print(i)
else:
    print('Выход чисел закончен.')
1
2
3
4
5
6
Выход чисел закончен.
```

Рис. 4.27. Код программы цикла **for/else**

Как и в цикле **while**, блок **else** выполняется после завершения цикла **for**, и на печать будет выведено **'Выход чисел закончен'**.

Пример 4. Цикл **for** с оператором **else** и **break** (рис. 4.28).



```
File Edit Format Run Options Window Help 0
for i in range(10):
    if i == 5:
        break
    else:
        print(i)
        i=i+1
else:
    print('Выход чисел закончился.')
1
2
3
4
>>>
```

Рис. 4.28. Код программы цикла **for/else/break**

В примере цикла **for** блок **else** не выполняется из-за **break** (прерывания). Блок **else**, примененный в цикле **for** или **while**, проверяет, был ли произведен выход из цикла инструкцией **break** или же естественным образом. Блок инструкций внутри **else** выполнится только в том случае, если выход из цикла произошел без помощи **break**.

Пример 5. Цикл **for** с **else** и **continue** (продолжение цикла) (рис. 4.29).

Инструкция управления циклом **continue** (продолжение цикла): в коде программы инструкция встретилась в середине цикла **i=5**, поэтому были пропущены все оставшиеся инструкции до конца цикла. А исполнение цикла продолжится со следующей итерации.

```

File Edit Format Run Options Window Help 0
for i in range(10): 1
    if i == 5: 2
        continue 3
    else: 4
        print(i) 5
        i=i+1 6
else: 7
    print('Вывод чисел закончился.') 8
9
Вывод чисел закончился.
>>>

```

Рис. 4.29. Код программы цикла **for/else/continue**

Цикл **for** сложнее цикла **while**, но выполняется намного быстрее. Этот цикл проходит по любому итерируемому объекту и во время каждого прохода выполняет тело цикла.

Знание. Понимание



1. Объясните, каким образом работает данный ниже синтаксис.

for <переменная> **in** <последовательность>:

<действие>

else:

<действие>

2. Сколько раз выполнится блок операций, следующий после **else**, когда проверяемое условие станет неверным?

3. В каком случае ветка **else** исполняться не будет?

4. Может ли ветка **else** использоваться внутри цикла **for**?

5. Какова главная задача оператора **else**?



Применение. Анализ



6. Определите, что получится в результате выполнения данной программы (рис. 4.30).



```

File Edit Format Run Options Window Hel
count = 3
while count < 8:
    print (count, " меньше 8")
    count = count + 1
else:
    print (count, " не меньше 8")

```

Рис. 4.30. Код программы «Меньше не меньше 8»

7. Дан код программы. Проанализируйте его. Определите условие задачи. Обоснуйте использование ветки **else**.

```
i = 1
while i <= 5:
    print(i ** 3)
    i + = 1
else:
    print('Цикл окончен, i =', i)
```



Синтез. Оценивание



8. Дан код программы. Определите условие. Возможно ли создать аналогичный код, используя цикл **for**? Обоснуйте свой ответ.

```
a = int(input())
while a != 0:
    if a % 2 == 0:
        print('четное число', a)
        break
    a = int(input())
else:
    print('Ни одного четного числа не встретилось')
```

9. На рисунке 4.31 приведен код программы. Напишите комментарии к каждой строке и объясните, каким образом выполняется программа. Введите число **8**. Что будет выведено на печать? Введите число **3**. Что будет выведено на печать?

```
File Edit Format Run Options Window Help
n=int(input("Введите целое число, больше 1="))
for d in range(2, n):
    if n % d == 0:
        print (n, 'Имеет делитель', d)
        break
    else:
        print (n, 'Простое число')
```

Рис. 4.31. Код программы «Простое число»

10. Напишите программу, в которой нужно вести **n** целых чисел. Если будет введено целое отрицательное число, то выведется сообщение **'Встретилось отрицательное число.'**, иначе – **'Все числа положительные.'** Оцените программу, удалив блок **break**.

11. Дан код программы по определению простого числа в диапазоне от 1 до 10. Оцените полученный результат. Усовершенствуйте код.

```
for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            print(n, '=', x, '*', n/x)
            break
        else:
            print(n, '- простое число')
```



Вывод таблицы символов

В Python в версии 3 для строк используется кодировка Unicode. В Python, в отличие от многих других языков программирования, нет такого типа данных как одиночный символ. В Python любой символ – это строка, длина которой равна единице. Первые 128 символов по таблице Unicode такие же, как и в таблице символов ASCII. Выведем их, начиная с пробела, кодовый номер которого 32. Чтобы привести вывод к табличной форме, будем переходить на новую строку после каждого десятого выведенного на экран символа. Для этого в коде, показанном ниже, используется оператор **if**. Функция **chr()** возвращает символ из таблицы Unicode, соответствующий переданному коду-числу.

```
for i in range(32, 128):
    print(chr(i), end=' ')
    if (i - 1) % 10 == 0:
        print()
```

Результат выполнения кода:

```
! « # $ % & ' ( )
* + , - . / 0 1 2 3
4 5 6 7 8 9 : ; < =
> ? @ A B C D E F G
H I J K L M N O P Q
R S T U V W X Y Z [
\ ] ^ _ ` A B C D E
F G H I J K L M N O
P Q R S T U V W X Y
Z { | } ~ □
```

4.6. ТРАССИРОВКА АЛГОРИТМА



ВЫ НАУЧИТЕСЬ

Осуществлять трассировку алгоритма.

КЛЮЧЕВЫЕ СЛОВА

Трассировка алгоритма	Алгоритмді талқылау	Algorithm Tracing
Отладка	Түзету	Debugging
Создание файла	Файл жасау	File creation
Открытие файла	Файлды ашу	Opening a file
Сохранение файла	Файлды сақтау	Saving a file



Что такое трассировка? Для чего нужна трассировка?

Отладка – это процесс поиска и исправления ошибок в программе, препятствующих корректной работе программы. Отладчик позволяет выполнять **трассировку** программы, наблюдать значения переменных, контролировать выводимые программой данные.



Трассировка – это процесс выполнения программы по шагам (*step-by-step*), инструкция за инструкцией. Во время работы программы инструкции выполняются одна за другой со скоростью работы процессора компьютера.

В случае неправильной работы программы необходимо видеть реальный порядок выполнения инструкций. Это можно сделать, выполнив трассировку программы.

Трассировка содержит важную информацию, которая может помочь найти ошибки в коде программы. Поэтому использование вывода трассировки при запуске кода – это возможность улучшить код программы. В Python можно использовать встроенный модуль **traceback** для работы и проверки трассировок.

Рассмотрим основные этапы выполнения программ в интегрированной среде Python, приведенные в таблице 4.1.

Таблица 4.1. Этапы выполнения программы


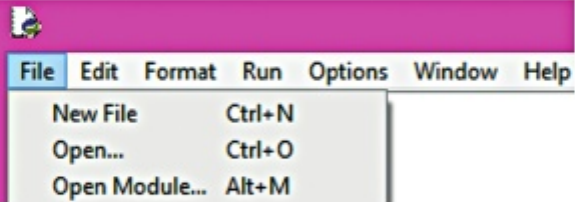
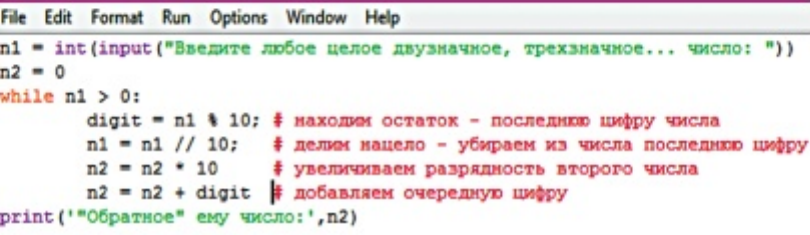
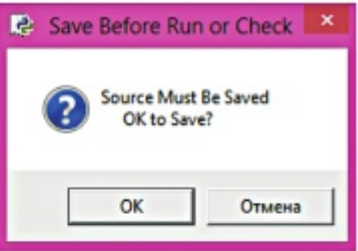
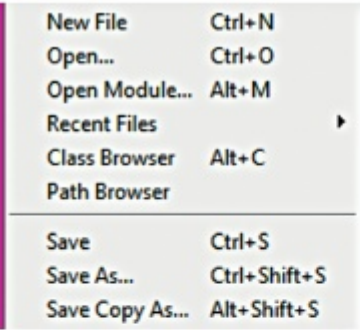
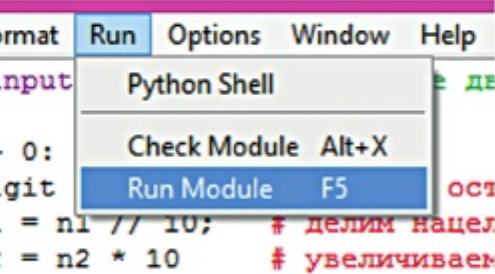
Этапы выполнения	Порядок выполнения
1. Загрузка IDLE Python	С помощью ярлыка  на Рабочем столе или с Панели задач
2. Создание или открытие файла	
3. Ввод и редактирование программы	 <pre>File Edit Format Run Options Window Help n1 = int(input("Введите любое целое двузначное, трехзначное... число: ")) n2 = 0 while n1 > 0: digit = n1 % 10; # находим остаток - последнюю цифру числа n1 = n1 // 10; # делим нацело - убираем из числа последнюю цифру n2 = n2 * 10 # увеличиваем разрядность второго числа n2 = n2 + digit # добавляем очередную цифру print("Обратное" ему число:', n2)</pre>
4. Сохранение файла	 
5. Запуск программы на выполнение с сохранением	

Таблица 4.1. Продолжение

Этапы выполнения	Порядок выполнения				
6. Отладка программы	<pre data-bbox="410 270 1200 356"> Введите любое целое двузначное, трехзначное... число: 871 "Обратное" ему число: 178 >>> </pre> <p data-bbox="400 386 1208 540">После запуска программы на выполнение открывается окно, в котором выводится сообщение об ошибке в случае ее обнаружения. После устранения ошибок необходимо снова запустить программу на выполнение. При этом IDLE Python снова выводит окно сохранения программы</p>				
7. Завершение работы ИСР	<table border="1" data-bbox="400 562 771 652"> <tr> <td>Close</td> <td>Alt+F4</td> </tr> <tr> <td>Exit</td> <td>Ctrl+Q</td> </tr> </table>	Close	Alt+F4	Exit	Ctrl+Q
Close	Alt+F4				
Exit	Ctrl+Q				

Программа может содержать различные ошибки, обнаруженные на этапе отладки. При устранении ошибок необходимо помнить, какого типа они могут быть (рис. 4.32).

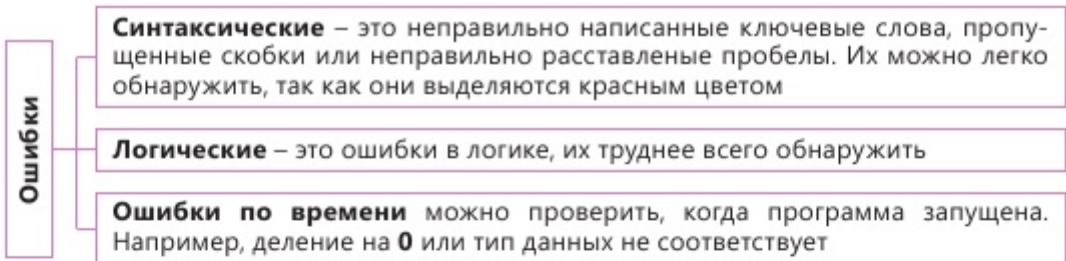


Рис. 4.32. Типы ошибок в IDLE Python

Чтобы избежать ошибки при выполнении программы, необходимо выполнить пошаговую трассировку. Проверка на каждом шаге позволяет определить решение поставленной задачи, вникнуть в логику работы программы и правильность рассуждений.

Чтобы представить, как проводится трассировка в программе, попробуем рассмотреть пример ручной проверки шаг за шагом.



В представленном коде программы (рис. 4.33) цикл выполняется до тех пор, пока **k < 300**. Если условие истинное, то **k** увеличивается в **5** раз, а переменная **s** увеличивается на **10**. Какое число будет выведено на печать? Выполните анализ и трассировку программы.

File Edit Format Run Options	Итерация 15
k=1	Итерация 25
s=5	Итерация 35
while k<300:	Итерация 45
s=s+10	Сумма= 45
k=k*5	>>>
print('Итерация ', s)	
print('Сумма=', s)	

Рис. 4.33. Код программы «Сумма»

Способ 1.**Анализ решения задачи**

1. Переменная **k** изначально равна **1**.
2. Начальное значение переменной **s** равно **5**.
3. Цикл выполняется до тех пор, пока **k < 300**.
4. На каждом шаге цикла значение переменной **k** увеличивается в **5** раз.
5. Условие цикла перестанет выполняться при **k = 625**.
6. Значение переменной **k** увеличивалось **5** раз четырежды.
7. Цикл выполнялся **4** раза.
8. На каждом шаге цикла значение переменной **s** увеличивалось на **10**.
9. За **4** шага цикла значение переменной **s** увеличивалось на **4 x 10 = 40**.
10. Значит, итоговое значение переменной **s** равно **5 + 40 = 45**, так как начальное значение переменной **s** равно **5**.

Способ 2.

Выполним трассировку программы по шагам (табл. 4.2).

Таблица 4.2. Таблица трассировки

Оператор	Условие	Переменные		Пояснения
		k	s	
k=1		1		Переменная k получает начальное значение, равное 1
s=5			5	Переменная s получает начальное значение, равное 5
while k<300:	1<300? True			Начинается цикл while , так как условие истинно, то выполняется тело цикла

Таблица 4.2. Продолжение

Оператор	Условие	Переменные		Пояснения
		k	s	
$s=s+10$			15	Значение переменной s (5) , увеличивается на 10
$k=k*5$		5		Значение переменной k (1) , увеличивается в 5 раз
while k<300:	5<300? True			Снова проверяется условие цикла while , так как условие истинно, то выполняется тело цикла
$s=s+10$			25	Значение переменной s (15) , увеличивается на 10
$k=k*5$		25		Значение переменной k (5) , увеличивается в 5 раз
while k<300:	25<300? True			Снова проверяется условие цикла while , так как условие истинно, то выполняется тело цикла
$s=s+10$			35	Значение переменной s (25) , увеличивается на 10
$k=k*5$		125		Значение переменной k (25) , увеличивается в 5 раз
while k<300:	125<300? True			Снова проверяется условие цикла while , так как условие истинно, то выполняется тело цикла
$s=s+10$			45	Значение переменной s (35) , увеличивается на 10
$k=k*5$		625		Значение переменной k (125) , увеличивается в 5 раз
while k<300:	625<300? False			Снова проверяется условие цикла while , так как условие ложное , то не выполняется тело цикла
print('Сумма=', s)				На экран будет выведено текущее значение 45
Ответ:				45

Знание. Понимание

1. Произнесите ключевые слова на трех языках.
2. Что такое отладка?
3. Что называется трассировкой?
4. Какие виды ошибок существуют при отладке программы?
5. В приведенном коде исправьте ошибки, чтобы код напечатал сумму чисел от 1 до 19.



