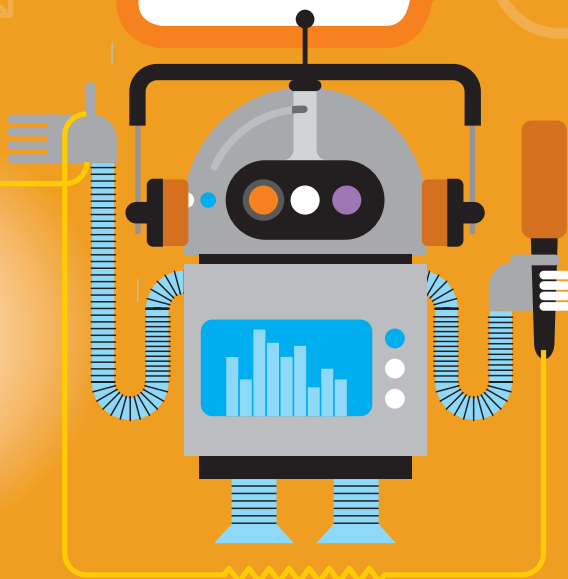


ИНФОРМАТИКА

9



1001





AZƏRBAYCAN RESPUBLİKASININ DÖVLƏT HİMNİ

Musiqisi *Üzeyir Hacıbəylinin,*
sözləri *Əhməd Cavadındır.*

Azərbaycan! Azərbaycan!
Ey qəhrəman övladın şanlı Vətəni!
Səndən ötrü can verməyə cümlə hazırız!
Səndən ötrü qan tökməyə cümlə qadiriz!
Üçrəngli bayrağınla məsud yaşa!

Minlərlə can qurban oldu,
Sinən hər bə meydan oldu!
Hüququndan keçən əsgər,
Hərə bir qəhrəman oldu!

Sən olasan gülüstan,
Sənə hər an can qurban!
Sənə min bir məhəbbət
Sinəmdə tutmuş məkan!

Namusunu hifz etməyə,
Bayrağını yüksəltməyə
Cümlə gənclər müştəqdir!
Şanlı Vətən! Şanlı Vətən!
Azərbaycan! Azərbaycan!



ГЕЙДАР АЛИЕВ
ОБЩЕНАЦИОНАЛЬНЫЙ ЛИДЕР
АЗЕРБАЙДЖАНСКОГО НАРОДА

РАМИН МАХМУДЗАДЕ, ИСМАИЛ САДЫГОВ, НАИДА ИСАЕВА

ИНФОРМАТИКА

9

УЧЕБНИК

по предмету информатика для 9-х классов общеобразовательных заведений

©Azərbaycan Respublikası Elm və Təhsil Nazirliyi



**Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International (CC BY-NC-SA 4.0)**

Bu nəşr Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International lisenziyası (CC BY-NC-SA 4.0) ilə www.trims.edu.az saytında əlçatandır. Bu nəşrin məzmunundan istifadə edərkən sözügedən lisenziyanın şərtlərini qəbul etmiş olursunuz:

İstinad zamanı nəşrin müəllif(lər)inin adı göstərilməlidir.



Nəşrdən kommersiya məqsədilə istifadə qadağandır.



Törəmə nəşrlər orijinal nəşrin lisenziya şərtlərilə yayılmalıdır.



Замечания и предложения, связанные с этим изданием, просим отправлять на электронные адреса: bn@bakineshr.az и derslik@edu.gov.az
Заранее благодарим за сотрудничество!

Информатика

Содержание

1. КОДИРОВАНИЕ

1. Растровая графика	8
2. Векторная графика	12
3. Работа в векторном редакторе.....	16
4. Кодирование графической информации	19
5. Кодирование звуковой информации	22
Обобщающие вопросы и задания	26

2. КОМПЬЮТЕР

6. Основные характеристики компьютера	28
7. Панель управления	32
8. Служебные программы. Дефрагментация	35
9. Служебные программы. Очистка диска	38
Обобщающие вопросы и задания	42

3. ЭЛЕКТРОННЫЕ ТАБЛИЦЫ

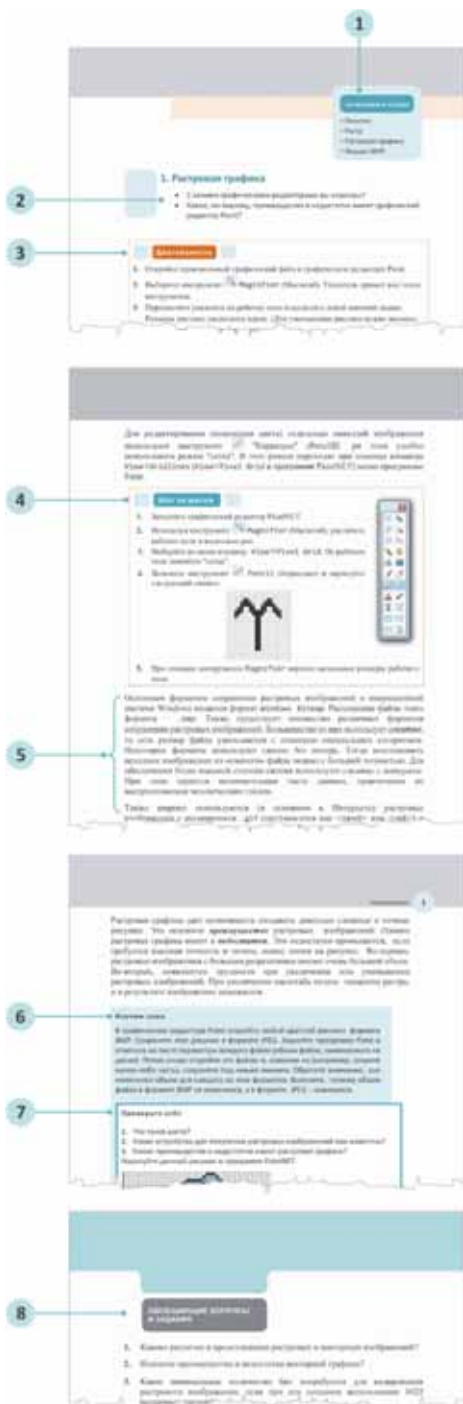
10. Абсолютные и относительные ссылки	44
11. Функции в электронной таблице	49
12. Визуализация данных. Диаграммы.....	54
13. Решение задач	60
Обобщающие вопросы и задания.....	66

4. ПРОГРАММИРОВАНИЕ

14. Работа с числами	70
15. Строки.....	75
16. Списки.....	80
17. Функция	86
18. Решение задач.....	91
Обобщающие вопросы и задания	98

5. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

19. Информационная модель на графах	102
20. Задачи, связанные с графами	106
21. Топологии сети	111
22. Веб-сайт	115
23. Шаблоны веб-сайтов	119
24. Языки веб-программирования	124
25. Адресация в Интернете	130
Обобщающие вопросы и задания	134
Литература.....	135



● **ЗНАКОМСТВО С УЧЕБНИКОМ**

1. **Ключевые слова.** Основные понятия, изучаемые по каждой теме.
2. **Мотивация.** Описаны различные ситуации и явления, которые завершаются вопросами.
3. **Деятельность.** Задания исследовательского характера, направленные на изучение интересных событий, процессов, выявление их причинно-следственных связей. Для обсуждения выполненной работы и выявления возможных ошибок предлагаются вопросы.
4. **Шаг за шагом.** Закрепление практических навыков.
5. **Разъяснения.** Основная часть урока: вводятся новые понятия, формулируются правила.
6. **Изучим сами.** Задания для самостоятельного изучения и применения своих знаний.
7. **Проверьте себя.** Предназначено для закрепления материалов по каждой теме, определения слабых сторон в обучении.
8. **Обобщающие вопросы и задания.** Даны вопросы и задания обобщающего характера по всему разделу.



КОДИРОВАНИЕ

1. Растровая графика
2. Векторная графика
3. Работа в векторном редакторе
4. Кодирование графической информации
5. Кодирование звуковой информации




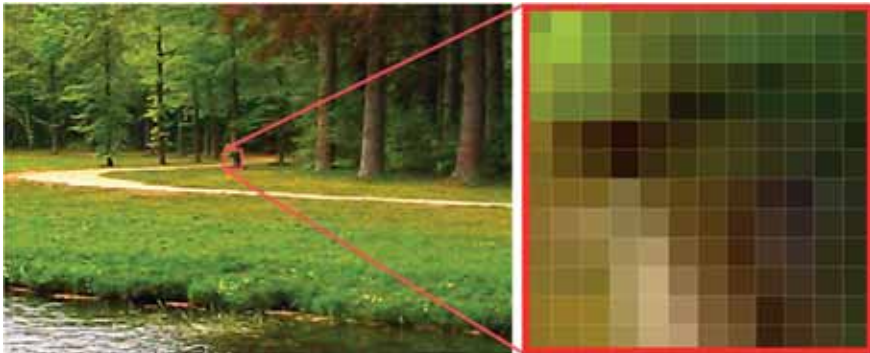
- Пиксель
- Растр
- Растровая графика
- Формат BMP

1. Растровая графика

- С какими графическими редакторами вы знакомы?
- Какие, по-вашему, преимущества и недостатки имеет графический редактор Paint?

Деятельность

1. Откройте произвольный графический файл в графическом редакторе Paint.
2. Выберите инструмент  Magnifier (Масштаб). Указатель примет вид этого инструмента.
3. Переместите указатель на рабочее поле и щелкните левой кнопкой мыши. Размеры рисунка увеличатся вдвое. (Для уменьшения рисунка нужно воспользоваться правой кнопкой мыши).
4. Повторите это действие несколько раз.

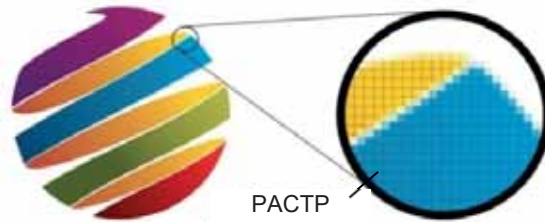


Обсудим:

- Что можно сказать о формировании изображения, посмотрев на его увеличение?

Вы уже знаете о том, что изображение на экране формируется из одинаковых маленьких точек – **пикселей**. Прямоугольная сетка точек, формирующая изображение на экране компьютера, называется **растром** (термин "растр" происходит от латинского слова "rastrum" и означает "грабли"). Изображение,

созданное при помощи вычислительной техники и сохраненное в памяти компьютера в виде набора битов, соответствующего пикселям экрана, называют *растровой графикой*.



Точки на экране создают целый рисунок, но если растровое изображение сильно увеличить, оно будет похоже на мозаику (сетку), состоящую из маленьких клеточек. Каждая точка растра характеризуется двумя параметрами: положением на экране и, если монитор цветной, – цветом, а если черно-белый, – степенью яркости.

Для создания растровых изображений и работы с ними существует много разных программ. С программой **Microsoft Paint** операционной системы Windows вы знакомы с младших классов. Для создания двухмерных рисунков можно использовать программы с более широкими возможностями, такие как **PaintNET**, **Corel Painter**, **Adobe PhotoShop** или **Corel PHOTO-PAINT**. Для работы с растровыми изображениями имеются и свободно распространяемые (бесплатные) программы. К таким программам можно отнести **PaintNET**, **GIMP**, **TuxPaint**, **StarOffice Image**.



PaintNET



Adobe PhotoShop




GIMP





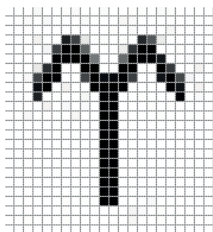
TuxPaint

Несмотря на то, что растровые изображения состоят из набора пикселей, обычно приходится работать не с отдельными пикселями, а целым фрагментом. Для этого в графическом редакторе предусмотрены разные инструменты: прямая линия, кривая, прямоугольник, овал (эллипс), многоугольник, скругленный прямоугольник, кисть, ластик, заливка цветом и др. Но иногда возникает необходимость работать с отдельными пикселями изображения. Например, предположим, на какой-то части созданного изображения необходимо разместить символ (Ÿ), состоящий всего из нескольких точек. Если попробовать выполнить это при помощи инструментов, с которыми до этого работали в редакторе Paint, это окажется не так просто. Методом, который вы сейчас изучите, это можно выполнить очень легко.

Для редактирования (изменения цвета) отдельных пикселей изображения используют инструмент  "Карандаш" (Pencil). При этом удобно использовать режим "сетка". В этот режим переходят при помощи команды View⇒Grid (View⇒Pixel Grid в программе PaintNET) меню программы Paint.

Шаг за шагом

1. Запустите графический редактор PaintNET.
2. Используя инструмент  Magnifier (Масштаб), увеличьте рабочее поле в несколько раз.
3. Выберите из меню команду View⇒Pixel Grid. На рабочем поле появится "сетка".
4. Возьмите инструмент  Pencil (Карандаш) и нарисуйте следующий символ.



5. При помощи инструмента Magnifier верните начальные размеры рабочего поля.



Основным форматом сохранения растровых изображений в операционной системе Windows является формат Windows Bitmap. Расширение файла этого формата – .bmp. Также существует множество различных форматов сохранения растровых изображений. Большинство из них использует *сжатие*, то есть размер файла уменьшается с помощью специальных алгоритмов. Некоторые форматы используют сжатие без потерь. Тогда восстановить исходное изображение из «сжатого» файла можно с большей точностью. Для обеспечения более высокой степени сжатия используют *сжатие с потерями*. При этом теряется незначительная часть данных, практически не воспринимаемая человеческим глазом.

Также широко используются (в основном в Интернете) растровые изображения с расширением .gif (произносится как <джиф> или <гиф>) и .jpeg (произносится как <джей-пэг>). В последнее время широкое распространение получил формат .png (произносится как <пинг>). Несмотря на то, что размер файла, сохраненного в этих форматах, заметно меньше, качество изображения в них почти не ухудшено. Формат .tiff используется для сохранения растровых изображений в высоком качестве.

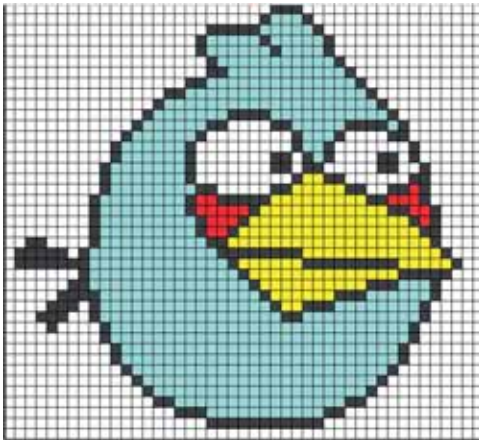
Растровая графика дает возможность создавать довольно сложные и точные рисунки. Это основное **преимущество** растровых изображений. Однако растровая графика имеет и **недостатки**. Эти недостатки проявляются, если требуется высокая точность и четкие, ясные линии на рисунке. Во-первых, растровые изображения с большим разрешением имеют очень большой объем. Во-вторых, появляются трудности при увеличении или уменьшении растровых изображений. При увеличении масштаба видны элементы растра, и в результате изображение искажается.

Изучим сами

В графическом редакторе Paint откройте любой цветной рисунок формата BMP. Сохраните этот рисунок в формате JPEG. Закройте программу Paint и отметьте на листе параметры каждого файла (объем файла, занимаемого на диске). Потом снова откройте эти файлы и, изменив их (например, сотрите какую-либо часть), сохраните под новым именем. Обратите внимание, как изменился объем для каждого из этих форматов. Выясните, почему объем файла в формате BMP не изменился, а в формате JPEG – изменился.

Проверьте себя

1. Что такое растр?
2. Какие устройства для получения растровых изображений вам известны?
3. Какие преимущества и недостатки имеет растровая графика?
4. Нарисуйте данный рисунок в программе PaintNET.



ключевые слова

- Векторная графика
- Формула
- Инструмент выделения
- Формат ODG

2. Векторная графика

- Какие формы представления функций в математике вам известны?

Деятельность

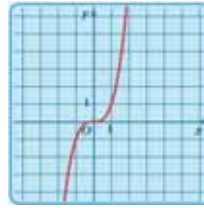
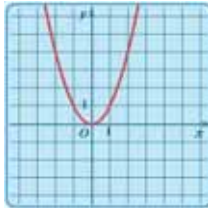
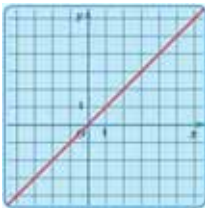
Определите соответствие формул представленным графикам.

$$y = \sin x$$

$$y = x^2$$

$$y = kx$$

$$y = x^3$$



Обсудим:

- Для чего требуется больше места для хранения на компьютере – для формул или соответствующих им графиков?

Существует два подхода в представлении графической информации на компьютере. С особенностями первого из них – *растровой графикой*, ее преимуществами и недостатками вы познакомились на прошлом уроке. Для решения проблем, возникающих из-за недостатков растровой графики, используют второй подход – *векторную графику*. Преимущества векторной графики проявляются при работе с рисунками, чертежами, схемами, для которых имеет значение сохранение четких и ясных контуров.

В векторной графике объекты формируются не с помощью отдельных точек (пикселей), а с помощью набора линий. Форма линий задается при помощи математических закономерностей. То есть для определения места, длины и направления линии на экране или при распечатке в векторной графике используют математические формулы. Например, как вы знаете, для того, чтобы нарисовать прямую линию, достаточно двух параметров. Уравнение

прямой линии задается формулой $y = kx + b$. Зная параметры k и b , можно нарисовать прямую на координатной плоскости. То есть при этом не нужно указывать координаты и цвет каждой точки прямой. В итоге размер графического изображения уменьшается. К тому же, изображение может быть увеличено или уменьшено без потери качества.

Сравнение растрового и векторного изображения



Вместе с тем в векторных изображениях нелегко показать плавный переход от одной цветовой гаммы к другой, рисовать произвольные границы и контуры. Этими качествами обладают фотографии и рисунки. Поэтому векторная графика используется, как правило, для создания схем, чертежей.

Для работы с векторными изображениями разработано много программ. Среди них наиболее распространены **CorelDRAW**, **Adobe Illustrator**, **Micromedia Freehand**. Предусмотренная для создания в Интернете анимаций программа **Micromedia Flash** тоже базируется на принципах векторной графики. Все указанные программы являются коммерческими продуктами, но есть и свободно распространяемые (бесплатные) программы. К ним относятся **Inkscape**, **OpenOffice.org Draw**, **Skencil**.



Adobe Freehand



Adobe Flash



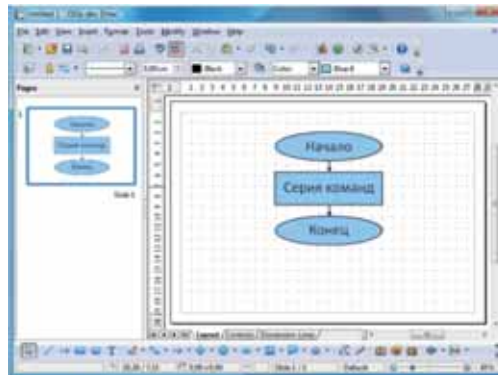
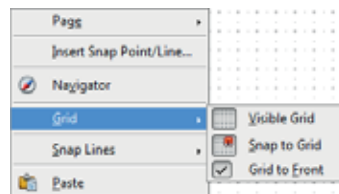
Inkscape

Во всех векторных редакторах набор основных инструментов для рисования и редактирования почти одинаковый. Среди них очень важен **инструмент выделения**. Программа OpenOffice.org Draw позволяет рисовать прямые линии, разные кривые, прямоугольники, окружности и другие фигуры. Инструменты трехмерной графики предусмотрены для рисования параллелепипеда, конуса, цилиндра и других трехмерных геометрических объектов. Кроме этого при помощи этой программы можно создавать разные элементы управления веб-страницами (кнопки, текстовые поля, радиокнопки и т.д.).