```
s = 1
for i in range(0,19):
    s = s + i
print(s)
```

Применение. Анализ

6. Составьте и оформите программу, которая выводит таблицу степеней **двойки** от **нулевой** до **пятой**.

```
1 2
2 4
3 8
4 16
5 32
```



Выполните трассировку программы и заполните таблицу 4.3.

Таблица 4.3. Трассировка программы

№	Оператор	Условие цикла	i	2 ** i	Примечание
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

7. Создайте и оформите программу, которая выводит таблицу значений функции $y = -2,4x^2 + 5x - 3$ в диапазоне от -2 до 2 с шагом $0,5$. Ниже приведен рекомендуемый вид экрана программы (рис. 4.34).

x	y
-2	-22.60
-1.5	-15.90
-1	-10.40
-0.5	-6.10
0	-3.00
0.5	-1.10
1	-0.40
1.5	-0.90
2	-2.60

Рис. 4.34. Рекомендуемый вид экрана программы

Выполните трассировку программы.



Синтез. Оценивание



8. Составьте блок-схему и таблицу трассировки для задачи, которая выводит на экран таблицу значений функции $y = 2x^2 - 5x - 8$ в диапазоне от -4 до 4 . Шаг изменения аргумента равен $0,5$.

Рекомендации для выполнения задания

Для переменных присвойте значения, указанные в диапазоне:

$$x_1 = -4;$$

$$x_2 = 4;$$

$$dx = 0,5.$$

Используйте цикл **while** ($x < x^2$) для проверки условия.

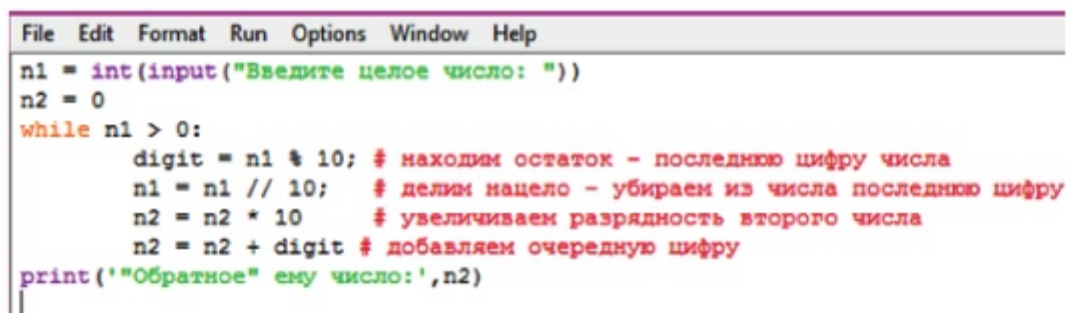
9. Выберите проблемную ситуацию, для которой необходимо написать проект для автоматизации расчета данных с использованием цикла. Сформулируйте проблему в виде задачи. Для решения задачи определите исходные данные. Выберите оптимальный вариант цикла.



Задача 1. «Перевернуть число». Напишите программу. Вводится целое число. Вывести число, обратное введенному по порядку составляющих его цифр. Например, введено 3425, надо вывести 5243.

Алгоритм решения задачи

1. Найдем остаток от деления на **10** исходного числа. Тем самым получим последнюю его цифру.
2. Добавим эту цифру к новому числу.
3. Разделим нацело на **10** исходное число. Тем самым избавимся от последней цифры в нем.
4. Снова найдем остаток от деления на **10** того, что осталось от первого числа. Запомним эту цифру.
5. Умножим на **10** второе число. Этим увеличим его разрядность до двух и сдвинем первую цифру в разряд десятков.
6. Добавим к полученному второму числу запомненную ранее цифру из первого числа.
7. Будем повторять действия п. 3–6, пока исходное число не уменьшится до нуля, т.е. пока не избавимся от всех его разрядов (рис. 4.35).



```
File Edit Format Run Options Window Help
n1 = int(input("Введите целое число: "))
n2 = 0
while n1 > 0:
    digit = n1 % 10; # находим остаток - последнюю цифру числа
    n1 = n1 // 10; # делим нацело - убираем из числа последнюю цифру
    n2 = n2 * 10 # увеличиваем разрядность второго числа
    n2 = n2 + digit # добавляем очередную цифру
print("Обратное" ему число:', n2)
```

Рис. 4.35. Программа «Перевернуть число»

Задача 2. «Найти сумму n -элементов ряда $1, -0,5, 0,25, -0,125, \dots$ »

Дан ряд чисел $1, -0,5, 0,25, -0,125, \dots$. Требуется найти сумму столько его элементов, сколько указал пользователь. Например, если $n = 3$, то надо сложить $1, -0,5, 0,25$, что в результате даст $0,75$.

При решении таких задач сначала необходимо определить, в чем заключается особенность предложенного ряда. В данном случае видим, что каждый последующий элемент в 2 раза меньше предыдущего по модулю и взят с противоположным знаком. Далее следует найти арифметическое действие, которое из предыдущего элемента дает последующий. Здесь, например, надо предыдущий элемент делить на -2 .

Алгоритм решения задачи

1. Присвоим переменной **a** первое число ряда.
2. Создадим счетчик суммированных элементов ряда **i**.

3. Создадим переменную для накопления суммы элементов ряда.
4. Пока счетчик не отсчитает заданного количества элементов ряда (т.е. пока $i < n$), будем выполнять следующие действия:
 - 1) К сумме добавим значение текущего элемента.
 - 2) Изменим значение текущего элемента на то, которое должно быть следующим. В данном случае разделим на -2 .
 - 3) Увеличим значение счетчика i на 1 .

```

File Edit Format Run Options Window Help
n = input("Количество элементов ряда: ")
n = int(n)
a = 1
i = 0
summa = 0
while i < n:
    summa += a
    a = a/-2
    i += 1
print(summa)

```

Количество элементов ряда: 10
0.666015625
>>> |

Рис. 4.36. Программа «Найти сумму элементов...»

Тестовые задания к разделу IV

1. Сколько раз будет выполнено тело цикла?

`while False:`

`<тело цикла>`

- A) 0.
- B) 1.
- C) 2.
- D) Бесконечный цикл.
- E) 10.

2. Из перечисленных слов выберите инструкции управления циклом в языке Python. (Возможен выбор нескольких вариантов ответа.)

- A) true.
- B) while.
- C) else.
- D) break.
- E) input.

3. Описание какого цикла приведено в предложении «Пока не пойму эту тему, я буду читать учебник»?

- A) С предусловием.
- B) С постусловием.
- C) С заданным числом повторений.
- D) С параметром.
- E) С бесконечным циклом.

4. На рис. 4.37 дан код программы. Какое значение будет иметь на выходе переменная z , если $x = 20$? Ответ округлите до целых чисел.