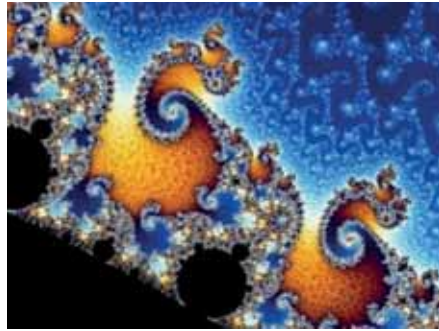
Созданные изображения можно сохранять в формате самой программы – **ODG** или экспортировать во все известные форматы – **BMP**, **GIF**, **JPEG**, **PNG**, **TIFF** и **WMF**.

Шаг за шагом

1. Запустите программу OpenOffice.org Draw. Откроется основное окно программы.
2. Для определения размеров области рисования и ее ориентации переместите указатель на область рисования, а затем щелкните правой кнопкой мыши. В открывшемся контекстном меню выберите пункт Page⇒PageSetup. Откроется соответствующее диалоговое окно.
3. В разделе формата Paper установите формат листа, его ориентацию (Landscape), поля по краям.
4. Для получения более точного изображения фигур, которые будут нарисованы, создайте сетку и сделайте ее видимой. Для этого в области рисунка снова откройте меню и активизируйте пункты, показывающие параметры сетки в пункте Grid.
5. Щелкните по кнопке эллипса (E l l i p s e), размещенной в нижней части программного окна, и нарисуйте соответствующую фигуру. Щелкните по кнопке Text для набора текста и внутри эллипса запишите слово **Начало**.
6. В той же последовательности, нарисовав фигуры прямоугольник (R e c t a n g l e) и снова эллипс (E l l i p s e), впишите в них, соответственно, слова **Серия команд** и **Конец**.
7. Щелкните по кнопке L i n e Ends w i t h Arrow и соедините стрелками нарисованные фигуры. В итоге должна получиться общая блок-схема линейного алгоритма.
8. Для перемещения и копирования блок-схемы, состоящей из отдельных объектов, ее необходимо сгруппировать. Для этого, используя один из известных вам методов, выделите все объекты. Потом установите указатель мыши на выделенных объектах и щелкните ее правой кнопкой. Из открывшегося контекстного меню выберите команду Group. Все выделенные объекты будут объединены в одну группу.
9. Для сохранения созданного векторного изображения выберите команду меню F i l e⇒Save A s. В открывшемся соответствующем диалоговом окне укажите тип файла ODF Draw i n g и присвойте файлу имя **alqoritm.odg**.
10. Выберите команду меню F i l e⇒Export. Укажите в открывшемся соответствующем диалоговом окне формат JPEG. Блок-схема сохранится как растровое изображение. Закройте программу OpenOffice.org Draw.



Фрактальная графика. Понятия «фрактал», «фрактальная геометрия» и «фрактальная графика», появившиеся в 70-х годах прошлого века, сегодня очень тесно вошли в повседневную жизнь математиков и программистов. Слово «фрактал» происходит от латинского слова «fractus», что означает «разделенный», «сломанный». Этот термин был предложен французским математиком Бенуа Мандельбротом в 1975 году.



Фрактал – это структура, части которой в некотором смысле напоминают сами себя. Многие объекты в природе обладают фрактальными свойствами: снежинка при увеличении становится фрактальной; деревья и растения растут по фрактальным алгоритмам; если посмотреть на береговые линии моря в увеличенном масштабе, то обнаружатся новые впадины и выступы, аналогичные прежним.

Как и векторная графика, фрактальная графика является вычисляемой. Другими словами, фрактальное изображение целиком строится по уравнению или системе уравнений. Поэтому в памяти компьютера для выполнения всех вычислений, ничего кроме формулы хранить не требуется. Изменяя лишь коэффициенты уравнения можно получить совершенно отличные друг от друга изображения.

Изучим сами

Перейдите в папку, где вы сохранили файлы `alqori tm. odg` и `alqori tm. jpg`, созданные в блоке “Деятельность”. Сначала сравните, сколько места занимают эти файлы на диске. Потом откройте эти файлы соответственно в программах OpenOffice.org Draw и Paint. Увеличьте изображения в несколько раз и сравните их качество (точность контуров). Совпадают ли результаты, которые вы видите, с тем, что вы изучили?

Проверьте себя



1. Что такое векторная графика?
2. Из чего формируются векторные изображения?
3. Какие преимущества и недостатки имеет векторная графика?
4. Нарисуйте блок-схемы алгоритмических конструкций “Полное ветвление” и “Неполное ветвление”.
5. Нарисуйте структурную схему устройства компьютера.

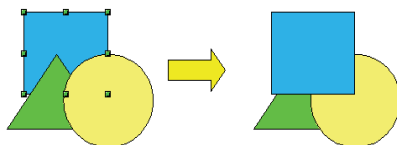
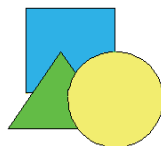
- Слой
- Группировать
- Разгруппировать

3. Работа в векторном редакторе

- Что такое графические примитивы?
- Какими графическими примитивами вы пользовались в программе Google SketchUp?

Деятельность

1. Нарисуйте следующий рисунок в графическом редакторе Paint.
2. Попробуйте изменить порядок расположения фигур друг на друге.
3. Запустите векторный редактор OpenOffice.org Draw.
4. На панели Drawing (Рисование) щелкните по кнопке  Basic Shapes. (панель Drawing обычно размещается в нижней части главного окна программы. Если она не отображена на экране, ее нужно активизировать при помощи команды меню View⇒Tool bars).
5. Выбрав из открывшегося списка сначала квадрат, затем треугольник и, наконец, окружность, получите заданный выше рисунок.
6. Выделите объект "квадрат", а затем выберите пункт  из меню Modify⇒Arrange⇒Bring to Front. Выделенный квадрат переместится на передний план.



Обсудим:

- Почему действие, легко выполнимое в векторном графическом редакторе, невозможно выполнить в растровом редакторе?


Особенность действий в векторной графике в том, что геометрические фигуры (объекты) остаются независимыми и после того, как нарисованы. То есть с объектами, которые составляют векторное изображение, можно и впоследствии выполнять действия. В растровых изображениях не так: геометрические фигуры (объекты) "свободны" только в момент их рисования, потом они превращаются в набор цветных точек.

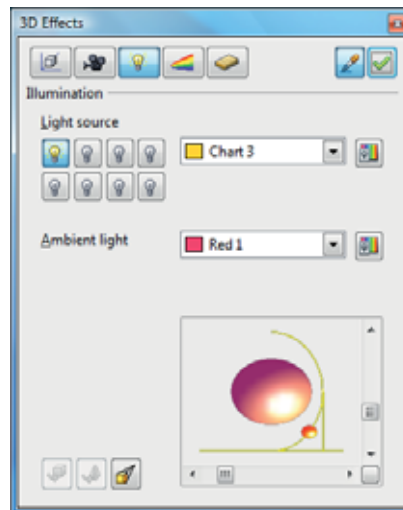
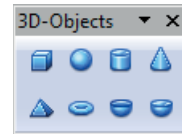
Каждый графический примитив, нарисованный в векторной графике, расположен на отдельном *слое*. То есть векторное изображение состоит из слоев, количество которых равно количеству составляющих его объектов. В векторном изображении объекты, хотя визуально и перекрывают друг друга, абсолютно независимы.

В векторном редакторе, сгруппировав (объединив) нарисованные примитивы, можно создать цельный объект. С действием **Группировать** (команда Group) вы познакомились на прошлом уроке. Действия, которые проводят над группой, относятся ко всем составляющим его объектам. Но иногда возникает необходимость производить определенные изменения с каким-то одним объектом в группе. Для этого используют противоположную группировке команду **Разгруппировать** (команда Ungroup).

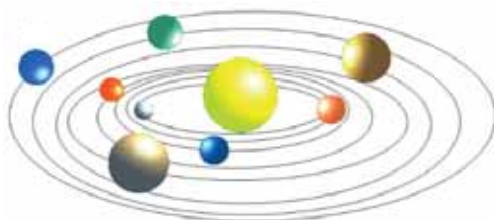
Вы уже создавали простые трехмерные фигуры в программе Google SketchUp. Программа OpenOffice.org Draw тоже имеет широкие возможности для работы с трехмерной графикой.

Шаг за шагом

1. Запустите программу OpenOffice.org Draw.
2. Выберите команду меню **File**⇒**New**⇒**Drawing**.
Откроется чистое поле для рисования.
3. Активизируйте панель инструментов 3D-Objects. Для этого выберите команду меню **View**⇒**Tool bars**⇒**3D-Objects**.
4. Последовательно постройте куб (Cube), шар (Sphere), цилиндр (Cylinder), конус (Cone), пирамиду (Pyramid), тор (Torus).
5. В поле рисунка выделите шар и щелкните на нем правой кнопкой мыши. Из открывшегося контекстного меню выберите пункт **3D Effects**.
Откроется соответствующее диалоговое окно.
6. Щелкните по кнопке **Illumination** (освещение), находящейся в верхней части окна. Выберите параметры **Light source** (Источник света) и **Ambient light** (Обтекающий свет).
При помощи мыши или полос прокрутки измените расположение источника света и проследите, как меняется освещенность трехмерной фигуры. Чтобы присвоить объекту выбранные свойства, щелкните по кнопке  **Assign** (Присвоение).



- Используя эти возможности редактора, создайте трехмерную модель Солнечной системы.



- Задайте файлу соответствующее имя и сохраните его в своей папке.

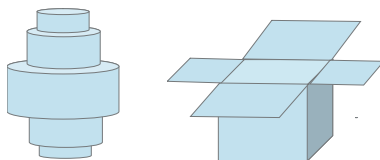
Изучим сами

- Выясните назначение кнопок на панели Drawing в программе OpenOffice.org Draw.
- Применяя другие пункты меню Arrange, обратите внимание на то, как будет меняться положение выделенного объекта по отношению к другим объектам.

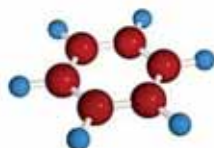


Проверьте себя

- С какими отличительными свойствами растровых и векторных изображений вы познакомились?
- Какие преимущества имеет работа со слоями в графическом редакторе?
- Почему в графическом редакторе Paint нет команды группировки?
- Создайте следующие фигуры в векторном редакторе.



- Нарисуйте в векторном редакторе модель молекулы химического вещества. Попробуйте определить название этого вещества.



ключевые слова

- Глубина цвета
- Дополняющие цвета
- Кодирование RGB
- Кодирование CMYK

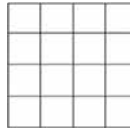
4. Кодирование графической информации

- Какие виды информации вы знаете?
- Как кодируется текстовая информация на компьютере?

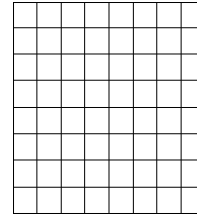
Деятельность

1. На листе в клетку начертите прямоугольники с соответствующим числом клеток по горизонтали и вертикали.
2. Найдите клетки, соответствующие "1", и закрасьте их.
3. Сравните полученные изображения.

0110
1001
1001
0110



00111100
01111110
11100111
11000011
11000011
11100111
01111110
00111100



Обсудим:

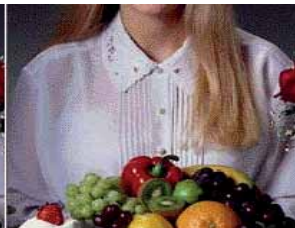
- Изображение какой таблицы получилось точнее? Почему? Что надо сделать, чтобы второе изображение получилось более качественным?
- Можно ли эти таблицы с 0 и 1 считать кодами полученных изображений?

Вне зависимости от формы представления графическая информация на экране компьютера кодируется в формате *растра*. При этом каждое изображение состоит из прямоугольной таблицы, сформированной цветными точками – пикселями. Цвет и яркость каждой точки выражается числом, что дает возможность использовать двоичный код для представления графической информации.

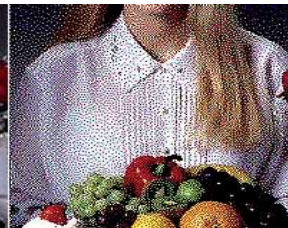
Количество битов, используемых для кодирования цвета 1 пикселя изображения, называют *глубиной цвета*. В цифровом видео для глубины цвета достаточно 15 бит. Для того чтобы на фотографии цвет был естественным, необходимо как минимум 24 бита.



24 бита (16 777 216 цветов)



8 бита (256 цветов)



4 бита (16 цветов)

Для того чтобы найти объем графического изображения, необходимо знать количество пикселей в ширину и в высоту (соответственно a и b), а также глубину цвета (n). Тогда объем графического изображения (V) можно рассчитать по следующей формуле:

$$V = a \cdot b \cdot n$$

В растровом изображении, имеющем размер 256×256 пикселей, использовано 512 цветов. Сколько места займет изображение в памяти компьютера?

Решение. Сначала найдем общее количество пикселей, формирующих изображение. Используя степени числа 2, можно упростить вычисления.

$$a \cdot b = 256 \cdot 256 = 2^8 \cdot 2^8 = 2^{16}.$$

1

Поскольку палитра содержит 512 цветов, то представим это число в виде степени 2 и найдем глубину цвета:

$$512 = 2^n \Leftrightarrow 2^9 = 2^n \Leftrightarrow n = 9.$$

Следовательно, для кодирования цвета было использовано 9 бит. Таким образом, объем памяти, занимаемый растровым изображением, будет

$$V = 2^{16} \cdot 9 \text{ бит} = 2^{13} \cdot 9 \text{ байт} = 2^3 \cdot 9 \text{ Кбайт} = 72 \text{ Кбайт}$$

На экране монитора любой цвет получается смешением **трех основных цветов**: *красного* (Red), *зеленого* (Green) и *синего* (Blue). Такая система кодирования называется **RGB** (читается как "ер-джи-би"). Смешивая эти цвета в определенном соотношении, можно получить любой другой цвет, воспринимаемый глазом человека; отсутствие всех трех цветов дает черный цвет, белый цвет получается 100-процентным присутствием всех трех цветов.

Если для кодирования яркости каждого из этих основных цветов используют 8 бит (из 256), то для кодирования каждой точки изображения необходимо 24 бита. Такая система кодирования позволяет однозначно определять 16,8 миллиона цветов. Режим применения 24 двоичных разрядов для кодирования цветной графики носит название **полноцветного** (True Color).



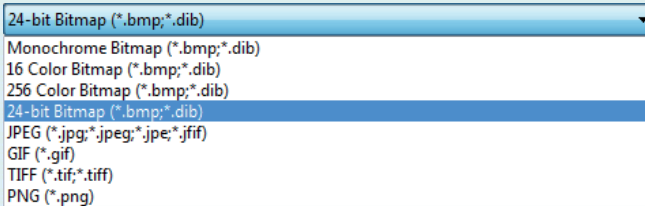
В полиграфии приемлемо использовать не три основных цвета, а **дополняющие их цвета**. Дополняющие цвета следующие: *голубой* (Cyan), *пурпурный* (Magenta) и *желтый* (Yellow). Так как вещества краски несо-

вершенны, при смешивании этих цветов не получается идеальный черный цвет. Поэтому при печати черный цвет (Black) используется отдельно. Такая система кодирования называется **СМУК** (читается как "цмик"); черный цвет, чтобы отличить его от синего, показан буквой K.

Если не требуется большой точности при отображении цветовой палитры, используют меньше битов. Например, кодирование изображений при помощи 16-битовых двоичных чисел называют **High Color**. Объем таких изображений (место, занимаемое на диске) в сравнении с объемом полноцветных (True Color) изображений намного меньше (естественно, и качество ниже).

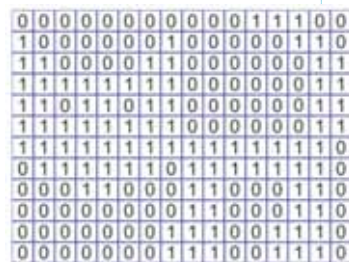
Изучим сами

Откройте любое цветное изображение формата BMP в графическом редакторе Paint. Сохраните этот рисунок в разных схемах кодирования (Monochrome Bitmap, 16 Color Bitmap, 256 Bitmap, 24-bit Bitmap). Закройте программу и перейдите в папку, в которой сохранили файлы. Сравните размеры сохраненных файлов и постарайтесь объяснить причину разницы между ними.



Проверьте себя

1. Какие цвета называют основными?
2. Определите, какое изображение получится, если закрасить клетки, соответствующие цифре 1.
3. Для сохранения растрового изображения размером 32×32 пикселей выделено 512 байт места. Какое наибольшее количество цветов может быть на палитре изображения?
4. Размер 256-цветного изображения 1024×640 пикселей. Определите объем графического файла.
5. Каким будет объем изображения размером 640×480 в 16-битовом кодировании?



- Аналоговый сигнал
- Цифровой сигнал
- Аналого-цифровой преобразователь
- Цифро-аналоговый преобразователь
- Звуковая карта
- Фонограф

5. Кодирование звуковой информации

- Что такое звук с точки зрения физики?
- Какие устройства компьютера работают со звуком?

Деятельность

Создайте в текстовом редакторе следующую таблицу. Запишите в соответствующие поля информацию о данных устройствах.

Устройство	Назначение, принцип работы
Микрофон	
Звуковые колонки	
Звуковая карта	

Обсудим:

– Как эти устройства связаны друг с другом в компьютере?

Компьютер, как и большинство современных электронных устройств (фото- и видеокамеры, мобильные телефоны и др.), является **цифровым устройством**, так как при хранении и обработке информации он работает с числами (цифрами).

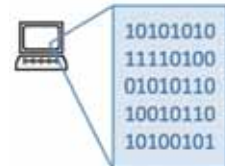
Часто для представления непрерывно изменяющихся физических величин используется термин "**аналоговый**". Например, звуковые волны, создаваемые голосовыми связками человека во время разговора, имеют аналоговую природу. Эти волны микрофон может преобразовать в переменные электрические сигналы, которые также являются **аналоговыми**.



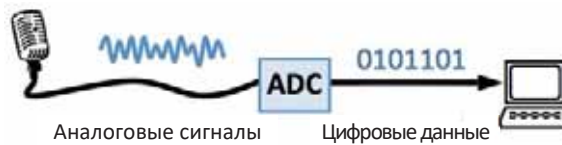
Аналоговые звуковые волны



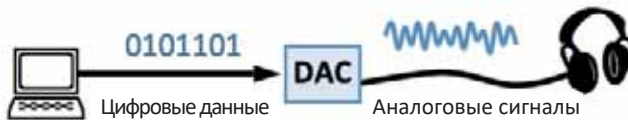
Аналоговый электрический сигнал



Так как компьютер является цифровым устройством, для работы его с аналоговыми устройствами необходим преобразователь. Такой преобразователь имеется на **звуковой карте** компьютера. Для преобразования электрических сигналов, поступающих из микрофона (аналоговых сигналов), в цифровые сигналы служит **аналого-цифровой преобразователь** (analog-to-digital converter, **ADC**), размещенный на звуковой карте.



Помимо этого, на звуковой карте компьютера размещено устройство, которое преобразовывает цифровые данные в аналоговые сигналы (звук). **Цифро-аналоговый преобразователь** (digital-to-analog converter, **DAC**) поступающие цифровые сигналы преобразовывает в аналоговые сигналы и передает их в наушники или на звуковые колонки. В свою очередь эти устройства передают аналоговые сигналы в виде звука на выход.



Цифро-аналоговый преобразователь и аналого-цифровой преобразователь есть и на других устройствах. Например, на плеере компакт-дисков (MP3-плеере) имеется цифро-аналоговый преобразователь для преобразования цифровых данных, считанных с диска, в аналоговый сигнал в виде звука.



При таком кодировании звука случаются определенные искажения, так как сигнал, полученный от цифро-аналогового преобразователя, немного отличается от оригинала. Это отличие особенно заметно при записи и озвучивании высококачественной оркестровой музыки.

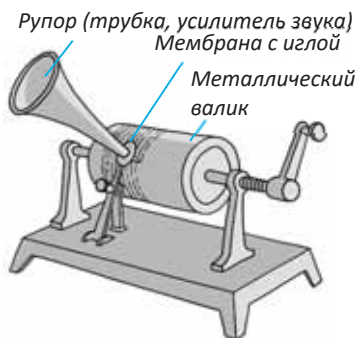
Как и остальные данные, звук на компьютере также хранится в файлах. В основном используют три типа звуковых файлов (аудиофайлов): **wav**, **mp3** и **midi**.

WAV. Файл этого формата – звуковой файл, который содержит фрагменты музыки в цифровой форме. В файлах **Wav** (с англ. *wave* – волна) хранятся звуки Windows и созданные вами аудиофайлы.

MP3. Это формат со сжатием звуковых файлов для уменьшения занимаемого ими объема на диске. Каждая минута MP3-файлов занимает приблизительно 1 Мбайт памяти. Произвольный звуковой файл, как правило, удается преобразовать в формат **mp3**.

MIDI. В файлах MIDI есть информация о нотах и музыкальных инструментах, которые их озвучивают. Синтезатор воспроизводит мелодию, как музыкант, работающий с нотами.

Схема первого фонографа



История звукозаписи. Вы узнали, как хранится (кодируется) звуковая информация на компьютере. О том, как записать звук и передать его последующим поколениям, люди задумывались очень давно.

Известно, что вибрирующий предмет издает звук. Например, если бить по барабану, он начинает дрожать (вибрировать) и заставляет двигаться (колебаться) частички окружающего воздуха. Возникающие при этом звуковые волны распространяются в воздухе и воспринимаются ухом.

В 1876 году был создан **телефон**. В нем звуковые вибрации воспринимала мембрана – тоненькая эластичная пластинка. Великого изобретателя Томаса Эдисона волновал вопрос, можно ли записать вибрацию пластинки. Эдисон взял звуковую трубку и присоединил к мембране иглу. Когда что-то касалось рупора изнутри, вместе с мембраной вибрировала и игла, которая оставляла след на бумаге под ней.

Таким образом, удалось записать звук, но осталась задача – озвучить его. У Эдисона появилась еще одна великая идея: вместо обычной бумаги он использовал оловянную фольгу. Когда иглу подносили к цилиндру, обернутому фольгой, она оставляла на фольге борозды. В зависимости от вибрации иглы менялась глубина этих борозд.

У Эдисона возникла новая мысль: ведь этот процесс можно повернуть вспять!

Если конец иглы движется при трении о бороздку, он, как в случае записи звука, заставит мембрану вибрировать! Эдисон нарисовал эскиз аппарата и передал работникам лаборатории. После того, как аппарат был готов, Эдисон испытал его. Он, вращая цилиндр, прочитал в звуковую трубку (рупор) известное детское стихотворение "У Мери был барашек". Игла

прочертила борозды на цилиндре. Эдисон поместил конец иглы в начало борозды и, прокрутив барабан, с радостью услышал в звуковой трубке свой голос.

Таким образом в мире появилось первое устройство для записи звука – **фонограф**. Дальнейшее его развитие стало причиной появления **граммофона** и **патефона**. В середине 20 века появился **электрофон** – электрический аналог фонографа.

Запись звука и преобразование файлов из одного формата в другой осуществляются с помощью специального программного обеспечения. Чтобы найти эти программы в Интернете, осуществите поиск со словами "**sound conversion software**" на сайте www.google.com.

При **кодировании видеoinформации** приходится кодировать как последовательность кадров (изображений), так и звук. При этом для сокращения общего объема файла пользуются различными методами. Например, между двумя соседними кадрами записываются только различия.

Изучим сами

Найдите в Интернете и скачайте на компьютер одну и ту же информацию в нескольких форматах (текст, звук, видео), например, файл поэмы Магомедгусейна Шахрияра "Неудərbabaya salam". Сравните размеры этих файлов. Файл какого формата имеет наименьший, а какого – наибольший объем? Постарайтесь объяснить причину этого.

Проверьте себя

1. Какую функцию выполняют устройства ADC и DAC?
2. Какие еще устройства, в названии которых имеется слово "фон", вам известны?
3. Как кодируется звуковая информация на компьютере?
4. Какие типы звуковых файлов существуют?
5. Что такое фонограф и как он работает?

ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

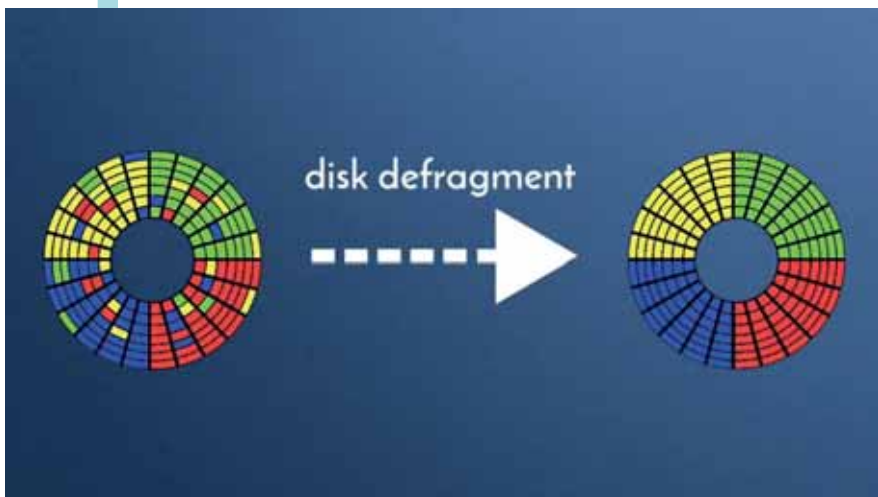
1. Каково различие в представлении растровых и векторных изображений?
2. Назовите преимущества и недостатки векторной графики?
3. Какое минимальное количество бит потребуется для кодирования растрового изображения, если при его создании использовано 1025 различных цветов?
4. Размер черно-белого изображения составляет 640×320 пикселей. Каков информационный объем данного изображения?
5. Какой файл займет в памяти компьютера больше места: текст из 2056 символов в кодировке Unicode или 4-цветное растровое изображение размером 10×10 пикселей?
6. В процессе преобразования растрового графического изображения количество цветов уменьшилось с 65536 до 16. Во сколько раз уменьшится размер памяти, занимаемый изображением?
7. Каково назначение звуковой карты компьютера?
8. Какие из причин кодирования информации верны?
 - 1) скрыть информацию от других;
 - 2) сократить объем информации;
 - 3) демонстрация превосходства одной формы представления над другой;
 - 4) информацию легче обрабатывать и передавать;
 - 5) переход из одной формы представления информации в другую.

A) 1, 3, 4 B) 1, 2, 4 C) 1, 2, 4, 5
D) 1, 3, 5 E) 1, 2, 3, 5
9. Для хранения растрового изображения размером 32×32 пикселя отведено 512 байтов памяти. Каково максимально возможное число цветов в палитре изображения?
10. Каковы основные цвета палитры CMYK?
 - A) красный, синий, зеленый
 - B) красный, желтый, зеленый
 - C) синий, желтый, зеленый
 - D) голубой, пурпурный, желтый
 - E) синий, желтый, пурпурный



КОМПЬЮТЕР

- 6. Основные характеристики компьютера
- 7. Панель управления
- 8. Служебные программы. Дефрагментация
- 9. Служебные программы. Очистка диска



ключевые слова

- Центральный процессор
- Тактовая частота
- Ядро
- Оперативная память
- Кэш-память
- Жесткий диск
- Видеоадаптер

6. Основные характеристики компьютера

- Какие виды персональных компьютеров вы знаете?
- Какую часть компьютера называют его "мозгом"?

Деятельность

Заполните таблицу на основе данных компьютера, за которым вы работаете. Для просмотра параметров компьютера следует на рабочем столе щелкнуть правой кнопкой мыши на значке Computer и из открывшегося контекстного меню выбрать команду Properties.

Параметр	Значение параметра
Операционная система	
Название процессора	
Скорость процессора	
Объем оперативной памяти	
Разрешение экрана монитора	

Обсудим:

- Как вы считаете, какой из этих показателей является наиболее важным для преимущества одного компьютера перед другим?

Вероятно, вам известно, что из года в год растет мощность и производительность (скорость вычислений) компьютеров. Этот рост связан в первую очередь с устройствами, из которых состоит компьютерная система. Чем выше будет производительность этих устройств, тем выше будет и общая производительность компьютера.

Основным устройством компьютера является **центральный процессор**, или просто **процессор**. Быстродействие процессора является основным из факторов, влияющих на производительность компьютера. Так уж сложилось, что за скорость процессора принимают *количество элементарных операций, выполняемых за одну секунду*. Это было основано на том, что компьютеры решали однотипные задачи, и для выполнения различных операций требовалось приблизительно одно и то же время. В современных компьютерах это не так. Команды, входящие в набор команд процессора, разные по сложности выполняемых действий, поэтому и время выполнения процессором различных

команд может различаться в несколько раз. К тому же скорость работы программ определяется не только процессором, но зависит и от других компонентов компьютерной системы (например, от производительности видеосистемы). Помимо этого, современные компьютеры выполняют некоторые операции (в том числе отдельные команды) *параллельно*, то есть одновременно.

Выполнение процессором любой команды можно представить как последовательность исполнения элементарных операций. Для любой элементарной операции, изменяющей состояние процессора, необходимо определенное время – *один такт*. Большинство команд выполняется за несколько тактов. Частота тактов процессора (*тактовая частота*) задается специальным генератором.

Если два процессора отличаются друг от друга только тактовой частотой, то их скорость обработки данных прямо пропорциональна частоте тактов. Поэтому тактовую частоту процессора можно рассматривать как основной показатель его производительности.

Для измерения частоты принята физическая величина 1 *герц* (Гц). Так как количество тактов в секунду в первых персональных компьютерах составляло несколько миллионов, для частоты тактов была принята стандартная величина, равная 1 *мегагерц* (1 МГц). Частота тактов современных персональных компьютеров измеряется *гигагерцами* (1 ГГц).

Тактовая частота процессора – не единственный показатель производительности компьютерной системы. Процессоры с одинаковой тактовой частотой могут иметь разную производительность. Кроме этого производительность компьютера зависит не только от процессора. Если другие компоненты компьютера не могут доставлять процессору данные со скоростью его обработки, процессор начинает простаивать, и, в результате общая производительность системы падает.

Часть центрального процессора, отвечающая за вычисления, называется *ядром*. Первые компьютеры имели одно ядро. В современных компьютерах их два и более. Наличие нескольких ядер позволяет параллельно выполнять несколько задач (программ), а это существенно повышает скорость компьютера. То есть чем больше ядер у центрального процессора, тем выше его производительность.



При работе компьютера между процессором и его оперативной памятью происходит непрерывный обмен данными. Скорость этого обмена является

основным показателем производительности компьютерной системы. Она зависит как от параметров микросхем памяти, так и от особенностей системы компьютера в целом. Обычно частота шины (электрического проводника, предусмотренного для передачи данных), соединяющей процессор и память, в несколько раз меньше рабочей частоты процессора, что является причиной задержек в его работе. Для сокращения этих задержек процессор оборудован специальной памятью – *кэш-памятью*. Размещенная на микросхеме процессора не очень большая по объему кэш-память работает с его частотой. Во время чтения-записи с оперативной памяти копия данных записывается в кэш-память. При повторном обращении процессора к этим данным они берутся сразу с кэш-памяти. Это устраняет простои.

Показатели скорости процессора и оперативной памяти компьютера оказывают существенное влияние на скорость выполнения вычислений. Но, как вы знаете, современные компьютеры используют не только для вычислений. Поэтому на производительность компьютера оказывают влияние и другие его характеристики:

- объем оперативной памяти;
- объем внешней памяти;
- производительность видеосистемы.

Один из параметров, существенно влияющих на быстродействие компьютера, – это *объем оперативной памяти* (RAM). Современные компьютеры работают в *многозадачном режиме*. В одно и то же время для каждой работающей задачи (программы) операционная система выделяет в оперативной памяти отдельную область. Если общий объем оперативной памяти, необходимой для всех задач, окажется больше объема имеющейся оперативной памяти, операционная система сформирует на жестком диске дополнительную память. Но, так как скорость обращения к жесткому диску намного ниже, чем к оперативной памяти, это снижает скорость работы компьютера. Установка на компьютере дополнительных модулей оперативной памяти снижает необходимость в таких действиях и повышает скорость работы системы в целом.

Одной из важных характеристик компьютера является объем его *жесткого диска*. Если объем диска, предусмотренного для длительного хранения программ и данных, недостаточен, возникает необходимость в дисках с большим объемом.

Технологии развиваются стремительно и уже жесткие диски стали заменять *SSD-накопители*. В отличие от жестких дисков, в устройствах SSD (Solid state drive) отсутствуют какие-либо движимые компоненты, и в сравнении с жесткими дисками их скорость намного больше. Для хранения данных в SSD используется флеш-память. Загрузка операционной системы, установленной на этих устройствах, занимает всего несколько секунд.

При работе на компьютере приходится постоянно смотреть на монитор, поэтому очень важно правильно выбрать сочетание *монитор-видеоадаптер*. Основной показатель при этом – *количество кадров в секунду*. Если эта вели-

чина меньше 70 кадров в секунду, глаз человека (и его мозг) ощущает мерцающие изображения на экране, что ведет к утомлению.

Для указания параметров компьютера используют ряд нестандартных сокращений и обозначений. Каждый покупатель компьютера когда-то сталкивался с такими записями из букв латинского алфавита и цифр, разделенных запятой или символом "|".

Intel® Core i5-9400F 2666 MHz | H310 Chipset | DDR4 16 GB | SSD 240 GB | NVIDIA® GeForce® GTX 1050Ti 4 GB

В данном примере надпись Intel® Core i5-9400F указывает на модель процессора и его тактовую частоту 2666 МГц. Надпись DDR4 16 GB указывает на модель и объем оперативной памяти, а SSD 240 GB указывает на то, что в качестве внутреннего накопителя используется SSD-носитель емкостью 240 ГБ.

Развитие аппаратного обеспечения компьютеров происходит высокими темпами, их средние характеристики ежегодно меняются. Тактовая частота процессоров Intel Core i9 составляет 5 ГГц. Объем оперативной памяти компьютеров общего назначения уже составляет несколько гигабайт. А в персональных компьютерах, предусмотренных для решения сложных задач, этот показатель достигает 16 гигабайт. Объем жесткого диска составляет 500 гигабайт и выше.

Изучим сами

У первого IBM PC компьютера 8088/86, выпущенного в 1981 году, тактовая частота процессора составляла 4,77 МГц. В 1993 году этот показатель у процессора Pentium составлял 66 МГц. В последнем выпуске 1997 года у процессора Pentium частота достигала 233 МГц.

Сравните тактовую частоту процессора вашего компьютера с аналогичным параметром самого мощного процессора Pentium. Во сколько раз ваш процессор сильнее него?

Проверьте себя

1. Какой основной параметр компьютера определяет его производительность?
2. Что такое тактовая частота и в каких единицах она измеряется?
3. Где расположена кэш-память и в чем заключается ее назначение?
4. Каковы объемы оперативной памяти и жесткого диска компьютера с приведенными ниже характеристиками?

Intel® Core™ i9-9900K 2400 MHz | Z390M Chipset | DDR4 32 GB | M2 SSD 512 GB
| HDD 2 TB | NVIDIA® GeForce® GTX1660Ti 6 GB

5. Какова скорость процессора компьютера, характеристики которого приведены ниже?

Intel® Core™ i9-11900K 3200 MHz | Z590 Chipset | DDR4 32 GB | SSD 1 TB NVMe |
HDD 1 TB | NVIDIA® GeForce® RTX 3090 24 GB

- Панель управления
- Раскладка клавиатуры

7. Панель управления

- В каких случаях возникает необходимость в упорядочении значков рабочего стола?
- Как можно отсортировать значки на рабочем столе в соответствии с типом программ?

Как вы знаете, операционная система компьютера управляет его аппаратным и программным обеспечением и играет роль посредника между пользователем и компьютером. Из-за разнообразия потребностей пользователей часто возникает необходимость изменения параметров в настройках системы.

В прошлом году вы узнали, как упорядочить значки на рабочем столе компьютера. Для этого используется контекстное меню, раскрывающееся при нажатии правой кнопки мыши на свободном месте рабочего стола. Для удобства пользователя все инструменты для настройки операционной системы Windows и ее элементов собраны в Control Panel (*Панель управления*).

Деятельность

1. Откройте меню Start.
2. Выбрав пункт Control Panel, откройте соответствующее окно.
3. Инструменты Панели управления сгруппированы по категориям (Category). Ознакомьтесь с инструментами в каждой категории.
4. Для отображения всех инструментов в одном списке войдите в раздел View by (Просмотр) и выберите из списка пункт Large icons (Крупные значки) или Small icons (Мелкие значки).
5. Закройте окно Control Panel.



Обсудим:

- На сколько категорий разделены инструменты Панели управления?
- В какой из категорий могут находиться инструменты для упорядочения значков рабочего стола?

Выбрав соответствующие инструменты, можно, к примеру, настроить языки операционной системы, внешний вид окон, отображение меню Start, создать нового пользователя на компьютере. Некоторые программы добавляют свои настройки в Панель управления, поэтому на разных компьютерах ее состав может различаться.

Многие пункты Панели управления практически не используются, так как поставленные по умолчанию настройки удовлетворяют большинство пользователей. К таким пунктам относятся Mouse (Мышь), Sound (Звук), Ease of Access Center (Центр Специальных возможностей). Но имеются пункты, обращение к которым при настройке операционной системы происходит часто. Это, например, Personalization (Персонализация). Многие инструменты Панели управления доступны не только из самой панели, но и из других мест. Например, чтобы перейти к пункту Personalization, достаточно щелкнуть правой кнопкой мыши на свободном месте рабочего стола и из контекстного меню выбрать одноименный пункт.

Раскладка клавиатуры определяет символы, которые будут выведены на экран при нажатии клавиш, и позволяет вводить все символы языка. После установки необходимой раскладки клавиатуры можно с легкостью переходить при наборе текста с одного языка (например, с английского) на другой (например, русский) с помощью языковой панели.

Шаг за шагом

1. Откройте Панель управления.



2. В категории (Часы, язык и регион) щелкните по ссылке **Change keyboards or other input methods** (Смена раскладки клавиатуры или других способов ввода).

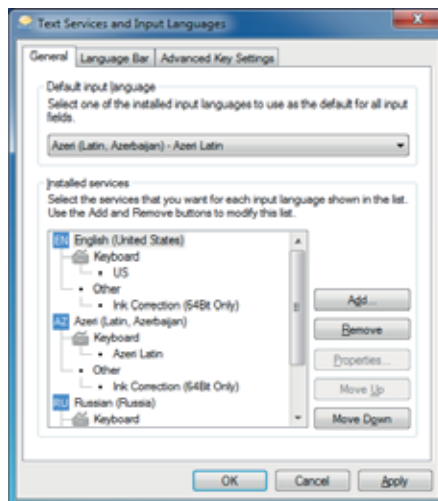
3. В открывшемся диалоговом окне перейдите на вкладку **Keyboards and Languages** (Клавиатура и языки) и щелкните по кнопке **Change keyboards...** (Изменить клавиатуру).