```
File Edit Format Run Options Window Help
x=int(input('Введите значение x'))
z=1
i=0
while i<4:
    z=(z+x)/2
    i=i+1
print(z)
```

Рис. 4.37. Код программы

- A) 1.
- B) 4.
- C) 8.
- D) 9.
- E) 10.

5. Сколько раз будет выполнено тело цикла?

for i in range (2, 8, 2):

<тело цикла>

- A) 1.
- B) 2.
- C) 3.
- D) 4.
- E) 8.

6. Сколько раз будет выполнено тело цикла?

for i in range (3):

<тело цикла>

- A) 0.
- B) 1.
- C) 2.
- D) 3.
- E) 4.

7. Что произойдет при попытке исполнения инструкции?

t = range (1, 4, -1)

- A) Переменной t будет присвоено значение 1.
- B) В переменной t будет сгенерировано пустое множество.
- C) Переменной t будет присвоено значение 4.
- D) Переменной t будет присвоено значение -1.
- E) Переменной t будет присвоено значение 0.

8. Установите соответствие между циклами на языке Python и их типами.

Укажите соответствие для всех трех вариантов ответа.

1. Цикл while с break	2. Цикл с предусловием	3. Цикл с параметром
__ while i < 5: i = i + 1	__ for i in range (8): k = random ()	__ while True: i = i + 2 if i > 10: break

- A) 3 1 2.
- B) 1 3 2.
- C) 2 1 3.
- D) 3 2 1.
- E) 1 2 3.

9. Какое значение примет переменная x после выполнения фрагмента программы (рис. 4.38)?

- A) 0
- B) 1.
- C) 2.
- D) 4.
- E) 5.

```
File Edit Format Ru
i=0
while i<=5:
    x=i
    i=i+2
print (x)
```

Рис. 4.38. Фрагмент программы

10. Как называются алгоритмы, содержащие конструкцию повторения?

- A) Рекурсивные.
- B) Линейные.
- C) Циклические.
- D) Разветвляющиеся.
- E) Процедурные.

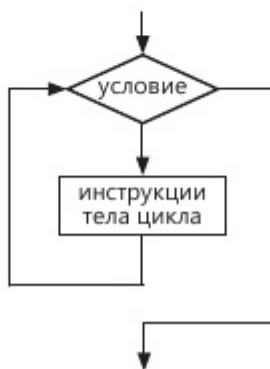


Рис. 4.39. Блок-схема цикла

11. Блок-схема какого вида цикла представлена на рисунке 4.39?

- A) С предусловием.
- B) С постусловием.
- C) С параметром.
- D) С окончанием.
- E) С продолжением.

12. Определите, что будет напечатано в результате выполнения инструкций (рис. 4.40).

- A) 0.
- B) 5.
- C) 6.
- D) 10.
- E) 123.

```
File Edit Format Run
s = 0
m = 123
while m > 0:
    d = m % 10
    s = s + d
    m = m // 10
print (s)
```

Рис. 4.40. Инструкции для выполнения






13. Определите, что будет напечатано в результате выполнения инструкций (рис. 4.41).

- A) 0.
- B) 1.
- C) -1.
- D) 20.
- E) 19.

```
File Edit Format Run
s = 1
for k in range(20):
    s = (-1) * s
print (s)
```

Рис. 4.41. Инструкции для выполнения

14. С помощью какого блока в блок-схеме изображается цикл с параметром?

- A) 
- B) 
- C) 
- D) 
- E) 

15. Определите, что будет напечатано в результате выполнения программы (рис. 4.42).

```
File Edit Format Run Options Window Help
s = 0
for k in range(3,11):
    s = s + k
print(s)
```

Рис. 4.42. Цикл for

- A) 50.
 B) 52.
 C) 67.
 D) 0.
 E) 55.

5.1. ПОСТАНОВКА ПРОБЛЕМЫ

ВЫ НАУЧИТЕСЬ

Создавать модели задач на языке программирования Python (пайтон).

КЛЮЧЕВЫЕ СЛОВА

Модель	Постановка проблемы
Модель	Мәселені қалыптастыру
Model	Formulation of the problem



Что такое проблема?
В каком случае возникает проблема?



Познавая мир, человек действует по схеме: «Проблема – исследование – решение проблемы». Проблема чаще всего возникает тогда, когда для решения какой-либо задачи не хватает имеющихся знаний. Предварительная постановка проблемы является научной творческой деятельностью человека. Многие открытия были сделаны благодаря возникшим проблемам. Например, в Казахстане в 2019 г. технологическая компания Hardware Solutions LLP создала беспилотного робота-доставщика, который решил «проблему последней мили». Проблема заключалась в том, что стоимость доставки со склада одной страны на склад в другой стоит столько же, сколько и доставка со склада до двери клиента.

Проблема – есть форма научного отображения проблемной ситуации. Она формулируется как выражение необходимости изучения и практических действий, направленных на выявление причин и на их разрешение. В процессе постановки проблемы выделяют следующие этапы (рис. 5.1):



Рис. 5.1. Постановка проблемы

1. Формулирование проблемы – это выдвижение центрального вопроса проблемы и предположительного описания ожидаемого результата.

2. Оценка проблемы. В оценку проблемы входит определение всех необходимых для ее решения условий и средств:

- методы исследования;
- источники информации;
- виды научного обсуждения программы и методик исследования;
- конечные и промежуточные результаты и др.

3. Обоснование проблемы. В детальное обоснование проблемы входят следующие компоненты:

– поиск аргументов в пользу необходимости ее решения, научной или практической ценности ожидаемых результатов;

– выдвижение исследователем возможных возражений против существования проблемы, возможности ее разрешения научным или практическим путем.

4. Структурирование проблемы. Процесс структурирования начинается с расщепления проблемы на второстепенные вопросы. Ответы на них позволят получить ответ на основной проблемный вопрос.



Какие задачи можно решать с помощью моделей? Как создавать модели задач в интегрированной среде разработки программ, применяя постановку проблем?

Мы возвращаемся к теме «Моделирование», которую изучили в предыдущих классах. Моделирование является эффективным средством для решения различных задач из окружающего мира. А модель – это условный образ объекта в виде компьютерных диаграмм, таблиц, схем, изображений, анимационных фрагментов, т.е. это алгоритм, программа, отображающие структуру и взаимосвязи между элементами объекта.

Для правильного выбора той или иной задачи приведем их классификацию по предметным областям. Для моделирования можно выделить следующие виды задач (см. рис. 5.2).

Для решения задачи первоначально надо поставить (сформулировать) проблему. Затем с помощью анализа информации об изучаемом явлении или объекте необходимо выделить **исходные данные**, после этого определить, что будет **результатом** и какая будет связь между **исходными данными** и **результатом**.



Рис. 5.2. Классификация видов задач

Для постановки проблемы необходимо ответить на следующие вопросы:

1. Что мы хотим сделать?
2. Для чего?
3. Какая должна быть программа?



Отвечая на вопросы, вы должны четко сформулировать идею задачи. Приведем несколько примеров задач.

- 1) «Хочу написать программу, которая будет выводить на экран визитную карточку о пользователе».
- 2) «Хочу написать программу, которая будет генерировать 2D-лабиринт и позволит пользователю перемещаться по нему».
- 3) «Хочу написать программу для устройства, которое будет определять и устанавливать необходимую освещенность для теплицы».
- 4) «Хочу написать программу, которая будет определять необходимые нагрузки для тренировок».

Задача, которая решается с помощью моделей, обычно связана с областью применения объекта, процесса или явления. Для одного объекта может быть множество моделей, и наоборот, разные объекты могут описываться од-

ной моделью. Например, в механике различные материальные тела (от планеты до песчинки) могут рассматриваться как материальные точки.

У каждого объекта имеется большое количество информационных аспектов. При решении конкретной задачи вас должны интересовать только определенные свойства изучаемого объекта. И тогда модель оказывается полезным, а иногда и единственным инструментом исследования.



Составьте программу «Тренировка».



Шаг за шагом

1. Постановка проблемы. Необходимо составить программу, определяющую, за сколько дней может похудеть спортсмен со 100 кг до 80 кг, если в день он сможет сбрасывать 1% своего веса от нормы предыдущего дня.

2. Формализация. Обозначим через переменную n вес спортсмена, переменную k – количество дней. Так как в день спортсмен сбрасывает 1% своего веса от нормы предыдущего дня, то вес определяется по формуле $n=n-n/100$. Спортсмен должен похудеть до 80 кг, следовательно, условием, при котором спортсмен должен сбрасывать вес, будет $n>80$. Для подсчета количества дней будем выполнять действие $k=k+1$. Можно вывести не только день, когда вес спортсмена будет меньше 80 кг, но и вес, который спортсмен будет сбрасывать ежедневно. В этом случае нужно объединить вывод результата и вычисления $n=n-n/100$, $k=k+1$ в тело цикла. Для решения задачи удобно использовать цикл с предусловием, так как количество повторений заранее неизвестно.

3. Блок-схема решения задачи «Тренировка» (рис. 5.3):

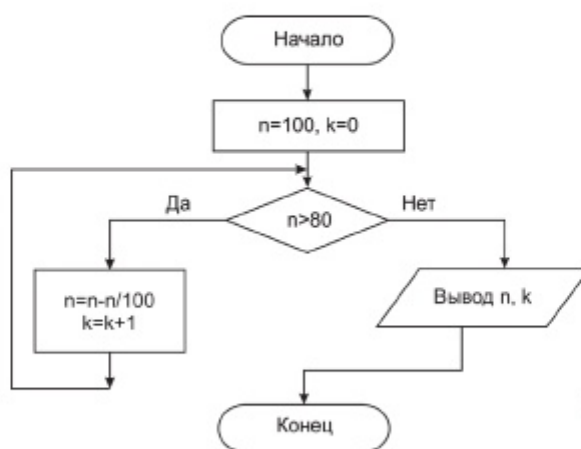
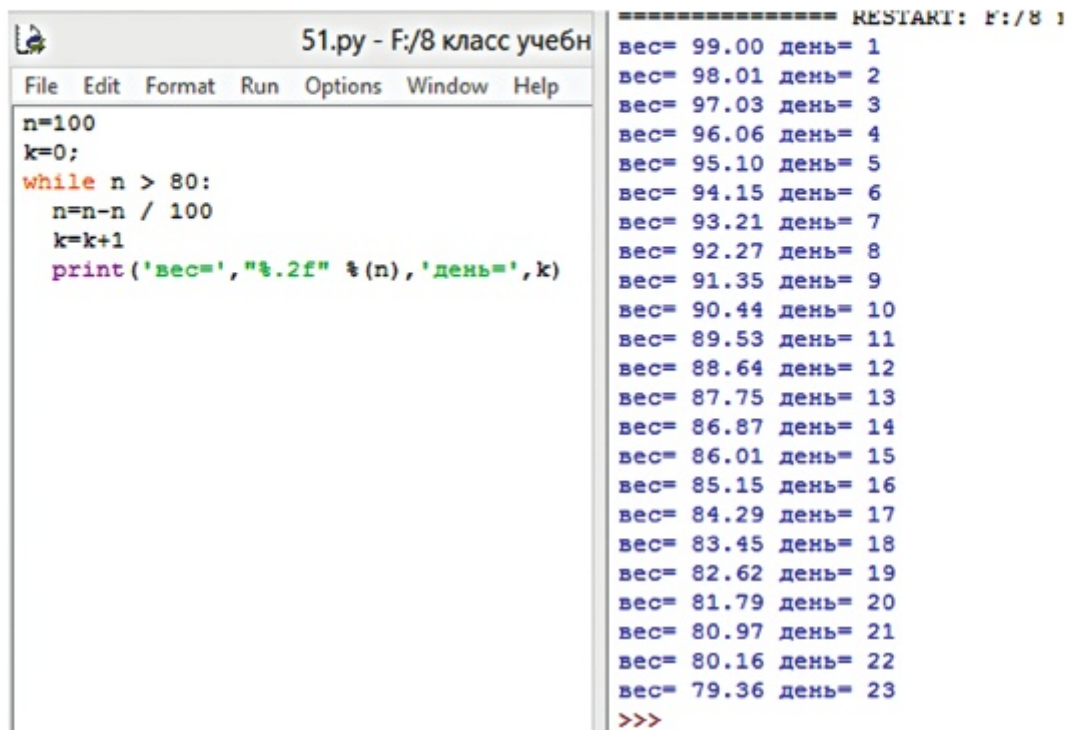


Рис. 5.3. Блок-схема решения задачи

4. Программа решения задачи (рис. 5.4).