4. В открывшемся диалоговом окне **Text Services and Input Languages** (Языки и службы текстового ввода) в разделе **Default input language** (Язык ввода по умолчанию) выберите язык, который хотите использовать по умолчанию. Если в списках нет этого языка, выполните следующие шаги.

5. В разделе **Installed services** (Установленные службы) щелкните по кнопке **Add** (Добавить).

6. В открывшемся списке выберите сначала язык, который хотите установить, например, пункт **Arabic (Saudi Arabia)**, потом выберите пункт **Keyboard** (Клавиатура).

7. Выберите приемлемую раскладку клавиатуры, например, **Arabic (101)** и щелкните по кнопке **OK**. Язык добавится в список.



8. Для того чтобы просмотреть раскладку клавиатуры для любого языка, щелкните по кнопке Properties (Свойства).



Изучим сами

Переведите Панель управления в режим просмотра Large icons (Крупные значки). Используя значок Mouse (Мышь), откройте диалоговое окно Mouse Properties (Свойства мыши). Просмотрите вкладки окна и измените параметры мыши (например, скорость перемещения указателя, его внешний вид).

Проверьте себя

1. Что такое Панель управления и из чего она состоит?
2. Что такое раскладка клавиатуры и какая раскладка активна на вашем компьютере в данный момент?
3. Какая комбинация клавиш используется на вашем компьютере для перехода от одной языковой раскладки к другой?

ключевые слова

- Служебные программы
- Сектор
- Дорожка
- Фрагментация
- Дефрагментация

8. Служебные программы. Дефрагментация

- Что такое программное обеспечение, и на какие виды оно делится?
- Какие программы относят к системному программному обеспечению?

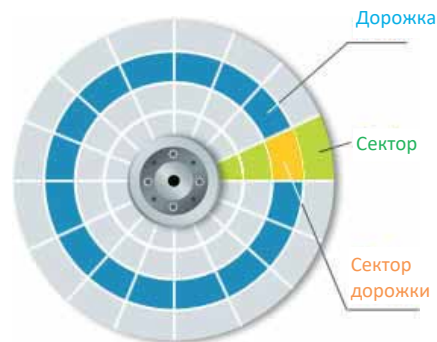
Большую часть системного программного обеспечения составляют *служебные программы*, или *утилиты*. Они предназначены для обслуживания, проверки и настройки компьютера, выполнения технических операций. Обычно эти программы управляются пользователем.

Функции служебных программ тесно связаны с возможностями операционной системы, поэтому большинство этих программ входит в ее состав. Но есть и самостоятельные служебные программы.

Все программы, которые имеются на компьютере, в том числе и данные, полученные в результате работы программ, вместе с операционной системой сохраняются на жестком диске. Поэтому жесткий диск заслуживает особого внимания. Обслуживание жесткого диска оказывает положительное влияние на общую производительность компьютера, а его отсутствие может снизить быстродействие компьютера.

Часть области на диске, где хранятся данные, называют *сектором*. Диск имеет верхнюю и нижнюю стороны, на каждой из них есть круговые *дорожки* и сектора. Каждая дорожка при форматировании делится на сектора определенного размера (обычно 512 байт). Во время обращения к диску его головки для чтения/записи проходят по дорожкам, считывая или записывая информацию в сектор или группу секторов. Таким образом, *сектор* – *наименьший участок памяти на диске, имеющий определенный размер*.

Иногда из-за механических повреждений или вследствие других причин на диске возникают испорченные сектора. Такие сектора называют *плохими* (bad sectors). Операционная система не может использовать их для хранения данных. Если часть файла, хранящегося

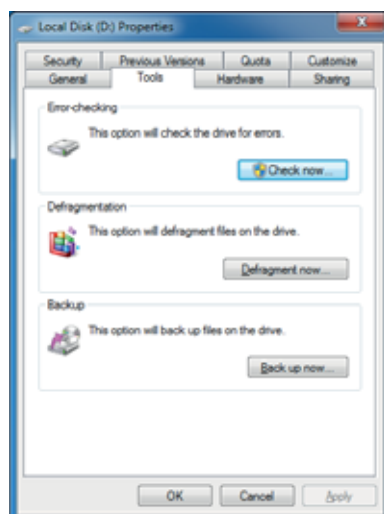


на диске, окажется на плохих секторах, операционная система не сможет прочитать файл и выдаст сообщение об ошибке (File system error). С помощью программы **ScanDisk**, имеющейся в составе операционной системы Windows, можно находить и исправлять ошибки в структуре жесткого диска.

Деятельность

Проверка жесткого диска на наличие ошибок

1. Щелкните дважды по значку Computer на рабочем столе.
2. Щелкните правой кнопкой мыши на значке диска C (или другого диска).
3. Выберите из контекстного меню команду Properties.
4. Перейдите во вкладку Tools и щелкните по кнопке Check now.
5. В открывшемся диалоговом окне щелкните по кнопке Start.
Горизонтальный индикатор показывает, какая часть проверки окончена. В конце работы будут выданы определенные статистические данные.



Обсудим:

- Нашлись ли плохие сектора на диске?
- Сколько свободного места имеется на диске?

На жестком диске компьютера имеются миллиарды секторов, поэтому операционная система работает не с отдельными секторами, а с группой их. Группа секторов называется **кластером**. Размер кластера определяется по размеру жесткого диска (обычно 4 Кб). Каждый файл занимает целое число кластеров. Операция по сохранению файла на жестком диске происходит так: операционная система сохраняет часть файла в одном свободном кластере, вторую часть во втором свободном кластере и т.д. Если запись производится на незаполненный диск, то кластеры, принадлежащие одному файлу, записываются подряд. Если диск переполнен, на нем может не найтись цельной области (последовательных кластеров), достаточной для размещения файла. Тем не менее файл все-таки запишется, если на диске много мелких областей, суммарный размер которых достаточен для записи. В этом случае файл записывается в виде нескольких фрагментов.

Если компьютер находится в использовании достаточное количество времени, многие файлы на нем оказываются "разбросаны" по диску фрагментами. Такое состояние файлов называется **фрагментацией**. При обращении к таким



разделенным на части файлам скорость работы компьютера уменьшается, поскольку поиск кластеров, в которых хранятся файлы, требует больше времени. Фрагментация может иметь место и в оперативной памяти компьютера. Это происходит при разбивке и очистке памяти. Для устранения фрагментации, то есть для **дефрагментации** диска имеются специальные утилиты. При дефрагментации разбросанные по диску файлы записываются на новые места в оптимальной форме. Для дефрагментации диска в системе Windows используется программа **Disk Defragmenter**.

Шаг за шагом

Дефрагментация жесткого диска

1. Откройте меню Start.
2. Выбрав пункт All Programs, перейдите в папку Accessories.
3. Откройте папку System Tools и выберите пункт Disk Defragmenter. Откроется соответствующее диалоговое окно.
4. Выберите необходимый диск и щелкните по кнопке Analyze disk. Начнется анализ диска. Этот процесс требует определенного времени, дождитесь его окончания.
5. Если программа предлагает выполнить дефрагментацию, щелкните по кнопке Defragment disk.

Частота проведения дефрагментации диска зависит от интенсивности вашей работы за компьютером. Если вы работаете каждый день, желательно проводить дефрагментацию один раз в неделю. Если вы за компьютером работаете редко, то дефрагментацию можно выполнять один раз в два-три месяца.

Изучим сами

Используя пункт Properties (Свойства) контекстного меню, сравните истинные размеры нескольких файлов (Size) на вашем компьютере с местом, которое они занимают на диске (Size on disk). Объясните для каждого случая разницу между истинным размером файла и размером области, которую он занимает на диске.

Проверьте себя

1. Что такое сектор и кластер?
2. Как файлы на диске хранятся в кластерах?
3. Почему на диске происходит фрагментация?
4. Определите, сколько кластеров на диске займет файл, если его объем 302 Кб?

- Disk Cleanup
- Task Manager

9. Служебные программы.

Очистка диска

- Имеются ли на вашем персональном компьютере ненужные вам программы или файлы? Что вы делаете, обнаружив их?
- Что означает зависание программы или компьютера? Что вы делаете в такой ситуации?

В процессе использования компьютера на нем неизбежно накапливаются разного рода ненужные файлы. Например, при просмотре веб-сайтов могут возникнуть временные файлы. Любой пользователь, обнаружив занимающие на диске лишнее место файлы, захочет их удалить. Для очистки диска от них предусмотрена служебная программа **Disk Cleanup**.

Деятельность

Очистка жесткого диска от ненужных программ

1. Откройте меню Start.
2. Найдите и запустите Disk Cleanup. Откроется соответствующее диалоговое окно.
3. Если в системе больше одного диска, выберите диск, который следует очистить, в противном случае перейдите к следующему пункту.
4. Выберите файлы, которые хотите удалить. Программа рассчитает и покажет, сколько дискового пространства она может освободить.
5. Щелкните по кнопке OK.

Обсудим:

- Сколько места занимают временные файлы Интернета (Temporary Internet Files) на диске, который вы хотите очистить?
- Сколько места освободилось на диске?



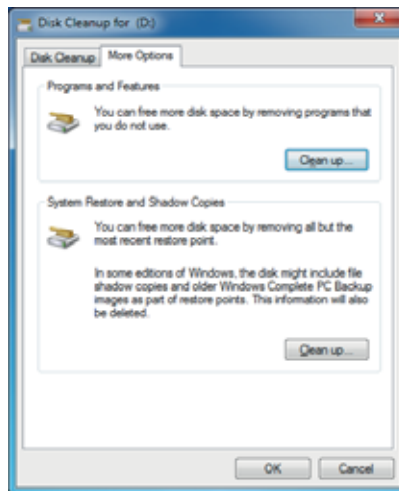
Рекомендуется запускать программу очистки дисков не менее одного раза в месяц. В диалоговом окне Disk Cleanup на странице More Options вы мо-

жете найти несколько параметров, предусмотренных для поиска компонентов, не используемых системой.

Программы, установленные когда-то на вашем компьютере, через определенное время могут оказаться ненужными, и вы захотите их удалить. Однако помните, что удаление программ, назначение которых вам точно неизвестно, может иметь нежелательные последствия!

Шаг за шагом**1****Удаление установленных программ**

1. Запустите программу Disk Cleanup.
2. Перейдите на страницу More Options.



3. В разделе Programs and Features щелкните по кнопке Clean up. Откроется список программ, установленных на вашем компьютере.
4. Щелкните по названию программы, которую хотите удалить. Имя программы выделится, а под ним появится ее описание.
5. После выделения нужной программы щелкните по кнопке Uninstall/Change. На экране появится уведомление о подтверждении операции удаления.
6. Если вы действительно хотите удалить выделенную программу из системы, щелкните по кнопке OK, в ином случае – по кнопке Cancel.

Иногда программа, с которой вы работаете, зависает, то есть перестает отвечать на ваши команды. Нажатие разных клавиш клавиатуры не дает никаких результатов. Операционная система пытается сама найти и решить проблему.

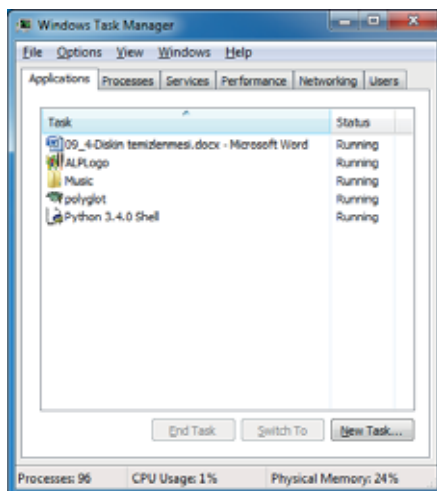
Но если у вас нет времени ждать, то при помощи служебной программы **Task Manager** вы можете завершить работу зависшей программы. Если этот метод и сэкономит ваше время, имейте в виду, что все несохраненные данные будут утеряны. Если документ, с которым вы работаете, для вас очень важен, подождите несколько минут, предоставив возможность операционной системе самой решить проблему.

Шаг за шагом

2

Завершение работы зависшей программы

1. Откройте контекстное меню панели задач. Для этого щелкните правой кнопкой мыши на свободном месте панели. Выберите из меню пункт **Start Task Manager** (или нажмите комбинацию клавиш <Ctrl+Shift+Esc>). Откроется соответствующее диалоговое окно.



2. Перейдите во вкладку **Applications**. Здесь отражается список выполняющихся на данный момент программ. Обратите внимание на состояние (статус) программ. Работающие на данный момент программы имеют статус, как правило, "Running" ("Работает"), а зависшие программы – "Not responding" ("Не отвечает").
3. Запустите программу **Paint** и проследите за изменениями во вкладке **Applications**.
4. Найдите в списке строку с названием программы **Paint** и щелкните по кнопке **End Task**. Выполнение программы прекратится и ее имя удалится из списка.

Следует иметь в виду, что эта операция проводится, как правило, с зависшими программами.

Изучим сами

Используя вкладку Processes (Процессы) диалогового окна программы Task Manager, узнайте, сколько процессов выполняется на данный момент на вашем компьютере, какую часть памяти они занимают. Определите, какой процесс занимает больше всего память. Запустите произвольный веб-браузер и проследите за изменением числа процессов во вкладке.

Name	Status	CPU	Memory	Disk	Network
Apps (2)					
Google Chrome		0%	54.6 MB	0 MB/s	0 Mbps
Task Manager		0.1%	12.9 MB	0 MB/s	0 Mbps
Background processes (57)					
ABBYY network license server (3...		0%	11.3 MB	0 MB/s	0 Mbps
ACMON (32 bit)		0%	0.1 MB	0 MB/s	0 Mbps
ActivateDesktop		0%	0.6 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service (...		0%	0.5 MB	0 MB/s	0 Mbps

Проверьте себя

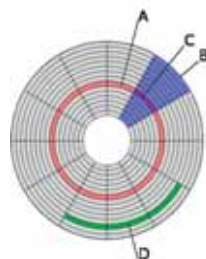
1. Вследствие чего на компьютере могут появиться ненужные файлы?
2. Объясните назначение программы Disk Cleanup.
3. Как убедиться, что работающая программа зависла?
4. Каким способом можно убрать из оперативной памяти компьютера зависшую программу?

ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите основные характеристики компьютера. Как можно узнать параметры компьютера?
2. Что используют для сокращения разницы между частотой шины, соединяющей процессор и память, и рабочей частотой процессора?
A) ядро В) кэш-память С) виртуальная память
D) жесткий диск Е) постоянная память
3. Как называется наименьший блок памяти определенного объема на диске?
4. На жестком диске записаны три файла объемом 305 Кб, 604 Кб и 2340 Кб. У каких файлов размер будет соответствовать занимаемому на диске месту?
5. Какой из следующих компьютеров оснащен как жестким диском, так и SSD-накопителем?
 - 1) Intel® Core i5-9400F 2666 MHz | H310 Chipset | DDR4 16 GB | SSD 240 GB | NVIDIA® GeForce® GTX 1050Ti 4 GB
 - 2) Intel® Core™ i9-9900K 2400 MHz | Z390M Chipset | DDR4 32 GB | M2 SSD 512 GB | HDD 2 TB | NVIDIA® GeForce® GTX1660Ti 6 GB
 - 3) Intel® Core™ i9-12900K 3600 MHz | Z690 Chipset | DDR4 64 GB | SSD 1 TB NVMe | NVIDIA® GeForce® RTX 3090 24 GB

6. Установите соответствие.

- | | |
|------|-------------------|
| 1) А | а) кластер |
| 2) В | б) сектор |
| 3) С | в) дуга |
| 4) D | д) дорожка |
| | е) сектор дорожки |



7. Что такое Панель управления?

- A) инструмент, позволяющий управлять перемещением курсора по экрану
- B) панель, расположенная на задней стороне корпуса компьютера
- C) панель, отображающая информацию о состоянии устройства
- D) программное обеспечение для управления внешним видом и работой операционной системы, для настройки параметров оборудования
- E) лицевая сторона корпуса компьютера, на которой расположены рычаги управления, тумблеры и индикаторы



ЭЛЕКТРОННЫЕ ТАБЛИЦЫ

10. Абсолютные и относительные ссылки
11. Функции в электронной таблице
12. Визуализация данных. Диаграммы
13. Решение задач



- Формула
- Ссылка
- Относительная ссылка
- Абсолютная ссылка
- Смешанная ссылка

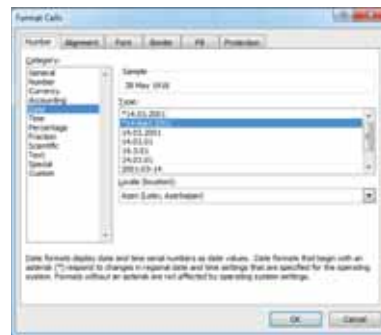
10. Абсолютные и относительные ссылки

- Для каких целей используют электронные таблицы?
- Как записывается формула в электронной таблице?

В программах для управления электронными таблицами (табличных процессорах) есть возможность работы с данными разного типа – символьными (текстами), числовыми, логическими и данными времени и даты. От типа данных в ячейке зависит, какие именно операции можно производить с ней. Например, если ввести в ячейку число 45440 и установить формат ячейки как текстовый, то содержимое ячейки будет воспринято как последовательность символов 4, 5, 4, 4, 0. Если для ячейки установлен числовой формат, то электронная таблица воспримет запись в ячейке как число. Если к ячейке применен формат даты, число в ячейке будет воспринято как 28 мая 2024 года (число дней с 01.01.1900 до 28.05.2024 составляет 45440).

Деятельность

1. Запустите электронную таблицу Microsoft Excel.
2. Введите в произвольную ячейку любое число, например, **6723**. Нажмите клавишу <Enter>.
3. Перейдите заново в заполненную ячейку. На панели инструментов выберите из списка **General** в разделе **Number** пункт **Date** и проследите за изменением содержимого ячейки.
4. Щелкните по кнопке **Format** на панели инструментов и из меню выберите пункт **Format Cells**. Откроется соответствующее диалоговое окно.
5. Перейдите во вкладку **Number** этого окна. Обратите внимание на типы данных, указанных в разделе **Category**.
6. Выбирая элементы списка, ознакомьтесь с образцами представления данных.



Обсудим:

- Сколько типов имеется у формата данных **Date** (Дата)?
- Какое число следует ввести в ячейку, чтобы при изменении формата на **Percentage** (Процентный) содержимое ячейки показывало **45%**?

Электронная таблица предназначена, в первую очередь, для автоматизации вычислений. Для этого в ячейки таблицы вводятся **формулы**. Особенность электронных таблиц именно в том, что при изменении одних данных связанные с ними формулы автоматически пересчитываются. Например, как только вы измените в ячейке количество проданного товара, стоимость по каждому товару будет автоматически пересчитана. Создав электронную таблицу с большим количеством данных и изменив значение одного или нескольких из них, можно пронаблюдать, как изменится результат.

Ввод любой формулы начинается со знака “=”. Если этот знак отсутствует, введенная формула будет восприниматься как текст. Формула может содержать числовые данные, символы, различные функции, а также адреса объектов таблицы. Например, =C4*D4 (в этой формуле звездочка “*” означает умножение; знак “=” указывает на то, что это формула, а не текст). Формулы с адресами ячеек можно сравнить с математическими уравнениями: вместо адресов ячеек в них используются переменные.

Адреса, которые используют в формулах, называют **ссылками**. Ссылки дают возможность связывать произвольные ячейки электронной таблицы и обрабатывать данные в ней. После записи формулы в ячейку электронная таблица отражает в ней не саму формулу, а результат вычисления по формуле. Однако когда ячейка активирована, формула для этой ячейки отображается в строке формул в верхней части экрана, и при необходимости ее можно отредактировать.

Формулу из одной ячейки можно скопировать в другие ячейки. При копировании формулы в другую ячейку она изменяется автоматически: имя столбца и номер строки в адресе ячейки в формуле изменяются в зависимости от того, на какое количество столбцов и строк была перемещена формула из исходной ячейки. Например, если ячейка B2 содержит формулу =C2 + 1, то при копировании формулы из этой ячейки формула изменится следующим образом.

	A	B	C	D
1	=B1 + 1		=D1 + 1	
2		=C2 + 1		=E2 + 1
3	=B3 + 1			
			=D4 + 1	

Перемещение формул. Зачастую для решения схожих задач используются одни и те же формулы. Копию определенной формулы можно использовать, чтобы не вводить ее каждый раз повторно в разные ячейки таблицы. Существует более удобный способ выполнить операцию копирования формулы для ячеек в той же строке или столбце. Для этого сначала выбирается ячейка, которую необходимо скопировать (это видно по жирной черной рамке вокруг ячейки). Затем указатель

мышью наводится на маленький квадратик в правом нижнем углу рамки (указатель примет форму +) и, удерживая нажатой левую кнопку мыши, перемещается указатель по строке или столбцу. При отпуске кнопки мыши, формула в исходной ячейке будет применена ко всем выделенным ячейкам. Этим же методом можно легко и быстро перенести значение любой ячейки в другие ячейки.

Относительные ссылки. Если при копировании или перемещении формулы из одной ячейки в другую в электронной таблице адреса ячеек, используемых в формуле – ссылках, изменяются автоматически, то такие ссылки называются *относительными ссылками*. Относительная ссылка записывается в обычном виде, например: F3, E7.

	A	B	C	D	E
1			=A1 * B1		
2				=B2 * C2	
3					=C3 * D3

Если формула, содержащая относительные ссылки на ячейки A1 и B1, перемещается из ячейки C1 в ячейку D2, то имена столбцов и номера строк в формуле сместятся на один шаг вправо и вниз соответственно. Если формулу переместить в ячейку E3, имена столбцов и номера строк в формуле сдвинутся на два шага вправо и вниз соответственно и так далее.

Абсолютные ссылки. В некоторых случаях нет необходимости менять ссылки при переносе формулы из одного места в другое. В таких случаях в электронной таблице используются *абсолютные ссылки*. В абсолютной ссылке букве столбца и номеру строки предшествует символ \$, например: \$F\$3, \$E\$7.

	A	B	C	D	E
1			=\$A\$1 * \$B\$1		
2				=\$A\$1 * \$B\$1	
3					=\$A\$1 * \$B\$1

Если формула, содержащая абсолютную ссылку на ячейки \$A\$1 и \$B\$1, перемещается из ячейки C1 в ячейку, отличную от ячейки C1, имена столбцов и номера строк в формуле не изменятся.

Смешанные ссылки. Если в формуле изменена одна из двух частей ссылки – буква столбца или номер строки, то используется *смешанная ссылка*. В этом случае символ \$ ставится только перед той частью ссылки, которая должна остаться неизменной. Например: \$F3, E\$7. Так, если в ячейку B2 записана формула =C\$2+\$C3+C\$4, то в результате перемещения ее из этой ячейки формула изменится так:

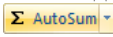
	A	B	C	D
1	=C\$2+\$C2+B\$4		=C\$2+\$C2+D\$4	
2		=C\$2+\$C3+C\$4		=C\$2+\$C3+E\$4
3	=C\$2+\$C4+B\$4			
4			=C\$2+\$C5+D\$4	

Шаг за шагом

Использование в электронной таблице относительных и абсолютных ссылок

1. Запустите электронную таблицу. Откроется пустой документ (рабочая книга).
2. Введите в ячейку A1 текст: **1 доллар США = ... манат**.
3. В ячейку A2 запишите текущий курс доллара США (например, 1,5).
4. Остальные ячейки заполните соответственно образцу.

	A	B	C	D	E	F	G
1	1 доллар США =... манат						
2	1,5						
3		№	Наименование товара	Цена (USD)	Количество	Стоимость (USD)	Стоимость (AZN)
4		1	Системный блок	200	2		
5		2	Монитор	250	2		
6		3	Клавиатура	30	2		
7		4	Мышь	15	2		
8		5	Принтер	90	1		
9		6	Сканер	32	1		
10			ИТОГО				

5. Введите в ячейку F4 формулу =D4*E4 и нажмите клавишу <Enter>. В ячейке появится результат вычисления по формуле – число 400, и курсор перейдет в соседнюю ячейку.
6. Снова перейдите в ячейку F4. Она будет выделена жирной рамкой. В программе Excel так выделяются активные ячейки. Наведите указатель мыши на небольшой квадратик в правом нижнем углу рамки. Указатель примет вид +. Нажав и удерживая левую кнопку мыши, протяните мышь до 9-й строки.
7. Формула ячейки F4 будет скопирована в диапазон ячеек F5: F9. Во всех ячейках выделенного диапазона будут отражены результаты вычисления по этим формулам. Переходя в эти ячейки, проследите, как меняются ссылки в формулах.
8. Запишите в ячейку G4 формулу =F4*\$A\$2. В ячейке появится стоимость системного блока в манатах. Используя инструкцию 6-го шага, скопируйте формулу в диапазон ячеек G5: G9.
9. Для того чтобы вычислить конечную стоимость устройств в долларах США, выделите диапазон ячеек F4: F10. Щелкните по кнопке  на панели инструментов или по соответствующей кнопке в меню Formulas. Будет вычислена конечная стоимость всех товаров, которая отразится в ячейке F10.
10. Аналогично вычислите стоимость товаров в манатах.
11. Измените в ячейке A2 курс доллара США по отношению к манату и проследите за изменениями в ячейках.
12. Сохраните файл, задав ему соответствующее имя.

	A	B	C	D	E	F	G
1	1 доллар США	=...	манат				
2		1,5					
3		№	Наименование товара	Цена (USD)	Количество	Стоимость (USD)	Стоимость (AZN)
4		1	Системный блок	200	2	400	600
5		2	Монитор	250	2	500	750
6		3	Клавиатура	30	2	60	90
7		4	Мышь	15	2	30	45
8		5	Принтер	90	1	90	135
9		6	Сканер	32	1	32	48
10			ИТОГО			1112	1668

Изучим сами

Чтобы изменить тип ссылки в формуле, записанной в любой ячейке, не нужно где-то добавлять или удалять символ «\$». Для этого выделите ячейку с формулой. В строке формул выберите ссылку, которую хотите изменить. Нажимайте клавишу <F4>, чтобы переключаться с одного типа ссылок на другой. Посмотрите, как меняется ссылка в строке формул.

Проверьте себя

1. Какие типы данных используются в электронных таблицах?
2. Что такое абсолютная ссылка в формуле и как она обозначается?
3. Создайте таблицу по данному образцу. Для этого
 - a) в ячейке A1 укажите текущий год;
 - b) запишите в соответствующие ячейки таблицы имена, фамилии, даты рождения ваших друзей;
 - c) для вычисления возраста первого вашего друга в списке в ячейку D4 впишите соответствующую формулу. Скопируйте эту формулу в остальные ячейки столбца D.

	A	B	C	D
1	2024			
2				
3	Фамилия	Имя	Год рождения	Возраст
4	Аббаслы	Эльхан	2009	15
5	Ализаде	Гюнай	2008	
6	Ягублу	Рустам	2010	

4. Укажите в ячейке A2 скорость автомобиля, в ячейке B2 время, которое он затрачивает на путь. Вычислите в ячейке C2 с помощью формулы путь, пройденный автомобилем ($s=vt$). Меняя значение скорости (содержимое ячейки A2) и времени (содержимое ячейки B2), проследите, как меняется пройденный путь (содержимое ячейки C2).

11. Функции в электронной таблице

- Что вы понимаете под функцией?
- Какие математические функции вы знаете?

В электронных таблицах можно более «компактно» записывать длинные формулы. Например, если вам нужно просуммировать числа во всех ячейках от A1 до A10, удобнее записать формулу

$$=A1+A2+A3+A4+A5+A6+A7+A8+A9+A10 \text{ в виде } = \text{SUM}(A1:A10).$$

Такие функции, как SUM, называют **встроенными функциями**. Программы для работы с электронными таблицами имеют множество таких функций.

Встроенные функции работают по принципу «черного ящика»: функции передаются *входные данные* или аргументы, и функция вычисляет некоторый результат или возвращает выходные данные. При работе с функциями основная задача пользователей – правильно отобразить аргументы. Процесс получения результата скрыт от пользователя. Пользователь не видит, какие именно формулы используются для расчета, в ячейке отражается только результат, возвращаемый функцией.


Ввод функции также начинается с символа “=”. После имени функции аргументы функции отображаются в круглых скобках. Аргументами могут быть выражения, состоящие из чисел, текста, адресов ячеек и других функций. Если аргументов несколько, между ними ставится точка с запятой (;). Например, в формулах =MIN(A1:A10; C1:C10), =SUM(A1:A10; C1:C10) аргументы или входные данные функций MIN и SUM являются ссылками на диапазоны A1:A10 и C1:C10. Выходные данные функции MIN – это наименьшее число в исходном диапазоне, а выходные данные функции SUM – сумма всех чисел в указанном диапазоне.

В таблице ниже перечислены наиболее часто используемые функции Excel.

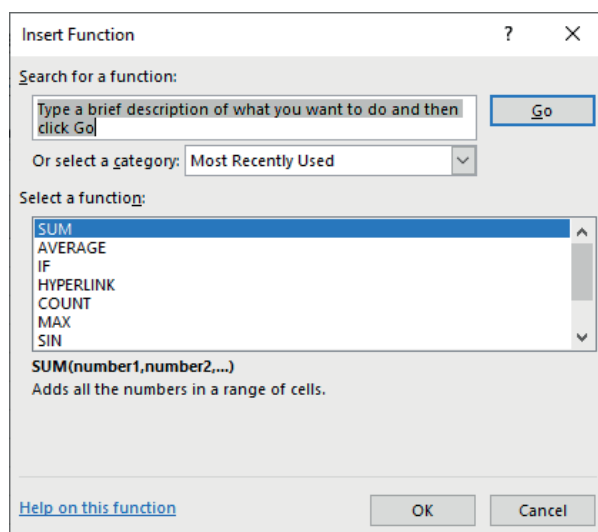
Функция	Объяснение
SUM(B2: B5)	Вычисляется сумма числовых значений в диапазоне B2: B5.
AVERAGE(B2: B5)	Вычисляется среднее арифметическое значение для диапазона B2: B5.
MAX(B2: B5)	Вычисляется максимальное значение для диапазона B2: B5.
MIN(B2: B5)	Вычисляется минимальное значение для диапазона B2: B5.
COUNT(B2: B5)	Вычисляется количество чисел в диапазоне B2: B5.

Существует несколько способов вставить функцию в формулу. Однако независимо от метода ввода необходимо учитывать два фактора: правильный ввод имени функции и правильное указание аргументов функции. Порядок указания аргументов также важен для многих функций. То есть аргументы таких функций должны быть введены в правильном порядке.

Во многих случаях *ввод функции в формулу вручную* является наиболее эффективным методом. Но это можно сделать и без использования клавиатуры. Для этого необходимо выполнить одно из следующих действий.

- Нажать кнопку  Insert Function.
- Выбрать команду **Formulas** ► **Insert Function**.
- Нажать комбинацию клавиш <Shift+F3>.

После выполнения одного из перечисленных действий откроется диалоговое окно **Insert Function**. В разделе **Select a function** следует выбрать нужную функцию и нажать на кнопку **OK**.



Если выбранная функция имеет аргументы, откроется диалоговое окно **Function Arguments**.



После ввода аргументов в ячейки Number1 и Number2 и нажатия кнопки ОК выбранная функция будет добавлена в формулу.

Многие функции в качестве аргументов могут принимать ссылки на диапазон ячеек. Например, в формулах $=\text{SUM}(A2:C7)$, $=\text{AVERAGE}(A2:C7)$, $=\text{MIN}(A2:C7)$ и $=\text{MAX}(A2:C7)$ в качестве аргумента для функций SUM, AVERAGE, MIN и MAX используется диапазон ячеек A2:C7. При изменении диапазона ячеек программа автоматически корректирует ссылки на диапазоны во всех этих формулах. Например, если в диапазон A2:C7 добавляется новая строка, формулы изменятся на $=\text{SUM}(A2:C8)$, $=\text{AVERAGE}(A2:C8)$, $=\text{MIN}(A2:C8)$ и $=\text{MAX}(A2:C8)$.

В некоторых случаях удобно ссылаться на целый столбец или строку как на аргумент функции. Например, формулы $=\text{SUM}(A:A)$, $=\text{AVERAGE}(A:A)$, $=\text{MIN}(A:A)$ и $=\text{MAX}(A:A)$ вычисляют соответственно сумму, среднее, минимальное и максимальное значения чисел в столбце A. Следует учесть, что Excel не использует в вычислениях ячейки столбца A, не содержащие значения. Однако при вводе новых значений в этот столбец формулы автоматически пересчитываются.

В электронной таблице число 10 записано в ячейку A1, формула $=A1/2$ – в ячейку B1 и формула $=\text{SUM}(A1:B1)*2$ – в ячейку C1. Какое значение будет в ячейке C1?

1

Решение: Сначала вычисляется значение в ячейке B1: $B1 = A1/2 = 10/2 = 5$. Значение в ячейке C1 равно удвоенной сумме значений в ячейках с A1 по B1: $=\text{SUM}(A1:B1)*2 = (A1 + B1) * 2 = (10 + 5) * 2 = 30$.

Ответ: 30.

В электронной таблице значение формулы $=\text{SUM}(A1:A4)$ равно 13, а значение формулы $=\text{AVERAGE}(A1:A5)$ равно 3. Каким будет значение формулы $=\text{SUM}(A1:A5)$?

2

Решение: Поскольку сумма числовых значений в диапазоне ячеек A1:A4 равна 13, а среднее арифметическое значение для диапазона ячеек A1:A5 равно 3, значение в ячейке A5 должно быть 2:

$$\begin{aligned}(A1 + A2 + A3 + A4 + A5) / 5 &= 3 \\ (13 + x) / 5 &= 3 \\ x &= 2\end{aligned}$$

Тогда значение формулы $=\text{SUM}(A1:A5)$ будет $13 + 2 = 15$.

Ответ: 15.

Список аргументов функции может быть пустым, то есть функция может не иметь аргументов. Примером таких функций является функция PI (формула =PI()), которая возвращает значение числа π с точностью до 15 цифр, функция TODAY (формула =TODAY()), которая возвращает текущую дату, и функция RAND (формула =RAND()), которая возвращает случайное число в интервале от 0 до 1.



Даже в том случае, если в функции нет аргументов, все равно после имени функции обязательно должны стоять круглые скобки. В противном случае Excel не распознает функцию и выдаст предупреждение об ошибке.

Большинство функций Excel принимают один аргумент. Например, математические функции SIN, COS и TAN (формулы =SIN(число), =COS(число) и =TAN(число)) имеют один аргумент. Они вычисляют синус, косинус и тангенс угла. Аргументами этих функций являются величины угла, выраженные в радианах. Другая функция с аргументом RADIANS преобразует градусы в радианы.

Если в функции более одного аргумента, иногда порядок их указания не важен, а иногда – имеет значение. Например, такие функции, как DATE, TIME, возвращают правильный результат только в том случае, если аргументы заданы в правильном порядке, но в таких функциях, как SUM, PRODUCT, порядок указания аргументов не имеет значения.

В электронной таблице значение формулы =AVERAGE(A6: C6) равно -2. Если значение ячейки D6 равно 5, каким будет значение =SUM(A6: D6)?

3

Решение:

По определению среднего арифметического значения
 $AVERAGE(A6: C6) = SUM(A6: C6) / 3 = -2$.

Следовательно, $SUM(A6: C6) = -6$. Тогда $SUM(A6: D6) = SUM(A6: C6) + D6 = -6 + 5 = -1$

Ответ: -1.

В электронной таблице в ячейке C3 записана дата 15.03.2023. Какой будет результат расчета по формуле =C3-19?

4

Решение:

Так как в феврале 2023 года 28 дней, то если вернуться на 19 дней назад от 15.03.2023, то получим дату 24.02.2023.

Ответ: 24.02.2023.

Изучим сами

Выделите столбцы В, С, D и Е электронной таблицы для функций ABS, SQRT, SIN и COS соответственно, а столбец А – для аргументов этих функций. Вычислите значения функций для 10 различных значениях аргумента. Что вы знаете об этих функциях?

Проверьте себя

1. Значение формулы =SUM(B1:B2) в электронной таблице равно 5. Если значение формулы =AVERAGE(B1:B3) равно 3, каково будет значение ячейки B3?
2. В электронной таблице значение формулы =SUM(A1:B1) равно 12, а значение формулы =SUM(D1:E1) равно 7. Если значение формулы =AVERAGE(A1:E1) равно 5, каким будет значение ячейки C1?
3. Приведен фрагмент электронной таблицы. Каким будет значение формулы =SUM(A2:B2; E2:F2)?

	A	B	C	D	E	F
1	3	4	5	7	1	2
2	7	5	7	4	0	9
3	3	6	4	8	6	5
4	4	5	8	9	6	0

4. Формулу =SUM(\$A\$2:\$B\$2; E2:F2), записанную в ячейку G2 электронной таблицы из предыдущего задания, переместили в ячейку G3. Каким будет значение этой ячейки?
5. В ячейке A1 электронной таблицы записана дата 01.01.2023. Каким будет результат вычисления формулы =A1+30?
6. В электронной таблице в ячейке B2 записано 8:30, а в ячейке C2 — 4:30. Каким будет результат вычисления формулы =B2+C2?

- Диаграмма
- Линейная диаграмма
- Круговая диаграмма
- Гистограмма
- Легенда

12. Визуализация данных. Диаграммы

- Как можно построить диаграмму в текстовом редакторе?
- С какими видами диаграмм вы знакомы?

Деятельность

1. Запустите электронную таблицу Microsoft Excel.
2. Постройте таблицу, связанную с численностью населения Азербайджанской Республики. Название таблицы запишите в ячейку А1, а саму таблицу разместите в диапазоне ячеек А3: J5.

Распределение численности населения по возрасту (на начало 2023 года)

	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70 и выше	Всего
Численность населения, тыс. человек	1373,7	1572,4	1400,0	1804,4	1345,8	1190,6	977,5	462,7	10127,1
Численность населения, в процентах	13,6	15,5	13,8	17,8	13,3	11,7	9,7	4,6	100

3. Задайте файлу соответствующее имя (например, **Население Азербайджана**) и сохраните его.

Обсудим:

- Какова численность населения страны в возрасте до 50 лет?
- При помощи какой диаграммы – столбчатой или круговой – удобнее представить данную таблицу?

Электронные таблицы предназначены в основном для быстрой и качественной обработки однотипных числовых данных большого объема, но представление данных в графической форме делает их более наглядными и понятными. Основным средством представления табличных данных в графической форме являются диаграммы.

Диаграммы предусмотрены для наглядного представления данных в графической форме. Они позволяют лучше анализировать данные, сравнивая их, выявлять невидимые в таблице закономерности.

В электронных таблицах можно строить *линейные, круговые, столбчатые* и другие диаграммы.



Линейная диаграмма (график)



Круговая диаграмма

Линейная диаграмма, или **график** – это тип диаграммы, на котором значения величин соединены между собой линиями, что позволяет проследить за изменениями величин при переходе от одной точки к другой.

Круговая диаграмма – тип диаграммы, представляющей собой круг, разделенный на сектора. Относительная величина каждого значения изображается в виде сектора круга, площадь которого соответствует вкладу этого значения в сумму значений.