```

===== RESTART: F:/8 1
вес= 99.00 день= 1
вес= 98.01 день= 2
вес= 97.03 день= 3
вес= 96.06 день= 4
вес= 95.10 день= 5
вес= 94.15 день= 6
вес= 93.21 день= 7
вес= 92.27 день= 8
вес= 91.35 день= 9
вес= 90.44 день= 10
вес= 89.53 день= 11
вес= 88.64 день= 12
вес= 87.75 день= 13
вес= 86.87 день= 14
вес= 86.01 день= 15
вес= 85.15 день= 16
вес= 84.29 день= 17
вес= 83.45 день= 18
вес= 82.62 день= 19
вес= 81.79 день= 20
вес= 80.97 день= 21
вес= 80.16 день= 22
вес= 79.36 день= 23
>>>

```

Рис. 5.4. Код программы

5. Тестирование и отладка программы. Введем данные и проверим работу программы, показанной на рис. 5.4.

Знание. Понимание



1. Как вы понимаете понятие «постановка проблемы»?
2. Что такое проблема?
3. Перечислите задачи, которые вы решали.
4. Объясните схему решения задач в интегрированной среде разработки на языке Python.
5. На какие вопросы нужно ответить при постановке проблемы?
6. Приведите примеры задач с четко сформулированной идеей.
7. Назовите этапы постановки проблемы.
8. Установите соответствие классификации типов задач.



1	Задачи с практическим содержанием	A	дискретизация непрерывных процессов
		B	организация данных
2	Задачи, для решения которых неизвестны точные методы решения	C	для них известны математические формулы
		D	экономики, экологии, лингвистики и т.д.
		E	расчет площадей
3	Задачи из различных областей деятельности человека	F	записная книжка
		G	физические законы
4	Задачи информационного моделирования	H	актуализация знаний по информатике для решения задач: экологические, шифровка текста и т.д.



Применение. Анализ



9. Спортсмен для подготовки к чемпионату Казахстана по бегу пробегал в день 10 км. Начав тренировки, он увеличивал норму на 10% от нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 30 дней? Какие знания для решения данной задачи вам необходимы? Опишите решение данной задачи, используя все этапы.

10. Составьте разные виды задач:

- с практическим содержанием;
- для решения которых неизвестны точные методы и которые содержат сложные расчеты;
- из различных областей деятельности человека;
- информационного моделирования.

11. К составленным задачам из предыдущего задания напишите постановку задачи без применения компьютера. Составьте математическую модель задачи.



Синтез. Оценивание



12. Подумайте, какие проблемные ситуации сложились в вашем городе, поселке, селе, школе. Какая из них представляет для вас наибольшую актуальность? Опишите ее, выделите основные противоречия и сформулируйте проблему в соответствии с этапами постановки проблемы.

13. Подумайте о возможных способах решения проблемы, выявленной в предыдущем задании.

Сформулируйте тему, цель и задачи проекта; определите целевую группу, участников, партнеров; составьте календарный план проекта, рассчитайте примерный бюджет и т.д.

5.2. РАЗРАБОТКА АЛГОРИТМА

ВЫ НАУЧИТЕСЬ

- Разрабатывать алгоритм решения задачи;
- осуществлять трассировку алгоритма.

КЛЮЧЕВЫЕ СЛОВА

Разработка алгоритма	Блок-схема
Алгоритмді әзірлеу	Блок-сызба
Algorithm development	Flowchart



Какой этап следует за этапом разработки модели?
Что такое алгоритм?



Вам известно, что **алгоритм** – это точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

При разработке алгоритма необходимо помнить об его основных свойствах:

- **понятности** для исполнителя;
- **дискретности** (последовательное выполнение простых шагов);
- **определенности** (четкое, однозначное);
- **результативности** (решение задачи за конечное число шагов);
- **массовости**.

Алгоритм решения задачи разрабатывается в общем виде. Для алгоритмов существуют разные формы представления:

- **словесная** (записи на естественном языке);
- **графическая** (изображения из графических символов);
- **псевдокоды** (полуформализованные описания алгоритмов на условном алгоритмическом языке);
- **программная** (тексты на языках программирования).

При написании алгоритма решения задачи необходимо в первую очередь ответить на вопрос «Как?». Хорошие решения, как правило, имеют следующие характеристики:

- простота;
- документирование;
- построение по модульному принципу. Это означает, что любая часть программы может быть повторно использована или изменена;
- полезные сообщения об ошибках.

Таким образом, прежде чем разработать алгоритм, мы должны выполнить этап постановки проблемы, затем этап постановки задачи.

1. Этап постановки задачи:

- сбор информации о задаче;
- формулировка условия задачи;
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных.

2. Анализ и исследование задачи, модели.

3. Разработка алгоритма.

При разработке алгоритма умение составлять блок-схемы является основной. Например, необходимо вывести на экран таблицу значений функции $y = x^2 + 3x - 4$, изменяя x от **1** до **10** с шагом **1**, и вычислить сумму полученных значений.



Составьте алгоритм и программу вывода таблицы значений (табуляция) функции $y = x^2 + 3x - 4$, изменяя x от **1** до **10** с шагом **1**, и вычислите сумму полученных значений.



Шаг за шагом

1. Постановка проблемы. Пользователь вводит значение x . Необходимо составить алгоритм и программу вывода таблицы значений функции $y = x^2 + 3x - 4$, изменяя x от **1** до **10** с шагом **1**, и вычислить сумму полученных значений.

2. Формализация. Исходными данными (аргументами) задачи являются значения x на интервале **[1; 10]**. Результатами будут значения функции $y = x^2 + 3x - 4$ и сумма полученных значений. Так как количество повторений тела цикла известно заранее, то можно использовать цикл с параметром. Сумму значений s будем вычислять по формуле $s = s + y$.

3. Составим блок-схему (рис. 5.5).

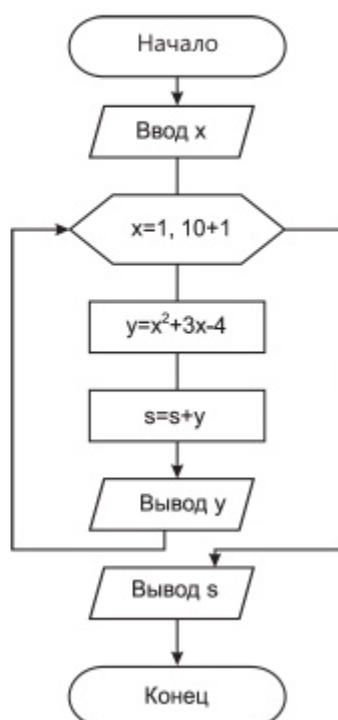


Рис. 5.5. Блок-схема «Табуляция»

4. Составим программу решения задачи.

Код программы показан на рис. 5.6.

<pre> File Edit Format Run Options Window Help #напечатать таблицу значений от 1 до 10 с шагом 1 s=0 for x in range(1, 11): y=x*x+3*x-4 print("при x=", x, "y=", y) s=s+y print ("сумма значений =",s) </pre>	<pre> ===== RESTART: при x= 1 y= 0 при x= 2 y= 6 при x= 3 y= 14 при x= 4 y= 24 при x= 5 y= 36 при x= 6 y= 50 при x= 7 y= 66 при x= 8 y= 84 при x= 9 y= 104 при x= 10 y= 126 сумма значений = 510 >>> </pre>
---	--

Рис. 5.6. Код программы «Табуляция»

5. Тестирование и отладка программы. Запустите программу. Проверьте правильность ее работы.



Знание. Понимание



1. Какие свойства нужно учитывать при составлении алгоритма?
2. Какие формы представления алгоритма вы знаете?
3. Что такое блок-схема? Почему блок-схема является распространенной формой представления алгоритма?
4. Каких этапов нужно придерживаться при решении задач?
5. Почему при разработке алгоритма обязательно проводить трассировку?
6. Назовите формы представления алгоритма и приведите примеры.



Применение. Анализ



7. Составьте алгоритм и программу вывода таблицы значений (табуляция) функции: $y = x^2 - 5x + 1$, изменяя x от -10 до 10 с шагом 2 , и вычислите сумму полученных значений.
8. По блок-схеме, представленной на рисунке 5.7, решите обратную задачу, определите условие задачи. Составьте программу, проведите трассировку в трассировочной таблице.

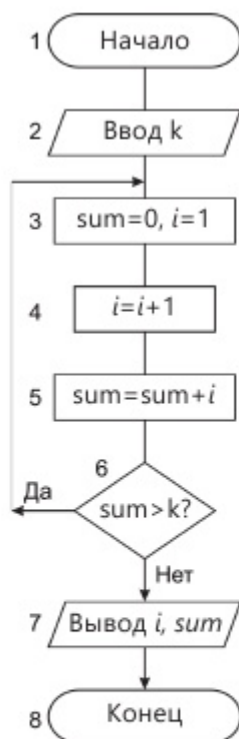


Рис. 5.7. Блок-схема

Синтез. Оценивание

9. Дано целое число n . Выведите все квадраты натуральных чисел, не превосходящие n , в порядке возрастания. Реализуйте задачу, применяя словесный алгоритм. Проведите трассировку программы.



10. Дано целое число не меньше 2. Выведите его наименьший натуральный делитель, отличный от 1. Составьте блок-схему и программу решения задачи. Проведите трассировку программы.



Составление алгоритмов решения задач – это работа творческая. Нет универсального способа, позволяющего без особого труда составлять любые алгоритмы.

Основной целью программирования является построение надежной, легко читаемой и модифицируемой программы, решающей поставленную задачу. Для этого программа должна иметь более простую структуру.

Рассмотрим основные методы и технологии разработки алгоритмов и программ.

Структурное программирование – метод создания простых, понятных и легко читаемых программ.

Процедурное программирование – метод построения программы как совокупности ее функциональных частей: процедур или функций.

Модульное программирование – организация программы в виде совокупности независимых частей (модулей) со строгим порядком их взаимодействия.

Объектно ориентированное программирование – метод программирования, основанный на использовании в программе совокупности объектов, каждый из которых содержит некоторые данные и методы их обработки.

Перечисленные методы реализуют одну из возможных технологий современного программирования: **нисходящую** или **восходящую**.

Нисходящее проектирование – технология разработки программ, при которой на каждом шаге проектирования задача разбивается на более мелкие подзадачи.

Восходящее проектирование – технология разработки программ, при которой сначала проектируются и отлаживаются подпрограммы для выполнения простых операций, после чего они связываются в единую программу.

5.3. ПРОГРАММИРОВАНИЕ АЛГОРИТМА



ВЫ НАУЧИТЕСЬ

- Создавать модели задач в интегрированной среде разработки программ;
- использовать компоненты интегрированной среды разработки программ;
- использовать операторы циклов в интегрированной среде разработки программ.

КЛЮЧЕВЫЕ СЛОВА

Программирование алгоритма	Алгоритмді программалау	Programming the algorithm
Комментарий	Түсініктеме	Comment
Идентификатор	Идентификатор	Identifier



Как создавать модели задач в интегрированной среде разработки программ?

После постановки проблемы и последующей разработки алгоритма переходим к решению задачи с помощью компьютерного моделирования. **Компьютерная модель** – это условный образ объекта в виде программы. Модель отображает структуру и взаимосвязи между элементами объекта.



Компьютерное моделирование – это метод решения задачи анализа или синтеза сложной системы на основе использования ее модели.

Основная задача компьютерного моделирования заключается в получении количественных и качественных результатов на основе имеющейся модели. На основе созданной компьютерной модели проводится серия вычислительных экспериментов в интегрированной среде разработки. Этап исследования свойств объектов или физических процессов с использованием компьютера позволяет найти оптимальные параметры, режимы работы и уточнить модель.

Рассмотрим процесс разработки программы в среде программирования. Для того чтобы выполнить программу на Python, надо пройти несколько этапов.

На рисунке 5.8 приведена упрощенная графическая схема решения задачи в интегрированной среде разработки.

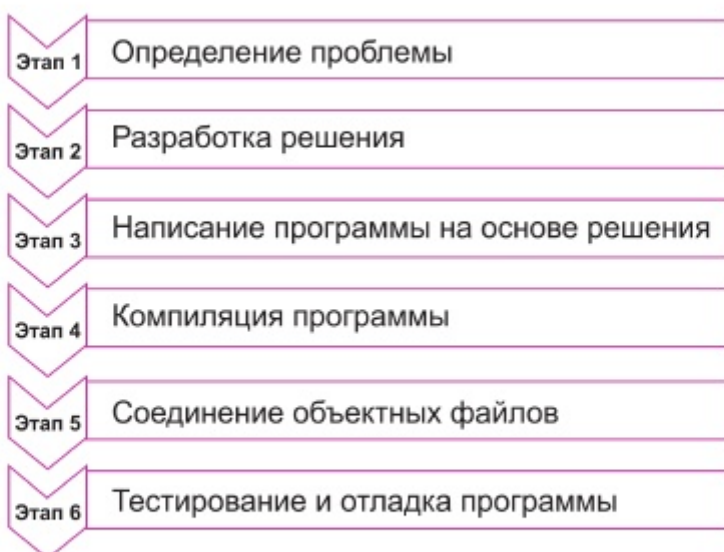


Рис. 5.8. Схема решения задачи в интегрированной среде разработки

Вспомним, что язык программирования Python – это интерпретируемый язык. Это означает, что программа, которая на нем написана, при каждом выполнении сначала переводится в двоичный код, а затем выполняется. На сегодняшний день он является наиболее популярным среди интерпретируемых языков. Программы на интерпретируемых языках выполняются интерпретатором и распространяются в виде исходного кода.

Вы уже знаете, что Python может работать в двух режимах:

- интерактивный;
- пакетный.

В интерактивном режиме Python ожидает ввод команд пользователя (рис. 5.9). В этом режиме интерпретатор немедленно выполняет введенные инструкции и выводит результат, но эти инструкции не сохраняются в файле. Интерактивное исполнение кода необходимо для того, чтобы немедленно протестировать фрагмент написанного кода.

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:54:25) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Рис. 5.9. Интерактивный режим Python

В пакетном режиме, то есть в режиме интерпретации файлов с исходным кодом, происходит запуск на исполнение написанных скриптов или модулей (рис. 5.10).

```
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Рис. 5.10. Пакетный режим Python

Выполнение состоит из следующих этапов (рис. 5.11).

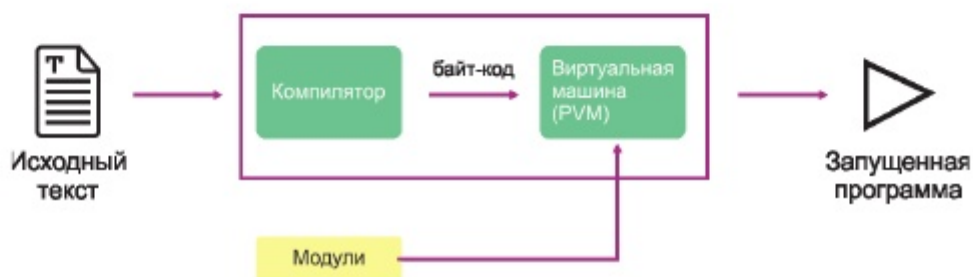


Рис. 5.11. Этапы выполнения программы в интегрированной среде разработки Python

1. Последовательная обработка всех операторов, которые записаны в скрипте.
2. Компиляция исходного кода в байт-код, который представляет собой язык программирования низкого уровня, независимый от платформы и операционной системы. Байт-код необходим для оптимизации процесса выполнения скрипта.
3. Исполнение полученного кода. На этом этапе виртуальная машина Python (PVM) циклично перебирает каждый оператор из скрипта и запускает его на исполнение.

Компиляция в байт-код производится автоматически, а PVM – это часть системы Python, которую вы установили вместе с интерпретатором и компилятором.



Напишите программу «Вклад в банке», по которой можно определить время, за которое можно получить нужный доход.

Шаг за шагом



1. Постановка проблемы. Вклад в банке составляет x тенге. Ежегодно он увеличивается на p процентов, после чего дробная часть тиын отбрасывается. Определите, через сколько лет вклад составит не менее y тенге.

2. Формализация. Программа получает на вход три натуральных числа: x – первоначальная сумма вклада, p – процент по вкладу, y – итоговая сумма вклада. В результате нужно получить целое число – количество лет, через которое будет получена нужная сумма.

Выражение «дробная часть отбрасывается» означает, что если у вас оказалось 123.4567 тенге, т.е. 123 тенге и 45.67 тиын, то после округления у вас получится 123 тенге и 45 тиын, т.е. 123.45 тенге.

Каждый год сумма вклада увеличивается на p процентов, т.е. $x = x + x \cdot p / 100$. Для того чтобы отбросить дробную часть тиын, используем встроенную функцию `int`. $x = \text{int}(100 \cdot x) / 100$.

Алгоритм решения этой задачи является циклическим.

3. Блок-схема (рис. 5.12) определения дохода по вкладу:

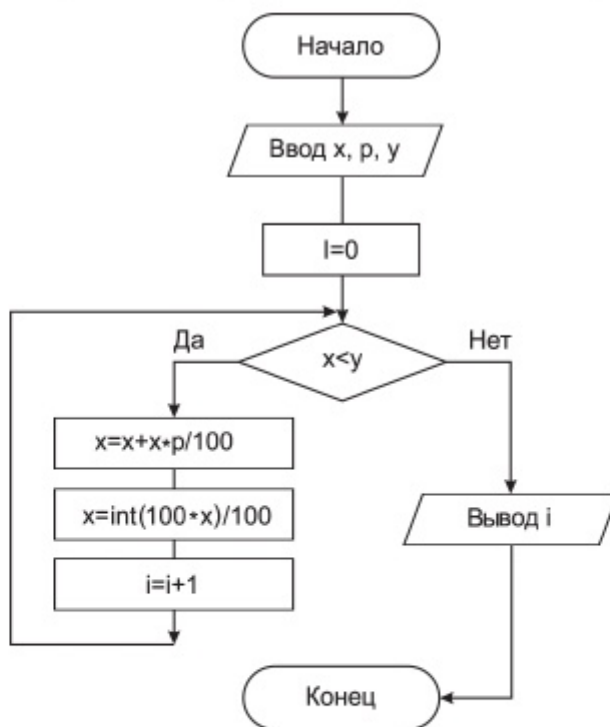


Рис. 5.12. Блок-схема «Вклад в банке»

4. Код программы решения задачи (рис. 5.13).

```

File Edit Format Run Options Window Help
x = int(input("введите первоначальную сумму вклада"))
p = int(input("введите процент по вкладу"))
y = int(input("введите итоговую сумму"))
i = 0
while x < y:
    x *= 1 + p / 100
    x = int(100 * x) / 100
    i += 1
print("вам понадобится", i, "лет")

Type "help", "copyright", "credits" or ">>>"
>>>
===== RESTART: F:/8 класс учеб
введите первоначальную сумму вклада5000
введите процент по вкладу15
введите итоговую сумму15000
вам понадобится 8 лет
>>>
  
```

Рис. 5.13. Блок-схема «Вклад в банке»

5. Тестирование и отладка программы.

Проведите тестирование программы.



Знание. Понимание



1. Для чего применяется компьютерное моделирование?
2. Приведите упрощенную схему решения задачи в интегрированной среде разработки.



3. Для чего нужен интерпретатор?
4. В каких режимах может работать Python?
5. Перечислите этапы выполнения программы в интегрированной среде разработки Python.
6. Установите правильный порядок этапов решения задачи в интегрированной среде разработки.

Этап 1	Разработка решения
Этап 2	Тестирование и отладка программы
Этап 3	Определение проблемы
Этап 4	Соединение объектных файлов
Этап 5	Компиляция программы
Этап 6	Написание программы на основе решения



Применение. Анализ



7. Заполните алгоритмические кроссворды: придумайте задания для приведенных блок-схем (рис. 5.14). Напишите к ним программы. Сделайте трассировку программ.