Столбчатая диаграмма

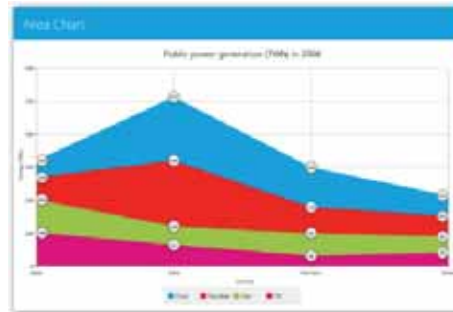


Диаграмма с областями

Столбчатая диаграмма, или **гистограмма** – тип диаграммы, состоящий из вертикальных или горизонтальных столбцов разной высоты (длины). Высоты этих столбцов соответствуют определенным значениям величин, что позволяет сравнивать их.

Диаграмма с областями – это тип диаграммы, в которой некоторые области заштрихованы или выделены определенным образом, чтобы отличить (отделить) одну группу данных от другой.

В электронной таблице данные, расположенные на одной строке или в одном столбце, называют **рядом** данных.

Диаграмма – сложный объект, который содержит несколько элементов: **ряды данных, оси, заголовок, легенду, область построения**.

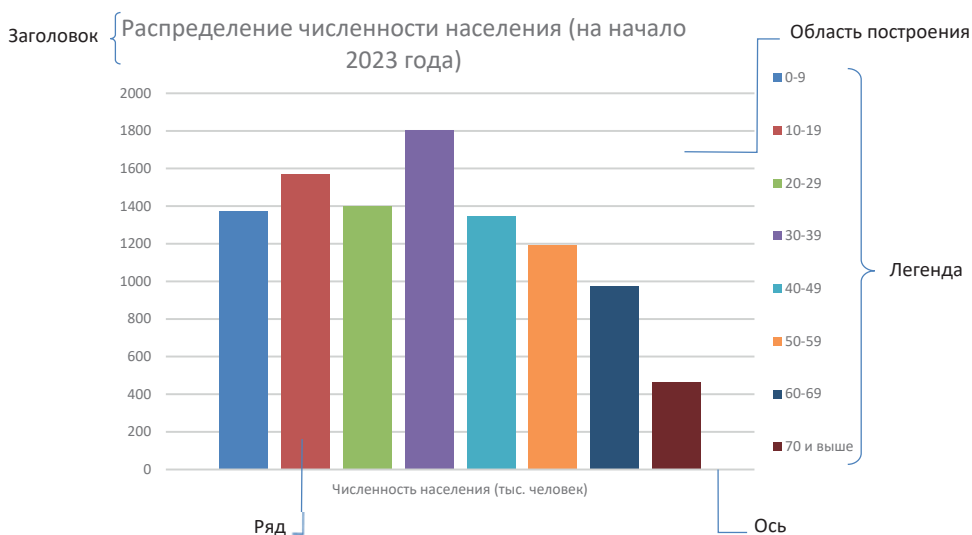


Диаграмма может быть построена как по одному **ряду**, так и по нескольким рядам. Для выделенного диапазона ячеек построение диаграммы ведется по нескольким рядам данных. В этом случае в качестве каждого ряда принимается соответствующая строка или столбец выделенного диапазона. **Заголовком** служит текст, задаваемый пользователем, который, как правило, размещается над диаграммой. При выводе диаграмм можно добавить **легенду** – список названий рядов (обозначений переменных). **Область построения**, ограниченная **осями**, предназначена для размещения рядов данных. Для удобства анализа результатов на область построения может быть добавлена сетка.

Чтобы построить диаграмму на основе заданной таблицы:

1. Сначала следует выбрать диапазон ячеек этой таблицы, содержащий значения, которые будут отображаться на диаграмме.
2. Затем в разделе Charts меню Insert нажать кнопку, соответствующую типу диаграммы. Будут предложены различные варианты выбранной диаграммы.
3. После выбора подходящего варианта в рабочей области в небольшой рамке будет отображена диаграмма. Одновременно на панели инструментов будут отображены параметры настройки макета и стиля диаграммы. При необходимости можно внести изменения в диаграмму – добавить название, подписи к осям, изменить цветовую гамму, стиль диаграммы.

Шаг за шагом

1

Построение диаграммы

1. Для построения диаграммы выделите диапазон ячеек.
2. В меню Insert в блоке Charts щелкните по кнопке Pie.

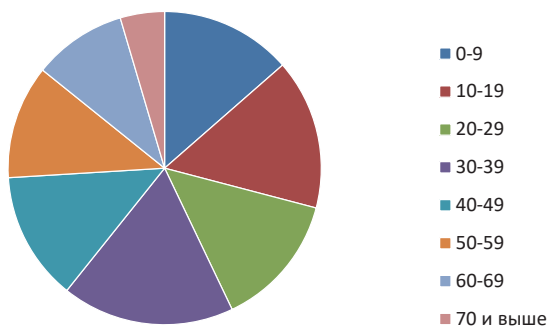


3. Будут предложены различные типы круговых диаграмм.



4. Выберите первый вариант. На рабочем поле в небольшой рамке будет построена круговая диаграмма. Если легенда, находящаяся справа от диаграммы, отображена не полностью, используя маркеры рамки, можно растянуть рамку до тех пор, пока легенда не отобразится полностью.

Численность населения, тыс. человек



5. Сохраните файл под соответствующим именем.

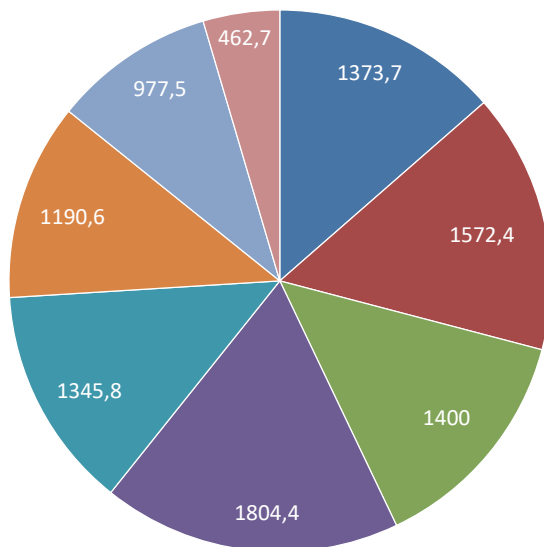
Настройка макета и стиля диаграммы

1. Одновременно на панели инструментов будут отображены варианты для выбора макета и стиля диаграммы.



2. Выберите первый макет (Layout 1) и десятый стиль (Style 10). В каждом секторе диаграммы будут указаны возраст группы и доля этого населения в процентах.
3. Щелкните по заголовку "**Численность населения (тысяч человек)**". Это поле перейдет в режим редактирования.
4. Введите в это поле текст "**Распределение численности населения Азербайджанской Республики по возрасту (на начало 2023 года)**".

Распределение численности населения Азербайджанской Республики по возрасту (на начало 2023 года)



5. Сохраните файл под существующим именем.

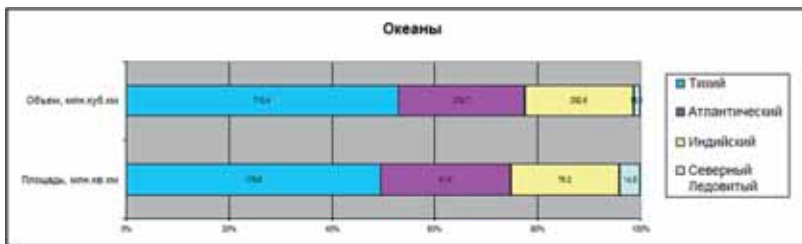
Изучим сами

В блоке «Шаг за шагом» сравните процентные значения на последней диаграмме со значениями в строке «Население (в процентах)» таблицы, приведенной в начале урока. Попробуйте выяснить, почему округлены значения в диаграмме, и попытайтесь восстановить дробную часть чисел.

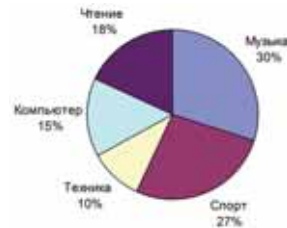
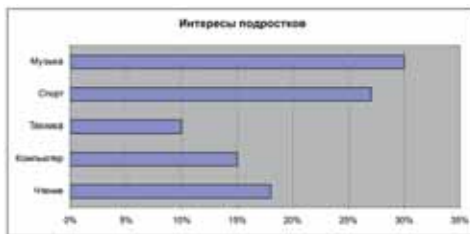
Проверьте себя

1. Из каких объектов состоит диаграмма?
2. Какие основные типы диаграмм имеются в электронных таблицах?
3. Постройте следующую таблицу и соответствующую ей диаграмму.

	А	В	С
1	Океаны		
2	Название	Площадь, млн. кв. км	Объем, млн. куб. км
3	Тихий	178,6	710,4
4	Атлантический	91,6	329,7
5	Индийский	76,2	282,6
6	Северный Ледовитый	14,8	18,1



4. Рассмотрите диаграммы и ответьте на вопросы.
 - Какая информация отображена на диаграммах? Как вы узнали об этом?
 - Как можно представить эту информацию в форме таблицы?
 - Что больше всего привлекает подростков?



13. Решение задач

1

Приведен фрагмент электронной таблицы. Каким будет значение в ячейке C2, если формулу из ячейки C1 переместить в ячейку C2?

	A	B	C
1	10	30	=A1 + B1 - 10
2	20	40	

Решение: Формула =A1 + B1 - 10 в ячейке C1 превратится в =A2 + B2 - 10 при перемещении ее в ячейку C2. Тогда значение в этой ячейке будет $20 + 40 - 10 = 50$.

Ответ: 50.

2

Приведен фрагмент таблицы. Каким будет значение в ячейке D3, если формулу из ячейки C2 переместить в ячейку D3?

	A	B	C	D
1	10	20	30	50
2	40	10	=A1 + C1 - B1	60
3	20	30		

Решение: Если формулу =A1+C1-B1 из ячейки C2 переместить в ячейку D3, она примет вид =B2+D2-C2. Тогда значение в этой ячейке будет $10 + 60 - (10 + 30 - 20) = 50$.

Ответ: 50.

3

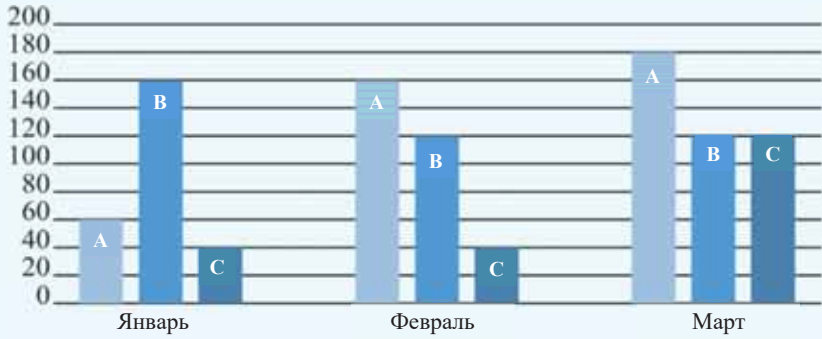
Приведен фрагмент электронной таблицы. Формулу из ячейки A3 переместили в ячейку C2. Адреса ячеек в формуле автоматически изменились при переносе. Каким будет числовое значение формулы в ячейке C2?

	A	B	C	D	E
1	40	4	100	70	9
2	30	3		60	10
3	=B3*B\$2	2	300	50	11
4	10	1	400	40	12

Решение: Когда формула =B3*B\$2 перемещается из ячейки A3 в ячейку C2, она принимает вид =B2*D\$2. Тогда значение в этой ячейке будет $3 \times 60 = 180$.

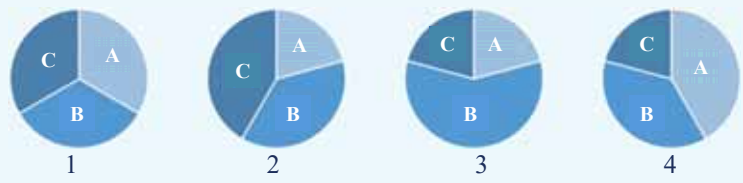
Ответ: 180.

На диаграмме показан объем продукции А, В, С, выпущенной в каждом месяце 1-го квартала.



Какая из круговых диаграмм правильно отражает долю продуктов, выпущенных в первом квартале?

4



Решение: На основании первой диаграммы можно, просуммировав данные, найти объем выпуска каждого продукта за 3 месяца:

$$\begin{aligned}
 A & 60 + 160 + 180 = 400; \\
 B & 160 + 120 + 120 = 400; \\
 C & 40 + 40 + 120 = 200.
 \end{aligned}$$

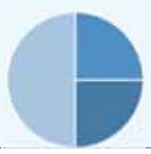
Это означает, что на правильной диаграмме сектора А и В должны быть одинакового размера (400), а сектор С должен быть в два раза меньше каждого из них (200). Такая диаграмма показана в варианте 4.

Ответ: 4.

Приведен фрагмент таблицы. Какое число следует записать в ячейку С1, чтобы диаграмма, построенная после расчета значений диапазона в ячейках А2: С2, соответствовала рисунку?

5

	А	В	С
1	2	4	
2	$=(B1 - A1) / 2$	$=2 - A1 / 2$	$=(C1 - A1) * 2 - 4$



Решение: Сначала вычисления производятся над ячейками с формулами, значения которых можно вычислить (кроме ячейки C1, значение которой пока неизвестно):

- Для ячейки A2 по формуле $= (B1 - A1) / 2$ получается значение 1;
- Для ячейки B2 по формуле $= 2 - A1 / 2$ получается значение 1;

Если проанализировать диаграмму, легко увидеть, что эти 1 соответствуют двум правым частям диаграммы. Следовательно, левый сектор, рассчитанный по формуле $= (C1 - A1) * 2 - 4$, соответствует значению 2. Поскольку значение ячейки A1 равно 2, получается простое уравнение: $(C1 - 2) * 2 - 4 = 2$. Отсюда получается $C1 = 5$.

Ответ: 5.

Приведен фрагмент таблицы. Какое число в результате вычислений нужно записать в ячейку B1, чтобы диаграмма, построенная на основе значений диапазона ячеек A2:D2, соответствовала рисунку?

	A	B	C	D
1	3		3	2
2	$= (C1 + A1) / 2$	$= C1 - D1$	$= A1 - D1$	$= B1 / 2$



6

Решение: Диаграмма построена на основе четырех значений диапазона ячеек A2:D2. Судя по всему, соотношение этих значений составляет 1:1:1:3. Поскольку значения ячеек A1, C1, D1 известны, вместо формул можно вписать значения ячеек диапазона A2:D2.

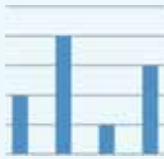
	A	B	C	D
1	3		3	2
2	3	1	1	$= B1 / 2$

Только ячейка D2 осталась незаполненной. Из приведенной диаграммы видно, что ее значение равно 1. Теперь можно найти неизвестное значение ячейки B1: $B1 = 1 \cdot 2 = 2$.

Ответ: 2.

Фрагмент таблицы представлен в режиме формул. Какая диаграмма получится на основе значений ячеек диапазона A1: D1?

	A	B	C	D
1	=C2 - B1	=B2 - C2	=B1 + C2	=(C1 - C2) * 3
2		3	2	



A)



B)



C)



D)

7

Решение: Вычисляются значения формул ячеек диапазона A1: D1.

$$B1 = 3 - 2 = 1;$$

$$A1 = 2 - 1 = 1;$$

$$C1 = 1 + 2 = 3;$$

$$D1 = 1 \cdot 3 = 3.$$

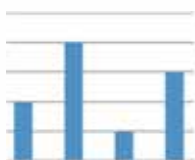
Этим значениям соответствует диаграмма в варианте C.

Ответ: C.

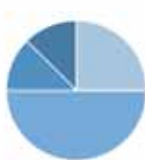
Проверьте себя

1. Приведен фрагмент электронной таблицы. Какая диаграмма получится на основе значений ячеек диапазона A1:A4?

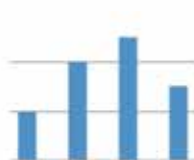
	A	B
1	=B1 + 1	1
2	=A1 + 2	2
3	=B2 - 1	
4	=A3	



A)



B)



C)



D)

2. Приведен фрагмент электронной таблицы. Какое число следует записать в ячейку C1, чтобы диаграмма, построенная после расчета значений диапазона в ячейках A2:C2, соответствовала рисунку?

	A	B	C
1	2	1	
2	=C1 - B1 * 2	=(B1 + C1) / A1	=A1 + B1



3. Приведен фрагмент электронной таблицы. Какое число следует записать в ячейку B1, чтобы диаграмма, построенная после расчета значений диапазона в ячейках A2:D2, соответствовала рисунку? Все ячейки в просматриваемом диапазоне имеют неотрицательные значения.

	A	B	C	D
1	7		4	5
2	=B2 + C2 + D2	= C2	=(A1 - D1) * (B1 - 3)	=(A1 - D1) * C1



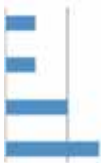
4. Приведен фрагмент электронной таблицы. Какое целое число следует записать в ячейку A1, чтобы диаграмма, построенная после расчета значений диапазона в ячейках A2:C2, соответствовала рисунку?

	A	B	C
1		$= (A1 - B2) / 2$	3
2	$= C1 * 2 - B1$	2	$= C1 + B2$



5. Приведен фрагмент электронной таблицы. Какая диаграмма получится на основе значений ячеек диапазона A2:D2?

	A	B	C	D
1		3	4	
2	$= C1 - B1$	$= B1 - A2 * 2$	$= C1 / 2$	$= B1 + B2$



A)



B)

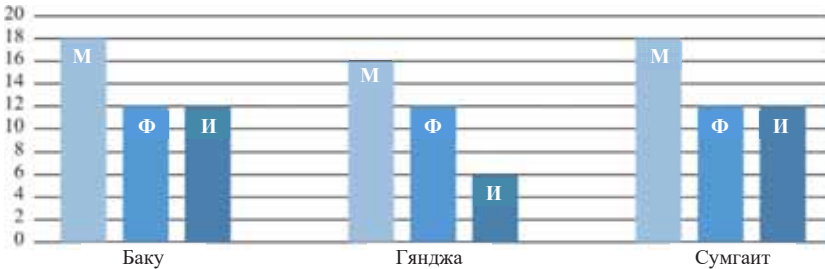


C)



D)

6. На диаграмме показано количество призеров олимпиады по информатике (И), математике (М), физике (Ф) из трёх городов Азербайджана.



Какая круговая диаграмма правильно отображает общее количество призеров по каждому предмету в этих трех городах?



A)



B)



C)



D)

ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

1. Как называется файл, созданный в программе Excel?
А) рабочая тетрадь
В) рабочий лист
С) документ
D) таблица
Е) книга
2. В чем разница между относительным, абсолютным и смешанным адресами?
3. Сколько ячеек находится в диапазоне A2: D5 электронной таблицы?
4. В электронной таблице указано общее количество топлива (в литрах), израсходованное транспортными средствами четырех транспортных компаний с 20 по 22 августа. Какая из этих четырех транспортных компаний имела наименьший средний расход топлива за 3 дня?

Компания	20 августа	21 августа	22 августа	Всего
А	2121	1600	4350	8071
В	110	228	87	425
С	360	800	300	1460
Д	450	392	90	932

5. Приведен фрагмент электронной таблицы. Какое значение будет в ячейке C2, если формулу из ячейки C1 скопировать в ячейку C2?

	А	В	С
1	10	30	=A1 * B1 - 30
2	20	40	

6. Приведен фрагмент электронной таблицы. Какое значение будет в ячейке D3, если формулу из ячейки C2 скопировать в ячейку D3?

	A	B	C	D
1	10	20	30	40
2	20	30	=A1 * C1 – B1	50
3	30	40		

7. Приведен фрагмент электронной таблицы. Формула из ячейки A3 скопирована в ячейку C2. Адреса ячеек в формуле были изменены автоматически при копировании. Каким будет числовое значение формулы в ячейке C2?