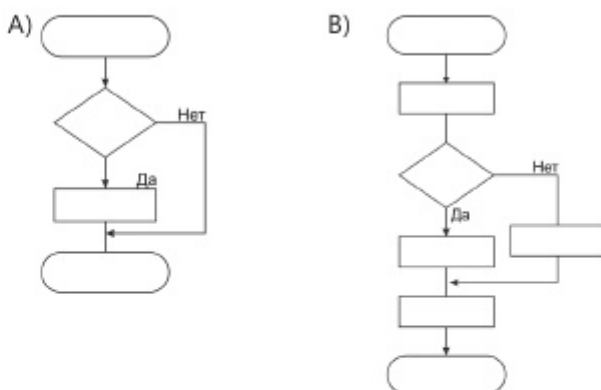


Рис. 5.14. Алгоритмические кроссворды

8. Последовательность состоит из натуральных чисел и завершается числом **0**. Определите значение наибольшего элемента последовательности. Составьте программу решения этой задачи. Ниже приведен алгоритм ее решения на алгоритмическом языке:

алг макс_последовательность

нач цел *max*, *element*

max=0

element=-1

нц пока *element*≠0

ввод *element*

если *element*>*max*

max=*element*

кц

вывод *max*

кон

Синтез. Оценивание



9. На рисунке 5.15 дан программный код. Составьте условие задачи и блок-схему.



```
File Edit Format Run Options
i=1
while i<=10:
    f=5*i
    print (f)
    i=i+2
```

Рис. 5.15. Программный код

10. По заданному числу n вычислите сумму факториалов: $1! + 2! + 3! + \dots + n!$ Решите задачу в соответствии со всеми этапами в интегрированной среде разработки.

Рекомендации для выполнения задания

Для определения постановки задачи разберите каждый оператор. Проведите трассировку программы.



Интересные задачи

1. **«Переворот числа».** Вводится число. Преобразовать его в другое число, цифры которого будут следовать в обратном порядке по сравнению с введенным числом.

2. **«Дано число».** Найти сумму и произведение его цифр.

3. **«Числа Фибоначчи».** Вывести на экран ряд чисел Фибоначчи, состоящий из n элементов.

Числа Фибоначчи – это элементы числовой последовательности 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ..., в которой каждое последующее число равно сумме двух предыдущих.

4. **«Вывод квадратов натуральных чисел».** Вывести все квадраты натуральных чисел, не превосходящие данного числа n .

Пример: $n=50$ | 1 4 9 16 25 36 49

5. Написать программу, подсчитывающую количество четных и нечетных цифр в числе.

6. Кубы чисел от a до b . Вывести на экран кубы чисел от a до b , которые вводит пользователь.

7. **«Вывод на экран таблицы ASCII».** ASCII – код для обмена информацией. В настоящее время чаще используется 8-битный вариант (под каждый символ выделяется 1 байт памяти). Проще говоря, получается таблица из 256 символов, соответствующих числам от 0 до 255.

8. Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, сколько амёб будет через 3, 6, 9, 12, ... 24 часа.

9. Составить программу, которая печатает таблицу умножения и сложения натуральных чисел в десятичной системе счисления.

10. Составить программу, которая печатает таблицу умножения и сложения натуральных чисел в шестнадцатеричной системе счисления.

5.4. ТЕСТИРОВАНИЕ ПРОГРАММЫ

ВЫ НАУЧИТЕСЬ

- Создавать модели задач в интегрированной среде разработки программ;
- осуществлять трассировку алгоритма.

КЛЮЧЕВЫЕ СЛОВА

Тестирование	Тестілеу	Testing
Отладка	Түзету	Debugging
Трассировка	Трассирлеу	Trace



Что такое тестирование программы?
Для чего выполняют тестирование программы?



Необходимым условием для проверки правильности выполнения программы является проведение тестирования. На тестах проверяется реализация программы запланированного алгоритма.

Тестирование программы – это процесс выполнения программы с целью обнаружения ошибки в программе на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор данных называется **тестовым** или просто **тестом**.



Процесс поиска и исправления ошибок называется **отладкой**.



Отладку можно представить в виде многократного повторения трех процессов:

1. На первом проводится тестирование программы и обнаружение ошибки.
2. На втором – поиск места ошибки в программе и документации.
3. На третьем – редактирование программы и документации с целью устранения обнаруженной ошибки.

К отладке нужно подходить ответственно. Ведь именно с помощью отладки отыскиваются и устраняются только ошибки, обнаруженные при тестировании.

Для тестирования программы необходимо заранее подготовить набор тестов. Эти тесты должны при заданном интервале времени, отведенном на тестирование, обнаруживать большее число ошибок.

Таким образом, при тестировании и отладке можно столкнуться с различными типами ошибок (рис. 5.16).



Рис. 5.16. Схема тестирования и отладки

Синтаксическими являются ошибки, которые нарушают синтаксис и пунктуацию языка. Например, если вы дали имя переменной, начиная с цифры, или забыли поставить двоеточие в заголовке сложной инструкции. Интерпретатор Python, встретив ошибочное выражение, останавливает выполнение программы и выводит соответствующее сообщение, указав на место возникновения ошибки (рис. 5.17).

```

>>> 2a=5
SyntaxError: invalid syntax
>>>
  
```

Рис. 5.17. Синтаксическая ошибка

В Python синтаксические ошибки относятся к ошибкам, а все остальные ошибки – к **исключениям**. Нарушение семантики означает, что даже если выражения написаны верно с точки зрения синтаксиса языка, программа не работает так, как от нее ожидалось.

Например, если происходит обращение к переменной, которой не было присвоено значение, в случае Python это означает, что переменная вообще не была объявлена, она не существует, и возникнет исключение **NameError** (рис. 5.18).

```

>>> x=1
>>> print (x+y)
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    print (x+y)
NameError: name 'y' is not defined
  
```

Рис. 5.18. Исключение NameError

Если при написании программы вы ошиблись с типами величин, то возникают исключения **ValueError** (ошибка значения) и **TypeError** (ошибка типа).

На рис. 5.19 показан пример, когда возникает исключение **ValueError**, потому что функция **int()** не может преобразовать строку **"Python"** к целому числу.

```
>>> int("Python")
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    int("Python")
ValueError: invalid literal for int() with base 10: 'Python'
^^^
```

Рис. 5.19. Исключение **ValueError**

В следующем примере (рис. 5.20) число **5** и строка **"25"** принадлежат разным типам, операнд сложения между которыми не поддерживается. При попытке их сложить возникает исключение **TypeError**.

```
>>> 5+"25"
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    5+"25"
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>>
```

Рис. 5.20. Исключение **TypeError**

Деление на ноль вызывает исключение **ZeroDivisionError** (рис. 5.21).

```
>>> 500/0
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    500/0
ZeroDivisionError: division by zero
>>>
```

Рис. 5.21. Исключение **ZeroDivisionError**

К логическим ошибкам можно отнести неправильную постановку задачи, неполный учет всех ситуаций, которые могут возникнуть, а также неверный алгоритм. Ошибки могут быть допущены на всех этапах решения задачи – от ее постановки до оформления. Поэтому очень важно провести тестовые расчеты и анализ результатов. Разновидности ошибок и соответствующие примеры приведены в таблице 5.1.

Таблица 5.1

Виды ошибок

Вид ошибки	Пример	Этап обнаружения
Синтаксические ошибки	Нарушение правил, определяемых языком программирования	Во время отладки на этапе компиляции программы
	Опечатки: перепутаны близкие по написанию символы, например цифра 1 и буквы l	
Ошибки времени исполнения (исключения)	Ошибки при выполнении операций: слишком большое число, деление на ноль, извлечение квадратного корня из отрицательного числа и т.п.	При первых запусках программы во время тестирования
	Ошибки в данных: неудачное определение возможного диапазона изменения данных	
	Ошибки ввода-вывода данных: неверное считывание входных данных, неверное задание форматов данных	
Алгоритмические (логические) ошибки	Ошибки анализа: неправильная постановка задачи, неполный учет всех ситуаций, которые могут возникнуть	Во время полномасштабного тестирования
	Семантические ошибки: непонимание порядка выполнения оператора, неверный алгоритм	



Даны два целых числа **a** и **b**. Выведите все числа от **a** до **b** включительно в порядке возрастания, если $a < b$, или в порядке убывания в противном случае.



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, если число **a** меньше **b**, то вывести все числа от **a** до **b** в порядке возрастания. Если число **a** больше **b**, то нужно вывести все числа от **a** до **b** в порядке убывания.

2. Формализация. Вводим два целых числа **a** и **b**. Если $a < b$, то, используя цикл с параметром, выводим все числа в промежутке от **a** до **b** с шагом **1**. Иначе выводим все числа в промежутке от **a** до **b** с шагом **-1**.

3. Блок-схема «Вывод чисел» показана на рисунке 5.22.

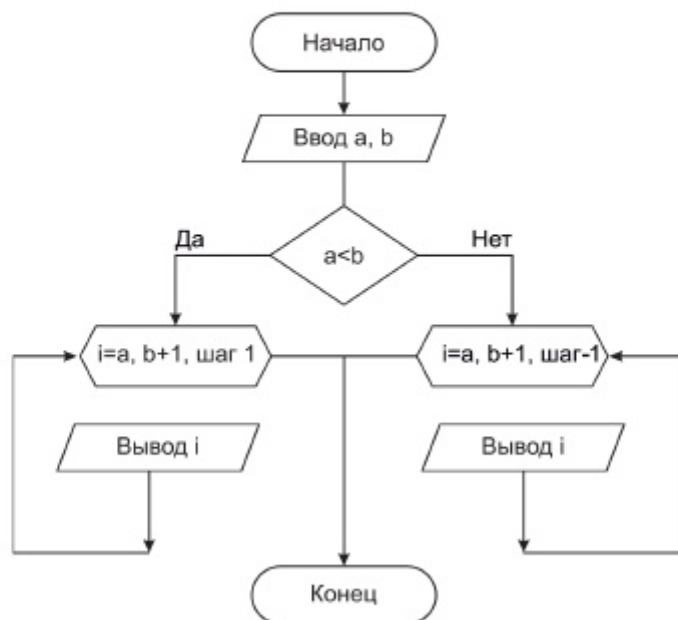


Рис. 5.22. Блок-схема «Вывод чисел»

4. Код программы решения задачи показан на рисунке 5.23.

```

File Edit Format Run Options Window Help
a = int(input())
b = int(input())
if a < b:
    for i in range(a, b + 1):
        print(i)
else:
    for i in range(a, b - 1, -1):
        print(i)
  
```

Рис. 5.23. Код программы «Вывод чисел»

5. Тестирование и отладка программы.

Необходимо ввести значения переменных при $a > b$, $a < b$, $a = b$ (см. табл. 5.2–5.4).

Таблица 5.2

Трассировочная таблица № 1

№	Команда алгоритма	Переменные			Выполняемые действия
		a	b	i	
1	Ввод a, b	5	8		
2	$a < b$				$5 < 8$ (true)
3	for i in range(a, b + 1): print(i)			5	
4	print(i)			6	
5	print(i)			7	
6	print(i)			8	

Таблица 5.3

Трассировочная таблица № 2

№	Команда алгоритма	Переменные			Выполняемые действия
		a	b	i	
1	Ввод a, b	8	5		
2	$a < b$				$8 < 5$ (false)
3	for i in range(a, b - 1, -1): print(i)			8	
4	print(i)			7	
5	print(i)			6	
6	print(i)			5	

Таблица 5.4

Трассировочная таблица № 3

№	Команда алгоритма	Переменные			Выполняемые действия
		a	b	i	
1	Ввод a, b	5	5		
2	$a < b$				$5 < 5$ (false)
3	for i in range(a, b - 1, -1): print(i)			5	

Знание. Понимание

1. Какова цель тестирования и отладки программы?
2. Какие типы ошибок существуют?
3. Приведите пример синтаксических ошибок и исключений.
4. Что входит в схему тестирования и отладки?
5. Если в программе есть логическая ошибка, то выполнится ли программа?