	A	B	C	D	E
	40	4	100	70	10
	30	3		60	11
	=B3*B \$2	2	300	50	12
	10	1	400	40	13

8. В ячейке A1 электронной таблицы записана формула =2*\$B\$4 – \$C1. Как будет выглядеть формула при копировании ее в ячейку B3?

- A) =4*\$B\$6 – \$C3
 B) =2*\$C\$4 – \$D1
 C) =2*\$B\$6 – \$D3
 D) =4*\$C\$5 – \$D3
 E) =2*\$B\$4 – \$C3

9. Формула, записанная в ячейку B11 электронной таблицы, была скопирована в ячейку A10. В результате значение в ячейке A10 вычисляется по формуле $x - 3y$, где x – значение ячейки C22, а y – значение ячейки D22. Какая из следующих формул может быть формулой в ячейке B11?

- A) =C22 – 3*D22
 B) =C\$22 – 3*D\$22
 C) =\$C22 – 3*\$D22
 D) =D\$22 – 3*\$D\$22
 E) =\$D22 – 3*\$D22

10. В ячейку A1 электронной таблицы записано число 10, в ячейку B1 – формула $=A1/2$, а в ячейку C1 – формула $=SUM(A1: B1) * 2$. Какое значение будет в ячейке C1?

11. В электронной таблице значение формулы $=SUM(A1: A4)$ равно 13, а значение формулы $=AVERAGE(A1: A5)$ равно 3. Каким будет значение формулы $=SUM(A1: A5)$?

12. Определите соответствие.

- | | |
|-------------------------|-------------------|
| 1) относительная ссылка | a) \$B5, C\$7 |
| 2) абсолютная ссылка | b) \$B\$5, C\$7 |
| 3) смешанная ссылка | c) \$B\$5, \$C\$7 |
| | d) B5\$, C7\$ |
| | e) B5, C7 |

13. Какой тип диаграммы уместно использовать, чтобы показать количество солнечных дней в каждом месяце года?

14. Приведен фрагмент электронной таблицы. Какое целое число следует записать в ячейку B1, чтобы после вычисления значений диапазона A2: D2 построенная диаграмма соответствовала рисунку?

	A	B	C	D
1	9		2	5
2	$=B2 + C2 + D2$	$=C2$	$=(A1 - D1) * (B1 - 5)$	$=(A1 - D1) * C1$





ПРОГРАММИРОВАНИЕ

- 14. Работа с числами
- 15. Строки
- 16. Списки
- 17. Функция
- 18. Решение задач



- Комментарии
- Отступ
- Арифметические операции
- Знаки сравнения

14. Работа с числами

- Что такое программа?
- Какие операторы языка программирования Python вам известны?

Деятельность

Ознакомьтесь с данным фрагментом программы на языке Python и определите, что она делает. Не выполняя ее на компьютере, определите значение переменной `i` в конце программы.

```
i = 1
while i <= 10:
    print (i*i)
    i = i + 1
```

Обсудим:

- Почему оператор `print` записан не на одном уровне с оператором `while`?
- Каким было бы конечное значение переменной `i`, если бы строка `i = i + 1` была написана на одном уровне с оператором `while`?

На языке программирования Python программный код является обычным текстовым файлом. Этот файл состоит из букв, цифр, пробелов, знаков препинания и других символов. Текст делится на строки. Каждый оператор Python может быть записан на нескольких строках. Для записи оператора на нескольких строках используют символ `"\"`.

Если через определенное время вернуться к тексту написанной ранее программы, вспомнить, что делает та или иная часть программы, будет достаточно трудно. Для того чтобы помнить детали программы и чтобы она была понятна другим, желательно в определенных ее местах размещать **комментарии**. Комментарии можно использовать для того, чтобы отметить, с какой целью была создана программа, кто является ее создателем, когда были внесены изменения в нее, указать назначение имеющихся в программе функций.

Комментарии на языке Python записываются после **символа "#"**. При переводе программы на машинный код комментарии не учитываются. Поэтому, если возникает необходимость "заморозить" какую-то часть программы, ее отмечают как комментарий.

Для правильного логического выполнения программы, написанной на языке Python, имеют большое значение ее структура и *отступы* в начале строк. Уровень отступа строк используется для объединения операторов в группы. Группа операторов может входить в состав другой группы.

Вы уже знакомы с числовыми величинами и арифметическими операциями над ними в языке Python. Еще раз напомним их.

Операция	Описание
$x + y$	Сложение (сумма чисел x и y)
$x - y$	Вычитание (разность чисел x и y)
$x * y$	Умножение (произведение чисел x и y)
x / y	Деление x на y (частное)
$x // y$	Целочисленное деление (результат – целая часть частного)
$x \% y$	Остаток, полученный от целочисленного деления x на y
$x ** y$	Возведение в степень (x в степени y)
$-x$	Изменение знака числа

При записи условия в программе используют *знаки сравнения*. В таблице показана запись этих знаков на языке ALPLogo и Python. Следует обратить особое внимание на запись знаков "равно" и "не равно", так как неправильное их использование может привести к программной ошибке.

ALPLogo	Python	Описание
=	==	Равно
<>	!=	Не равно
>	>	Больше
<	<	Меньше
>=	>=	Больше или равно
<=	<=	Меньше или равно

Разложение числа на цифры. Как вы знаете, при делении произвольного числа на 10 полученное частное определяет цифру этого числа в разряде единиц. Например, если цифра числа n в разряде единиц равна a , то

$$a = n \% 10$$

Понятно, что если отбросить последнюю цифру, то последняя цифра полученного нового числа будет цифрой разряда *десятков* первоначального числа. При помощи указанной операции, отделив последнюю цифру нового числа, получаем цифру десятка первоначального числа. Таким способом можно определить цифры любого числа. Для того чтобы отбросить последнюю цифру (последний разряд) числа, его следует разделить на 10 и взять целую часть, то есть использовать *целочисленное деление* ($//$). Например, при отбрасывании последней цифры числа n получается число m , то есть

$$m = n // 10$$

Следующая программа разбивает *двузначное число* (n), введенное с клавиатуры, на его цифры (a и b), находит сумму (S) и произведение (p) цифр. Найденные значения выводятся на экран. Если введенное число не является двузначным, на экран выводится сообщение об этом.

```
n = input(' Введите двузначное целое число: ')
n = int(n)

if n > 9 and n < 100:
    a = n % 10                # Цифра в разряде единиц
    b = n // 10              # Цифра в разряде десятков
    s = a + b
    p = a * b
    print(' Сумма цифр =', s)
    print(' Произведение цифр =', p)
else:
    print(' Число введено не верно!')
```

Теперь рассмотрим обобщенный случай этой задачи – для случая целого положительного числа, количество цифр которого заранее неизвестно.

Дано любое положительное целое число, количество цифр которого заранее неизвестно. Найдите сумму и произведение его цифр.

Решение: Обозначим данное число через n , сумму его цифр через S , а произведение его цифр через p . Тогда алгоритм решения задачи можно представить следующим образом:

1. Присвоить переменной S в качестве начального значения 0.
2. Присвоить переменной p в качестве начального значения 1.
3. До тех пор, пока n больше нуля,
 - найти остаток, полученный от деления n на 10 (последнюю цифру числа), прибавить его к сумме и увеличить произведение;
 - для удаления последней цифры (разряда) числа n произвести целочисленное деление его на 10.

1

```
n = input(' Введите число: ')
n = int(n)
s = 0
p = 1
while n > 0:
    r = n % 10
    s = s + r
    p = p * r
    n = n // 10
print(' Сумма цифр =', s)
print(' Произведение цифр =', p)
```

Заданы целое число n и все целые числа от 1, 2, ..., n , кроме одного. Написать программу, которая находит недостающее число. Например, если n равно 5, а введенные числа 2, 3, 1, 5, то недостающее число будет 4. Таким образом, результат должен быть 4.

Решение. Обозначим сумму заданных чисел через s_1 , а сумму всех чисел от 1 до n через s_2 . Тогда недостающее число можно определить как $s_2 - s_1$.

Пример. Для данного примера:

- Сумма чисел от 1 до 5 равна $s_2 = 1 + 2 + 3 + 4 + 5 = 15$;
- Сумма заданных чисел равна $s_1 = 2 + 3 + 1 + 5 = 11$.

Значит, недостающее число будет $s_2 - s_1 = 15 - 11 = 4$.

Сначала вводится значение n .

```
n = int( input(' Введите количество чисел: ' ) )
```

По известной из математики формуле вычисляется сумма s_2 чисел от 1 до n .

```
s2 = ( 1 + n ) * n / 2
```

2

Затем вводятся числа и вычисляется их сумма s_1 .

```
for i in range(1, n):
    x = int( input() )
    s1 += x
```

Выводится на экран разность сумм s_2 и s_1 .

```
print(s2 - s1)
```

В итоге, программа будет выглядеть так:

```
n = int( input(' Введите количество чисел: ' ) )
s2 = ( 1 + n ) * n / 2
s1 = 0
for i in range(1, n):
    x = int( input() )
    s1 += x
print(s2 - s1)
```

Составные условия. Нередко возникает необходимость проверки одновременно несколько условий. Для этого язык Python предлагает три логические операции: `and` (логическое И), `or` (логическое ИЛИ) и `not` (логическое НЕ). Если необходимо одновременно выполнить несколько условий, используется операция *AND*. Операция *OR* предусмотрена для указания того, что выполняется хотя бы одно из двух или более условий. Операция *NOT* обозначает противоположное условие (т.е. обратное первоначальному условию). Другими словами, если данное условие истинно, то обратное ему ложно, и наоборот.

Изучим сами

Из Баку в Тебриз самолет летает только по понедельникам и четвергам. Предположим, переменная `day` предназначена для хранения номера дня недели (1 – понедельник, 7 – воскресенье). Следующий фрагмент программы определяет, полетит ли самолет в заданный день.

```
if day == 1 or day == 4:
    print ("Полетит!")
else:
    print ("Полёта нет.")
```

Напишите этот фрагмент, используя операции `and` и `not`.

Проверьте себя

1. Как задаются комментарии на языке программирования Python?
2. Дано двузначное целое число. Вывести первую слева цифру (десятки) числа, а затем через два пробела цифру справа (единицы).
3. Выведите третью справа цифру заданного натурального числа.
4. Составьте программу, определяющую делимость заданного числа на 3.
5. Запишите данный алгоритм в виде программы на языке Python.
 - 1) Введите произвольное двузначное число.
 - 2) Умножьте его само на себя.
 - 3) Сложите полученное число с произведением введенного числа и 4.
 - 4) Прибавьте к полученному результату число 3.
 - 5) Разделите полученный результат на число, которое на 3 больше введенного числа.
 - 6) Вычтите выбранное вами число из полученного частного.
 - 7) Выведите результат на экран.

ключевые слова

- Строка
- Пустая строка
- Срез
- Индекс
- Длина строки
- Метод

15. Строки

- Что такое строковый тип данных в программировании?
- Как строку, состоящую из цифр, отличить от числа на языке Python?

Деятельность

Наберите и выполните программу в среде Python.

```
s1 = 'a'
s2 = 'l'
s3 = 'p'
print (s1 + s2 + s3)
print ("a", "l", "p")
```

Обсудим:

- Какого типа величины использованы в программе?
- Одинаков ли результат у двух операторов `print`? Если нет, то что надо сделать, чтобы был одинаковый?

На языке Python одним из часто используемых типов данных является строка.

Строка – это последовательность произвольных символов. Строка обычно заключается в одиночные кавычки ('), но возможно обособление строки обычными кавычками ("). Например, ' Это пример. ', "Борются не силой, а знанием. ". Строка может состоять из букв, цифр и других символов, в том числе из пробела. Строки можно присваивать переменным. Например:

```
>>> a = 'Азербайджанская '
>>> b = 'Республика'
```

Как и над числами, над строками можно производить определенные действия. Например, при сложении двух строк образуется строка, полученная "склеиванием" двух строк.

```
>>> c = a + b
>>> print (c)
Азербайджанская Республика
```

```
>>> c = a + 'Демократическая ' + b
>>> print (c)
Азербайджанская Демократическая Республика
```

Каждый символ в строке в зависимости от местоположения имеет свой порядковый номер, по которому можно найти символ в строке или вырезать его из строки. На языке Python нумерация символов строки начинается с 0. Второй символ будет иметь номер 1, третий 2 и т.д. В программировании номер именуется *индексом*. По индексу можно определить конкретный символ строки. Например:

```
>>> s = 'Строки'
>>> s[2]
'р'
>>> s[-3]
'о'
```

Используя два индекса, можно выделить фрагмент строки, называемый *срезом* строки.

```
>>> s[0:5]
'Строк'
```

Как видите, несмотря на то, что вторым индексом диапазона указано число 5, стоящий на этой позиции символ не входит в срез строки. Другими словами, символ, стоящий на позиции второго индекса, никогда не входит в срез строки. Задавая диапазон, один из его индексов можно не указывать. Если в срезе опущен первый индекс, то Python автоматически вырезает строку с нулевого символа. Если опущен второй индекс, то в срез попадают все символы до конца строки.

```
>>> s = 'Строки'
>>> s[:5]
'Строк'
>>> s[5:]
'и'
```

Под *длиной строки* подразумевается количество всех символов строки, включая пробелы. С помощью специальной функции `len()` можно определить длину строки.

```
>>> s = 'Бакинский Государственный Университет'
>>> len(s)
37
```

Как вы знаете, во время работы программы все используемые в ней данные хранятся в оперативной памяти компьютера. Программа воспринимает данные, расположенные на других устройствах, как *внешние*. Процесс получения внешних данных называется *вводом*, а процесс переноса данных на внешние устройства – *выводом*. Как и для числовых данных, для ввода строковых величин с клавиатуры используется функция `input()`. Эта функция воспринимает вводимую величину как строку. Например, при вводе с клавиатуры с помощью команды `a = input()` числа 123 переменная `a` получает в качестве значения не 123, а '123', то есть `a = '123'`. Поэтому иногда приходится менять тип величин с одного на другой. Как и в других языках программирования, для

перевода величин с одного типа на другой на языке Python имеются функции. Например, для перевода числовой величины в строковую используют функцию `str()`, а для перевода строковой величины в числовую – функцию `int()`.

```
>>> int('123')          >>> str(123)
123                      '123'
```

В Python для печати данных или, другими словами, для их вывода используется функция `print()`. С помощью этой функции можно вывести на экран как тексты, так и значения переменных.

```
>>> a = 14
>>> print('Орхану', a, 'лет.')
Орхану 14 лет.
```

В этом примере функция `print()` все значения выдает на печать в одной строке, и для отделения их друг от друга ставит между ними пробел. Для обособления значений имеются и другие способы. Например, между выводимыми значениями можно поставить в качестве символов запятую, дефис. Символ, используемый в качестве разделителя, указывается с помощью параметра `sep` (например, `sep = '-'`). Если задано `sep = '\n'`, то каждое значение будет напечатано с новой строки.

```
>>> a = 14
>>> print('Орхану', a, 'лет.', sep = '\n')
Орхану
14
лет.
```

Как вам известно, любой объект имеет определенные характеристики. В информатике различают два вида характеристик объекта: *свойства объекта* и *методы объекта*. Чтобы показать на языке программирования метод произвольного объекта, используют специальную запись: сначала записывают имя объекта, потом – метод. Между ними ставится точка (например, `s1.upper()`, `S.find('a', 5, 10)`). На языке Python строки рассматриваются как объекты и, как и у каждого объекта, у них имеются свои методы. Чаще всего используется метод `find()`. Этот метод служит для нахождения подстроки в заданной строке. Если подстрока будет найдена, метод возвращает индекс первого символа первого вхождения искомой подстроки. Если подстрока не найдена, метод возвращает значение `-1`. Например:

```
>>> S = 'Не стыдно не знать, стыдно не учиться.'
>>> print(S.find('учиться'))
30
>>> print(S.find('ст'))
3
>>> print(S.find('книга'))
-1
```

У метода `find()` могут быть и другие параметры. Если вызвать метод `find()` с тремя параметрами `S.find(T, a, b)`, то поиск будет осуществляться в подстроке `S[a:b]`. Метод `S.find(T, a, b)` возвращает индекс первого символа первого вхождения подстроки `T` в строку `S`.

```
>>> print(S.find('a', 5, 10))
```

8

Составьте программу, в которой с клавиатуры вводятся ваша фамилия и имя, а выводятся в обратном порядке – сначала имя, а затем фамилия.

Решение: Пусть `s` обозначает входную строку, `lastname` и `firstname` – первое и второе слова в строке, а `k` – позицию пробела между словами в строке. Алгоритм решения задачи будет такой:

- 1) Вводится строка.
- 2) С помощью метода `find()` определяется позиция пробела между словами.
- 3) Часть строки от начала до пробела присваивается переменной `lastname`.
- 4) Часть строки с позиции после пробела и до конца присваивается переменной `firstname`.
- 5) Образуется новая строка, состоящая из второго слова (`firstname`), пробела и первого слова (`lastname`).
- 6) Новая строка выводится на печать.

1

```
s = input()
p = s.find(' ')
lastname = s[:p]           # первое слово – фамилия
firstname = s[p + 1:]     # второе слово – имя
s = firstname + ' ' + lastname
print(s)
```

У объекта "строка" имеются и другие методы, которые приведены в следующей таблице.

Метод	Описание
<code>s1.count(s0)</code>	<p>Определяет количество вхождений подстроки <code>s0</code> в строку <code>s1</code>:</p> <pre>>>> s1 = 'abrakadabra' >>> s1.count('ab')</pre> <p>2</p>
<code>s1.strip()</code>	<p>Удаляет пробелы в начале и в конце строки:</p> <pre>>>> s1 = ' abc ' >>> s1.strip() 'abc'</pre>

<code>s1.replace(s2, s3)</code>	Заменяет фрагмент строки <code>s2</code> на фрагмент <code>s3</code> : <pre>>>> s1 = 'Кыргызстан' >>> s1.replace('Кыргыз', 'Казах') 'Казахстан'</pre>
<code>s1.capitalize()</code>	Преобразует первую букву в заглавную: <pre>>>> s1 = 'будьте внимательны!' >>> s1.capitalize() 'Будьте внимательны!'</pre>
<code>s1.upper()</code>	Преобразует все буквы строки в заглавные. Например: <pre>>>> s1 = 'Гусар Губа Хачмаз' >>> s1.upper() 'ГУСАР ГУБА ХАЧМАЗ'</pre>
<code>s1.lower()</code>	Преобразует все буквы строки в строчные. Например: <pre>>>> s1 = 'Голубой-Пурпурный-Желтый-Черный' >>> s1.lower() 'голубой-пурпурный-желтый-черный'</pre>

Изучим сами

```
s = input()
print(s[2])
...
```

Этот фрагмент программы соответствует двум первым шагам алгоритма. Допишите операторы, соответствующие заданному алгоритму.

Алгоритм

1. Ввести с клавиатуры произвольную строку с не менее 3 символами.
2. Вывести третий символ введенной строки.
3. Вывести последний символ строки.
4. Вывести первые пять символов строки.
5. Вывести все символы строки, кроме двух последних.
6. Вывести символы строки с четными индексами.
7. Вывести символы с нечетными индексами.
8. Вывести длину строки.

Проверьте себя

1. Что такое строка?
2. Что означают в программе операции ввода и вывода? Какими функциями на Python они представлены?
3. Что такое метод?
4. Составьте программу, выводящую буквы слова в обратном порядке.
5. Выведите на экран символ, расположенный в середине строки. Если количество символов четно, выведите два символа.

- Список
- Элементы списка
- Индекс
- Срез
- Цикл

16. Списки

- Как создаются маркированные и нумерованные списки в текстовом редакторе?
- К какому виду списка относятся указанные здесь вопросы?