**Применение. Анализ**

6. С клавиатуры вводятся числа, последнее число **0**. Найдите сумму этих чисел. Какие ошибки допущены в программе? Устраните ошибки. Введите программный код в редактор. Откомпилируйте и выполните программу. Проверьте правильность выполнения программы с помощью приведенных ниже тестов.



Ввод	Вывод
1 2 3 4 5 6 7 8 9 0	55
100500 100500 0	201000

```
sum = 1
a = 0
while a!=0:
    a=int(input())
    sum = sum + a
while a = 0:
print(a)
```

7. Даны два типа циклов. Какой цикл будет выполняться быстрее? Каким образом это можно узнать?

```
i = 0
a=10
while i < a:
    print(i)
    i = i + 1
```

```
for l in range(1,10):
    print(i)
```

8. Выявите наиболее существенные признаки следующих понятий:
 - тестирование;
 - отладка;

- правила тестирования;
- принципы составления тестов;
- методы тестирования;
- способы отладки;
- виды ошибок.



Синтез. Оценивание



9. На рисунке 5.24 приведен кроссворд со словами. Выполните обратную задачу: напишите определения к данным словам (по горизонтали и вертикали).

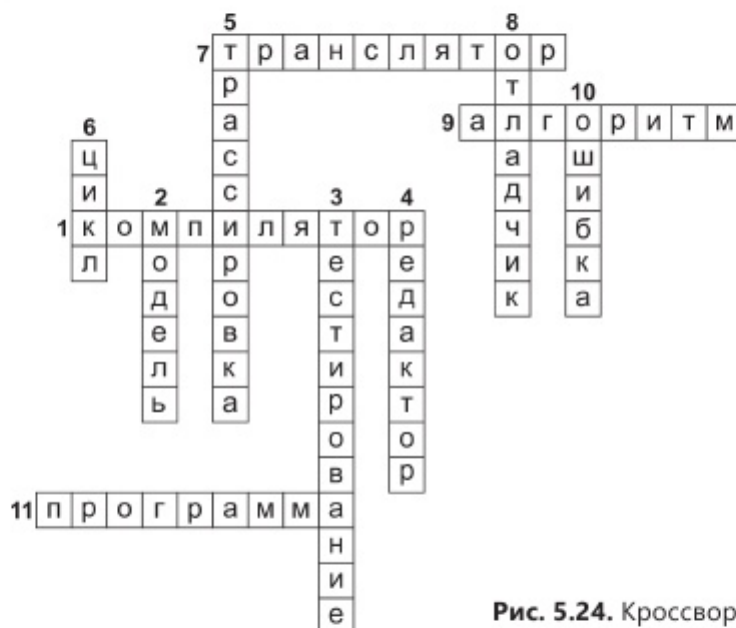


Рис. 5.24. Кроссворд

10. Проведите исследование по выбранной вами идее проекта. Выполните следующие этапы:

- 1) Постановка проблемы.
- 2) Постановка задачи.
- 3) Создание математической модели.
- 4) Разработка алгоритма.
- 5) Разработка программы.
- 6) Отладка и тестирование.
- 7) Компиляция и выполнение.
- 8) Документация: к алгоритму, программе.
- 9) Презентация проекта.
- 10) Защита проекта.

Тестовые задания к разделу V

1. Последовательность действий, выполнение которых приводит к конечному результату, называется:

- A) указанием;
- B) нет верного ответа;
- C) алгоритмом;
- D) циклом;
- E) ветвлением.

2. Расставьте в правильной последовательности «Этапы решения задач с помощью компьютера».

- 1) Построение алгоритма.
 - 2) Формализация. Моделирование.
 - 3) Программирование.
 - 4) Постановка проблемы.
 - 5) Отладка и тестирование программы.
- A) 1, 2, 3, 4, 5.
 - B) 4, 2, 1, 3, 5.
 - C) 4, 1, 3, 2, 5.
 - E) 2, 1, 4, 3, 5.

3. Сколько раз будет выполнено тело цикла?

```
for i in range (1, 10, 2):
    <тело цикла>
```

- A) 1.
- B) 3.
- C) 5.
- D) 7.
- E) 10.

4. Каким будет значение i после исполнения инструкций (рис. 5.25)?

- A) 0.
- B) 1.
- C) 5.
- D) 0.625.
- E) 0,5.

```
File Edit Format
i, n = 0, 625
while n > 0:
    n = n // 5
    i = i + 1
print(i)
```

Рис. 5.25. Инструкции

5. Дан код программы в Python (рис. 5.26). Определите, что будет выведено на печать.

- A) «Найдено».
- B) «Готово».
- C) Ошибка в коде.
- D) «Найдено» и «Готово».
- E) Выход из цикла.

```
File Edit Format Run Options Window
for j in 'Привет! \\'Как дела? ':
    if j == '\\':
        print("Найдено")
        break
else:
    print("Готово")
```

Рис. 5.26. Код программы 1

6. Что будет выведено на печать этим кодом программы (рис. 5.27)?

- A) Ошибка, так как *i* не присвоено число.
- B) Числа: 1, 3 и 5.
- C) Числа: 0, 2 и 4.
- D) Числа: 1 и 3.
- E) Число 5 и 2.

```
File Edit Format Run Options
for i in range(5):
    if i % 2 == 0:
        continue
    print(i)
```

Рис. 5.27. Код программы 2

7. Алгоритм называется циклическим, если:

- A) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- B) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- C) ход его выполнения зависит от истинности тех или иных условий;
- D) он включает в себя вспомогательный алгоритм;
- E) выполнение его действий приводит к конечному результату.

8. Каким будет максимальное значение *i* при исполнении цикла?

```
for i in range (1, 5):
    <тело цикла>
```

- A) 1.
- B) 2.
- C) 3.
- D) 4.
- E) 5.

9. Цикл с предусловием (while) выполняется так:

- A) выполняется тело цикла, изменяется параметр цикла, проверяется условие продолжения выполнения цикла;

В) изменяется параметр цикла, проверяется условие продолжения выполнения цикла, выполняется тело цикла;

С) проверяется условие продолжения выполнения цикла, выполняется тело цикла, изменяется параметр цикла;

Д) тело цикла выполняется n раз (n – натуральное);

Е) определяется, сколько раз должен быть выполнен цикл, и далее цикл с предусловием сводится к циклу с параметром.

10. Цикл с постусловием (while True) выполняется так:

А) выполняется тело цикла, изменяется параметр цикла, проверяется условие продолжения выполнения цикла;

В) изменяется параметр цикла, проверяется условие продолжения выполнения цикла, выполняется тело цикла;

С) проверяется условие продолжения выполнения цикла, выполняется тело цикла, изменяется параметр цикла;

Д) тело цикла выполняется n раз (n – натуральное);

Е) определяется, сколько раз должен быть выполнен цикл, и далее цикл с предусловием сводится к циклу с параметром.

11. Какое число будет выведено на экран в результате выполнения цикла, показанного на рис. 5.28?

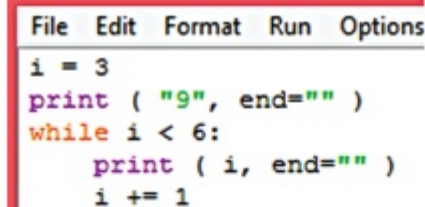
А) 3861.

В) 9345.

С) 9123.

Д) 1239.

Е) 3939.



```
File Edit Format Run Options
i = 3
print ( "9", end="" )
while i < 6:
    print ( i, end="" )
    i += 1
```

Рис. 5.28. Цикл

12. Расставьте в правильном порядке этапы постановки проблемы.

1) Обоснование проблемы.

2) Структурирование проблемы.

3) Формулирование проблемы.

4) Оценка проблемы.

А) 1, 2, 3, 4.

В) 3, 4, 1, 2.

С) 1, 4, 3, 2.

Д) 2, 1, 3, 4.

Е) 3, 1, 2, 4.

13. Что будет выведено кодом программы на печать (рис. 5.29), если ввести число 15?

- A) Сумма: 5, умножение: 5.
- B) Сумма: 8, умножение: 7.
- C) Сумма: 6, умножение: 5.
- D) Сумма: 9, умножение: 4.
- E) Сумма: 7, умножение: 8.

```
File Edit Format Run Options Window Help
n = input('Введите целое число: ')
sum = 0
mult = 1
for i in n:
    sum += int(i)
    mult *= int(i)
print('сумма=', sum)
print('умножение=', mult)
```

Рис. 5.29. Код программы 3

14. Заполните таблицу трассировки для кода программы, приведенного в таблице 5.5.

Таблица 5.5. Таблица трассировки

Операторы	Условие	Переменные			Вывод на экран
		k	s	x	
n = 7					
s = 0					
x = 0					
for k in range(3, n + 1):					
s = s + k					
if s%k == 0:					
print(s)					
else:					
x = x + 1					

15. Дан код программы (рис. 5.30). Выполните трассировку программы, составив таблицу. Что получится на экране, если ввести пять чисел: 11, 22, 33, 44, 55?

```
File Edit Format Run Options Window Help
print('Введите числа: 11, 22, 33, 44, 55')
s=0; p=1
for i in range (5):
    a=int(input())
    s=s+a
    p=p*a
print('Сумма введенных чисел s=', s)
print('Произведение введенных чисел', p)
```

Рис. 5.30. Код программы 4

ГЛОССАРИЙ

DoS-атака – атака на веб-сервер или другую компьютерную систему для вывода ее из строя. При этом создаются условия, затрудняющие доступ пользователей к сайту.

DDoS (*Distributed Denial of Service*) – распределенная атака на отказ в обслуживании. Это одновременная атака с большого количества компьютеров.

Абсолютная ссылка – ссылка, которая позволяет зафиксировать строку или столбец (или строку и столбец одновременно), на которые должна ссылаться формула.

Аккаунт – учетная запись пользователя, то есть набор личных данных, хранящихся на каком-либо сайте или интернет-сервисе.

Алгоритм – точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Алфавит – множество используемых символов в языке.

Бот – автономное программное обеспечение, которое может состоять из вирусов, брандмауэров, программ для удаленного управления компьютером, а также инструментов для скрытия от операционной системы.

Ботнет (англ. *botnet*) – компьютерная сеть, состоящая из некоторого количества хостов с ботами.

Брандмауэр – программно-аппаратный комплекс, контролирующий сетевые пакеты или установленные программы. Применяется для защиты системы от компьютерных атак.

Величина 2^n – объем адресного пространства процессора, то есть объем оперативной памяти, физически адресуемой процессором.

Вложенные циклы – когда в теле одного цикла (внешнего) размещается другой цикл (внутренний). Внешняя часть вложенного цикла выполняется только после того, как завершит работу внутренняя часть.

Восходящее проектирование – технология разработки программ, при которой сначала проектируются и отлаживаются подпрограммы для выполнения простых операций, после чего они связываются в единую программу.

Входящий трафик – поток информации, получаемой из сети Интернет.

Гистограмма – диаграмма, в которой отдельные значения представлены вертикальными столбцами различной высоты.

Грифинг – преследование других игроков в многопользовательских онлайн-играх.

Группировка данных – объединение диапазона ячеек.

Диаграмма – форма графического представления числовых значений, которая позволяет облегчить интерпретацию числовых данных.

Интернет-трафик (англ. *Traffic* – движение) – объем информации, передаваемой через компьютерную сеть за определенный период времени.

Инфографика – способ подачи информации, данных и знаний с помощью графического изображения.

Информационная безопасность – сохранение и защита информации, в том числе системы и оборудования, которые предназначены для применения, сохранения и передачи этой информации.

Исходящий трафик – отправляемые данные (текстовая, графическая, видео и т.д.) с использованием сети Интернет.

Итерация цикла – один проход этого цикла.

Зависимая ячейка – любая ячейка, которая содержит формулу.

Канал связи (англ. *channel, data line*) – система технических средств и среда распространения сигналов для односторонней передачи данных (информации) от отправителя (источника) к получателю (приёмнику).

Кибербуллинг – травля, оскорбления или угрозы, высказываемые жертве с помощью средств электронной коммуникации, в частности, сообщений в социальных сетях, мгновенных сообщений, электронных писем и СМС.

Компьютерное моделирование – метод решения задачи анализа или синтеза сложной системы на основе использования ее модели.

Консолидация данных – объединение нескольких диапазонов данных в один диапазон.

Константа – число либо текст. Она не меняется в процессе расчетов.

Конфиденциальная информация – информация, являющаяся конфиденциальной, то есть доверительной, не подлежащей огласке, секретной.

Круговая диаграмма – диаграмма, которая представлена в виде круга, разбитого на секторы, и в которой допускается только один ряд данных.

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

Линейчатая диаграмма – диаграмма, в которой отдельные значения представлены вертикальными столбиками различной высоты.

Логические ошибки – ошибки в логике, их труднее всего обнаружить.

Модульное программирование – организация программы в виде совокупности независимых частей (модулей) со строгим порядком их взаимодействия.

Мощность алфавита (N) – количество символов в алфавите (его размер).

Объектно ориентированное программирование – метод программирования, основанный на использовании в программе совокупности объектов, каждый из которых содержит некоторые данные и методы их обработки.

Нисходящее проектирование – технология разработки программ, при которой на каждом шаге проектирования задача разбивается на более мелкие подзадачи.

Отладка – процесс поиска и исправления ошибок в программе, препятствующих корректной работе программы.

Относительная ссылка – обозначение ячейки, составленное из номера столбца и строки.

Ошибки по времени можно проверить, когда программа запущена. Например, деление на 0 или тип данных не соответствует.

Параметр цикла – переменная, которая принимает новые значения при каждом повторении цикла.

Прикладная задача – задача, поставленная вне информатики, но решаемая методами использования информационно-коммуникационных технологий.

Проблема – форма научного отображения проблемной ситуации. Она формулируется как выражение необходимости изучения и практических действий, направленных на выявление причин и на их разрешение.

Пропускная способность – реальная полоса пропускания, измеренная в определенное время.

Процессор (центральный процессор, *Central Processing Unit*) – микросхема, обрабатывающая информацию и координирующая работу всех устройств компьютера.

Процедурное программирование – метод построения программы как совокупности ее функциональных частей: процедур или функций.

Разрядность процессора – число одновременно обрабатываемых процессором битов. Поэтому процессор может быть 8-, 16-, 32-, 64-разрядным.

Регистр – ячейка процессора, в которой хранится машинное слово. Машинное слово представляет собой некоторое число (или команду), записанное в двоичном коде.

Сводная таблица – отчет, построенный на основе табличных данных.

Синтаксические ошибки – неправильно написанные ключевые слова, пропущенные скобки или неправильно расставленные пробелы. Их можно легко обнаружить, так как они выделяются красным цветом.

Скорость передачи информации – количество информации, передаваемое за единицу времени.

Спам – рассылка навязчивой, нежелательной рекламной информации.

Спарклайны – отслеживание тенденции в рядах данных (колебание цен, объем продаж).

Ссылка – указание на адрес ячейки.

Статистическая таблица – система, построенная из горизонтальных строк и вертикальных столбцов.

Структурное программирование – метод создания простых, понятных и легко читаемых программ.

Табулирование функции – вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения с определенным шагом.

Тактовая частота – количество тактов (операций), выполненных процессором в секунду.

Тело цикла состоит из одной или нескольких инструкций, записанных с отступом одинаковой величины.

Тестирование программы – процесс выполнения программы с целью обнаружения ошибки в программе на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ.

Трассировка – процесс выполнения программы по шагам (*step-by-step*), инструкция за инструкцией.

Троллинг – провокационное поведение с атакой на жертву.

Фишинг – сетевое мошенничество.

Флейминг – бесцельная дискуссия в чате, личной переписке или комментариях с негативными эмоциями.

Функция – встроенный инструмент Excel, способный вернуть значение в зависимости от переданного ему параметра. Она предназначена для расчетов, вычислений и анализа данных.

Хейтеры – люди, для которых главным жизненным ощущением является неприязнь, ненависть.

Хейтинг – полноценная травля в виде негативных комментариев, сообщений в адрес конкретного человека или явления.

Хейт-спич – публичное высказывание, выражающее ненависть к определенному человеку, группе людей или явлению.

Цикл – любая многократно исполняемая последовательность инструкций, пока действует заданное условие.

Циклический процесс – вычислительный процесс, предусматривающий многократное повторение одного и того же действия или одних и тех же операций при различных исходных значениях.

Циклическая программа – программа, реализующая циклический процесс.

Шаг цикла – значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе.

Электромагнитное излучение – это электромагнитные волны, которые возбуждаются разными излучающими объектами.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. *Банкрашков А. В.* Программирование для детей на языке Python. – М.: АСТ, 2018. – 288 с.
2. *Босова Л., Босова А., Коломенская Ю. Г.* Занимательные задачи по информатике. – М.: Издательство: Бином. Лаборатория знаний, 2013.
3. *Вордерман К.* Программирование на Python. Иллюстрированное руководство для детей / К. Вордерман, К. Стили, К. Квигли. – М.: Манн, Иванов и Фербер, 2017. – 346 с.
4. *Горячев А.В.* Практикум по информационным технологиям / А. В. Горячев, Ю. А. Шафрин. – М.: Бином, 2016. – 272 с.
5. *Златопольский Д. М.* 1700 заданий по Microsoft Excel. – СПб: БХВ-Петербург, 2003.
6. *Информатика: Энциклопедический словарь для начинающих* / ред. Д. А. Поспелов. – М.: Педагогика-Пресс, 2013. – 352 с.
7. *Леонтьев В.П.* Новейшая энциклопедия персонального компьютера 2008. – М.: ОЛМА Медиа Групп, 2008. – 960 с.
8. *Леонтьев В.* Персональный компьютер. – М.: ЗАО «Олма Медиа Групп», 2008. – 800 с.
9. *МакГрат М.* Программирование на Python для начинающих. – М.: Эксмо, 2015. – 192 с.
10. *Молодцов В. А., Рыжикова Н. Б., Головки Т. Г.* Репетитор по информатике. – Ростов-на-Дону, 2004.
11. *Мухамбетжанова С. Т., Тен А. С., Даутова Т. К., Сагымбаева А. Е.* Информатика: учебник для 7 класса общеобразовательных школ. – Алматы: Атамұра, 2015.
12. *Основы компьютерных сетей. Учебное пособие.* – Алматы: Microsoft Corporation, 2008.
13. *Попов А.* Excel: Практическое руководство. – М., 2000.
14. *Петроченков А. В.* Персональный компьютер – просто и ясно! – М.: Смоленск: Русич, 2013. – 400 с.
15. *Саммерфилд М.* Программирование на Python 3. Подробное руководство. – М.: Символ, 2016. – 608 с.
16. *Семакин И. Г., Залогова Л. А., Русаков С. В., Шестакова Л. В.* Информатика и ИКТ. 8 класс. – М.: Бином. Лаборатория знаний, 2012.
17. *Семакин И.* Информатика. Базовый курс. 7–9 классы / И. Семакин, Л. Залогова, С. Русаков. – М.: Бином. Лаборатория знаний, 2015. – 390 с.
18. *Семакин И. Г., Вараскин Г. С.* Структурированный конспект базового курса. Приложение к учебнику: Информатика. Базовый курс 7–9 классы. – М.: Лаборатория базовых знаний, 2001.

ИНТЕРНЕТ-РЕСУРСЫ

1. <https://ru.wikipedia.org/wiki>.
2. <http://schools.keldysh.ru/sch444/MUSEUM/>.
3. <http://www.computer-museum.ru/index.php>.
4. <http://school-collection.edu.ru>.
5. *Шауцуква Л.З.* Информатика 10–11. – М.: Просвещение, 2000 г. [<http://book.kbsu.ru/theory/index.html>]
6. *Алексеев Е. Г.* Информатика. Мультимедийный электронный учебник / Е. Г. Алексеев, С. Д. Богатырев [<http://inf.e-alekseev.ru/text/toc.html>]
7. Словарь терминов по информатике [http://uchu2008.narod.ru/razdely/informatika/slovar_terminov.html]
8. Словарь основных понятий и терминов [<http://book.kbsu.ru/theory/definition.html>]
9. Словарь [<http://pandia.ru/text/78/419/69850.php>]
10. Словарь терминов по информатике [http://wiki.iteach.ru/images/5/5dСловарь_терминов.pdf]
11. <https://kopilkaurokov.ru/informatika/uroki/modielirovaniie-v-excel>.

Учебное издание

**Мухамбетжанова Сауле Талапеденовна
Тен Алира Сунтаковна
Демидова Лариса Геннадьевна**

ИНФОРМАТИКА

Учебник для 8 класса общеобразовательной школы

Зав. редакцией *Н. Жиенгалиев*

Редактор *Л. Холина*

Художественные редакторы *А. Айдархан, З. Огай, А. Лукманов*

Технический редактор *О. Рысалиева*

Компьютерная верстка *Н. Развинавичене*

ИБ № 036

Сдано в набор 21.05.2021. Подписано в печать 23.07.2021. Формат 70x90 1/16⁶
Бумага офсетная. Гарнитура «Сегое». Печать офсетная. Усл.печ.л. 12,87.
Уч.-изд.л. 8,72. Тираж 7000 экз. Заказ № 6001.

ТОО «Корпорация «Атамұра», 050000, г. Алматы, пр. Абылай хана, 75.

Полиграфкомбинат ТОО «Корпорация «Атамұра», Республика Казахстан,
050002, г. Алматы, ул. М. Макатаева, 41