Деятельность

```
s = 0
for i in range(1, 101):
    s = s + i
print(s)
```

Обсудим:

- Что является результатом выполнения данного фрагмента программы?
- Какая алгоритмическая структура использована в этом фрагменте?
- Что делает функция range()?

Для совместного хранения многочисленных данных в языке Python широко используют списки. **Список** – это набор объектов, хранимых в определенном порядке. Объекты, составляющие список, называют его **элементами**. В списках можно хранить числа, строки, другие списки или все вместе. Каждому элементу списка соответствует порядковый номер (**индекс**), по которому к нему можно обращаться. В любое время можно изменить элементы списка, добавив к списку новые элементы или какие-то элементы удалив из списка. На языке Python элементы списка записывают в квадратных скобках и отделяются друг от друга запятой. В качестве примера ниже приведен список дней недели, сохраненный в переменной `lst`.

```
>>> lst = ['Понедельник', 'Вторник',
           'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье']
```

На языке Python элементы списка нумеруются начиная с нуля. В примере, приведенном выше, номер (индекс) элемента 'Понедельник' равен 0. Для того чтобы обратиться к индексу `i` списка `lst`, нужно просто указать `lst[i]`.

```
>>> lst[2]
'Среда'
```

В списке можно с легкостью изменить любой элемент. Для этого элементу с соответствующим индексом нужно просто присвоить новое значение. Присваивание `lst[i] = x` изменяет значения элемента `lst` с индексом `i` на `x`. Например, после выполнения команды `lst[3] = '4-й день'` в списке дней недели изменится 4-й элемент (с индексом 3).

```
>>> lst[3] = '4-й день'
>>> lst
['Понедельник', 'Вторник',
'Среда', '4-й день', 'Пятница', 'Суббота', 'Воскресенье']
```

Иногда приходится использовать не весь список, а какой-то его *срез* (подсписок). Например, если из списка `lst` нам необходимы только элементы с индексами от `i` до `j`, это можно показать как `lst[i:j]`. (учтите, что элемент с индексом `i` входит в последний список, а элемент с индексом `j` – нет).

```
>>> lst[1:3]
['Вторник', 'Среда']
```

В программировании, особенно при работе со списками (во многих языках это массивы), использовать *циклы* очень удобно. Например, при выполнении фрагмента программы

```
lst = ['Понедельник', 'Вторник',
'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье']
for s in lst:
    print(s)
```

дни недели будут выведены в следующем виде:

```
Понедельник
Вторник
Среда
Четверг
Пятница
Суббота
Воскресенье
```

Для списков, составленных только из чисел, существуют специальные функции. Например, функция `sum()` вычисляет сумму элементов списка:

```
>>> mas = [1, 2, 3, 4]
>>> sum(mas)
10
```

В таблице представлены основные функции (операции), связанные со списками, и их описание.

Функция или операция	Описание
<code>len(lst)</code>	<p>Определяет число элементов списка:</p> <pre>>>> lst = [1, 2, 3] >>> len(lst) 3</pre>
<code>lst1 + lst2</code>	<p>Элементы списка <code>lst2</code> добавляются в конец списка <code>lst1</code>:</p> <pre>>>> lst1 = [1, 2, 3] >>> lst2 = ['один', 'два', 'три'] >>> lst1 + lst2 [1, 2, 3, 'один', 'два', 'три']</pre>
<code>min(lst)</code>	<p>Определяет минимальный элемент списка:</p> <pre>>>> lst1 = [1, 2, 3, 85, -42, 33, 84] >>> min(lst1) -42</pre> <p>В величинах строкового типа за основу берется порядок в алфавите.</p> <pre>>>> lst2 = ['семь', 'восемь', 'девять'] >>> min(lst2) 'восемь'</pre>
<code>max(lst)</code>	<p>Определяется максимальный элемент списка:</p> <pre>>>> lst = [1, 2, 3, 85, -42, 33, 84] >>> max(lst) 85</pre>
<code>del lst[i]</code>	<p>Из списка удаляется элемент с номером <code>i</code>:</p> <pre>>>> lst = [1, 2, 3, 'один', 'два', 'три'] >>> del lst[2] >>> lst [1, 2, 'один', 'два', 'три']</pre>

Как и строка, список в Python также является объектом и имеет свои методы. Ниже в таблице указаны некоторые методы, которые часто используются при работе со списками.

Метод	Описание
<code>lst.append(x)</code>	В конец списка добавляется элемент <code>x</code> : <pre>>>> lst = [1, 2, 3] >>> lst.append(4) >>> lst [1, 2, 3, 4]</pre>
<code>lst.count(x)</code>	Определяется число элементов, равных <code>x</code> : <pre>>>> lst = ['один', 'два', 'два', 'три'] >>> lst.count('два') 2</pre>
<code>lst.index(x)</code>	Определяет место (индекс) первого вхождения элемента <code>x</code> в список: <pre>>>> lst = ['один', 'два', 'два', 'три'] >>> lst.index('два') 1</pre>
<code>lst.remove(x)</code>	Удаляет первый элемент с указанным значением <code>x</code> : <pre>>>> lst = ['один', 'два', 'два', 'три'] >>> lst.remove('два') >>> lst ['один', 'два', 'три']</pre>
<code>lst.insert(i, x)</code>	Вставляет в списке на <code>i</code> позицию элемент <code>x</code> или список: <pre>>>> lst = ['один', 'два', 'два', 'три'] >>> lst.insert(2, 'один') >>> lst ['один', 'два', 'один', 'два', 'три']</pre>
<code>lst.sort()</code>	Сортирует список по возрастанию (в алфавитном порядке): <pre>>>> lst = ['один', 'два', 'один', 'два', 'три'] >>> lst.sort() >>> lst ['один', 'один', 'два', 'два', 'три']</pre>
<code>lst.reverse()</code>	Размещает элементы списка в обратном порядке: <pre>>>> lst = [1, 2, 3, 'один', 'два', 'три'] >>> lst.reverse() >>> lst ['три', 'два', 'один', 3, 2, 1]</pre>

Как видите, списки в языке Python – очень удобная структура данных и, используя их, можно решать различные задачи.

Для преобразования строки в список в языке Python используют функцию `list()`:

```
>>> s = 'степь'
>>> lst = list(s)
>>> lst
['с', 'т', 'е', 'п', 'ь']
```

Познакомимся еще с двумя методами, которые часто используют при решении задач. Один из них – метод `join()`. Этот метод вставляет между элементами списка указанную строку и создает одну строку (то есть при помощи данной строки объединяет элементы списка). Например:

```
>>> lst = ['1', '2', '3']
>>> s = '---'.join(lst)
>>> s
'1---2---3'
```

В противоположность этому метод `split()` делит строку при помощи данного разделительного символа и составляет из частей список. Например:

```
>>> s = '12 34 56'
>>> lst = s.split(' ')      # пробел - разделительный символ
>>> lst
['12', '34', '56']
```

По правилам правописания в тексте после запятой всегда ставится пробел. Следующая программа находит в данном тексте ошибки такого типа и исправляет их. Проверьте работу программы.

Решение: Обозначим вводимый в программу текст (строку) через `s`, список, соответствующий тексту, через `lst`, а индекс элемента списка через `i`. Алгоритм решения задачи будет такой:

1

- 1) Вводится текст (строка).
- 2) С помощью функции `list()` строка преобразуется в список.
- 3) Проверяются с начала списка по одному все его элементы. Если элемент, который следует после элемента с символом ' ', не является пробелом, после этого элемента списка добавляется новый элемент – пробел.
- 4) Новый список преобразуется в строку (текст).
- 5) Измененный текст выводится.

```
s = input('Введите текст: ')
lst = list(s)

i = 0
while i < len(lst):
    if (lst[i] == ',' and lst[i+1] != ' '):
        lst.insert(i+1, ' ')
        i = i + 1

s = ''.join(lst)
print(s)
```

Изучим сами

Используя функции `list()` и `count(x)`, напишите программу, подсчитывающую количество букв 'a' в заданном тексте.

Объекты, составляющие список, называют его элементами. В списках можно хранить числа, строки, другие списки или все вместе. Каждому элементу списка соответствует порядковый номер, по которому к нему можно обращаться.

Проверьте себя

1. Что такое список и какие величины могут быть его элементами?
2. Какие действия могут быть выполнены над списками?
3. Создайте список из чисел от 1 до 100 при помощи функций `range()` и `list()`.
4. Напишите программу, которая находит не проставленные после знаков препинания пробелы и исправляет ошибки.
5. Напишите программу, которая находит в данной строке (тексте) самое короткое и самое длинное слово.

Ключевые слова

- Подпрограмма
- Процедура
- Функция
- Вызов функции
- Формальный параметр
- Фактический параметр

17. Функция

- Что такое подпрограммы и какие преимущества они дают программисту?
- Какую функцию выполняет команда `input()`?

Деятельность

Просмотрите следующий программный код и определите, для решения какой задачи он предназначен.

```
n = abs(int(input()))
a = (n // 100) % 10
b = (n // 10) % 10
c = n % 10
print(a, b, c)
```

Обсудим:

- Как описать словами последовательность команд в первой строке?

Программисты могут понять программный код, состоящий из нескольких сотен строк. Но чем больше строк в программе, тем сложнее понять ее. Даже если и понятна функция каждого оператора, в таких программах трудно определить, для каких целей использован в данном месте тот или иной оператор. Вносить изменения или исправлять ошибки в такой программе также нелегко.

Для решения этой проблемы программа делится на отдельные **подпрограммы**, выполняющие простые операции. Таким образом, конечная программа состоит не из отдельных операторов, а из отдельных блоков кода, каждый из которых имеет свое имя. Использование подпрограмм сокращает время написания программы, уменьшает объем программного кода и делает структуру программы более понятной. В то же время такой подход сокращает количество ошибок при написании программы.

Подпрограммы обычно делят на две категории: *процедуры* и *функции*. **Процедура** служит лишь для выполнения последовательности операторов. Подпрограммы, известные вам из среды программирования ALPLogo, являются процедурами. **Функция** вычисляет определенное значение, и возвращает его программе (подпрограмме), вызывающей ее. В некоторых языках программирования, в том числе в языке Python, подпрограммы не делят на процедуры и функции, их все рассматривают как функции. В таких языках процедуры – это функции,

которые не возвращают никаких значений. Как было отмечено, каждая функция имеет свое *имя*. Для выполнения любой функции должно быть обращение к ее имени, то есть функция должна быть **вызвана**. Функцию можно вызывать из любого места программы.

В языке Python существует множество функций различного назначения. Например, в программе, размещенной в блоке "Деятельность", использовано четыре функции: `abs()`, `int()`, `input()` и `print()`. Это стандартные функции, которые имеются на Python. Несмотря на это, при решении задач возникает необходимость в создании новых, собственных функций. Естественно, на языке Python существует возможность создавать новые функции.

На языке Python описание функции начинается ключевым словом `def`, после которого следует имя функции. Круглые скобки, которые указываются после имени функции, отличают функцию от обычной переменной. Двоеточие указывает начало команд, входящих в функцию. Например, опишем простую функцию с названием `greeting()`:

```
def greeting():
    print(' Добро пожаловать! ')
```

Для того чтобы вызвать функцию из любого места программы, как и для стандартных функций необходимо указать ее имя как отдельный оператор.

```
greeting()
```

Результат работы этой функции всегда одинаковый, то есть, если сказать на языке математики, – у этой функций нет аргументов. В программировании, как правило, такие функции не используются. Обычно функция получает данные от вызывающей ее программы. Данные передаются функции в виде параметров. При создании функции значения ее параметров пока не известны. При описании функции после ее имени указываются так называемые формальные параметры. **Формальные параметры** – произвольные переменные, которые определяют передаваемые данные. Они необходимы только для описания действий, которые выполняет функция.

При вызове функции после ее имени указываются так называемые **фактические параметры**, значения, которые передаются в функцию. Формальные параметры заменяются фактическими значениями при выполнении операторов функции.

```
def salam(s):
    print(s)

salam(' Доброе утро! ')
salam(' Добрый вечер! ')
```

Формальный параметр

Фактический параметр

При выполнении вышеуказанного образца программы на экран будут выведены следующие строки:

Доброе утро!
Добрый вечер!

1

Напишите функцию, которая возвращает наибольшее из двух целых чисел.

Решение: Алгоритм определения наибольшего из двух чисел вам известен. Просто нужно представить этот алгоритм в виде функции:

```
def max2(a, b) :  
    if a > b :  
        maximum = a  
    else :  
        maximum = b  
    return maximum
```

2

Напишите программу, которая преобразует год мусульманского календаря Хиджры в год по Григорианскому (европейскому) календарю и наоборот, а также выводит результаты на экран. При этом вычисления производятся по формулам $M = H - H / 33 + 622$ и $H = M - 622 + (M - 622) / 32$. Здесь переменная M указывает год по Григорианскому календарю, H – год Хиджры.

Решение: Опишем две функции: одна из них – функция `mi_ladi_hicri()` на основе формулы будет переводить переданный как параметр григорианский год в год Хиджры, вторая функция – `hicri_mi_ladi()` будет переводить год Хиджры в григорианский год и выдавать результат на экран.

```
def mi_ladi_hicri(s):  
    M = int(s)  
    H = M - 622 + (M - 622) // 32  
    print(' Год Хиджры: ', H)  
  
def hicri_mi_ladi(s):  
    H = int(s)  
    M = H - (H // 33) + 622  
    print(' Григорианский год: ', M)  
  
s = input('Mi_ladi ili: ')  
mi_ladi_hicri(s)  
  
s = input('Hicri ili: ')  
hicri_mi_ladi(s)
```

В этом примере каждая функция на основе полученных извне значений проводит вычисления и выводит результат на экран. Но в большинстве случаев при решении практических задач удобнее возвращать полученное значение функции вызывающей ее программе. Для этой цели в программе Python предусмотрена команда `return`. Эта команда записывается в конце функции и после нее указывается значение, которое должна вернуть функция.

Как вы знаете, функция `input()` независимо от типа величины, вводимой с клавиатуры, всегда возвращает строку. А данная ниже новая функция `num_input()` возвращает число.

```
def num_input(prompt):
    typed = input(prompt)    # Вводимая строка
                            # сохраняется в переменной typed
    num = int(typed)        # Строка преобразуется в число и
                            # сохраняется в переменной num
    return num              # Возвращается значение переменной

a = num_input('Введите a: ')
b = num_input('Введите b: ')
print('a * b =', a * b)
```

Любую программу, приведенную в предыдущих темах, можно представить в виде функции. Например, программу, вычисляющую сумму цифр заданного положительного целого числа, можно написать с использованием функции так:

```
def sumDigits(n) :
    s = 0
    while n != 0 :
        q = n % 10
        s += q
        n = n // 10
    return s

n = int(input('Введите число: '))

print('Сумму цифр =', sumDigits(n))
```

Изучим сами

Программу, связанную с календарем, можно изменить так, чтобы значения нового календаря выводились на экран не в составе функций, а в вызывающей их программе.

```
def miladi_hicri (S):  
    M = int(S)  
    H = M - 622 + (M - 622) // 32  
    return H  
  
def hicri_miladi (S):  
    H = int(S)  
    M = H - (H // 33) + 622  
    return M  
  
S = input(' Григорианский год: ')  
H = miladi_hicri (S)  
print(' Год Хиджры: ', H)  
  
S = input(' Год Хиджры: ')  
M = hicri_miladi (S)  
print(' Григорианский год: ', M)
```

Проверьте себя

1. Какие преимущества дает использование подпрограмм?
2. Какие типы подпрограмм существуют?
3. Чем функция отличается от процедуры?
4. Какая функция в языке Python соответствует устройствам вывода компьютера?
5. Используя программу календаря, определите год своего рождения по мусульманскому летоисчислению.
6. Напишите функцию, вычисляющую периметр треугольника по данным значениям его сторон.

18. Решение задач

Наибольший общий делитель (НОД). Из математики известно, что *наибольший общий делитель* двух чисел – это наибольшее число, на которое эти числа делятся без остатка. В курсе математики приведен такой алгоритм нахождения НОД: заданные числа разложить на простые множители, а затем выбрать те из них, которые принадлежат обоим разложениям. Обычно на практике этот алгоритм не применяется, поскольку разложение числа на простые множители – достаточно сложная задача, к тому же необходимо найти среди множителей одинаковые. Поэтому для нахождения НОД часто используется метод, называемый *алгоритмом Евклида*. Это один из известных и популярных алгоритмов.

Нахождение НОД методом вычитания

Напишите программу, которая находит наибольший общий делитель двух заданных целых чисел.

Решение. Для нахождения НОД можно использовать алгоритм Евклида:

- 1) Из большего числа вычтешь меньшее.
- 2) Если результат равен 0, то значит числа равны друг другу и являются НОД (следует выйти из цикла).
- 3) Если результат вычитания не равен 0, то большее число заменить на результат вычитания.
- 4) Перейти к шагу 1.

1

Пример: Нахождение НОД для 30 и 18.

$$30 - 18 = 12$$

$$18 - 12 = 6$$

$$12 - 6 = 6$$

$$6 - 6 = 0.$$

Значит НОД (30, 18) = 6.

```
a = int(input(' Введите первое число: '))
b = int(input(' Введите второе число: '))

while a != b:
    if a > b:
        a = a - b
    else:
        b = b - a
print ('НОД =', a)
```

Этот алгоритм для нахождения НОД двух чисел не самый оптимальный. Например, для $a = 1000\ 000$ и $b = 2$ этот алгоритм будет выполняться 500 000 раз. Следующий пример показывает, как можно улучшить этот алгоритм для больших значений a или b .

Нахождение НОД методом деления

Этот метод очень похож на метод вычитания. Просто на каждом шаге вместо вычитания применяется деление. В результате получить НОД можно за меньшее количество шагов:

- 1) Разделить большее число на меньшее.
- 2) Если остаток равен 0, то меньшее число и есть НОД (следует выйти из цикла).
- 3) Если остаток не равен 0, то большее число заменить на остаток от деления.
- 4) Перейти к шагу 1.

2

Пример: Нахождение НОД для 30 и 18.

$$30 / 18 = 1 \text{ (остаток 12)}$$

$$18 / 12 = 1 \text{ (остаток 6)}$$

$$12 / 6 = 2 \text{ (остаток 0)}$$

Итак, $\text{НОД}(30, 18) = 6$.

```
a = int(input(' Введите первое число: '))
b = int(input(' Введите второе число: '))
while a != 0 and b != 0:
    if a > b:
        a = a % b
    else:
        b = b % a
print ('НОД =', a + b)
```

Программа на языке Python с использованием функции

3

```
def nod(a, b):
    while a != 0 and b != 0:
        if a > b:
            a = a % b
        else:
            b = b % a
    return a + b

a = int(input(' Введите первое число: '))
b = int(input(' Введите второе число: '))
print (nod(a, b))
```

Примечание. В цикле остаток, полученный при делении, присваивается переменным a или b . Если остаток равен нулю (мы не знаем, он в a или в b , поэтому проверяем оба условия), то цикл завершается. В конце, так как не известно, какой переменной присвоен НОД, на выход выдается сумма a и b (одна из переменных всегда будет равна 0 и не окажет влияния на сумму).

Факториал числа. *Факториалом* натурального числа называют произведение натуральных чисел от 1 до этого числа включительно. Например, факториал числа 5 равен $1 \times 2 \times 3 \times 4 \times 5 = 120$. Формулу вычисления факториала можно записать так:

$$n! = 1 \times 2 \times \dots \times n;$$

где n – данное число, а $n!$ – факториал этого числа.

Вычисление факториала

Напишите программу, которая вычисляет факториал заданного числа, используя формулу расчета факториала и цикл `while`.

```
n = input("Число, факториал которого нужно найти: ")
n = int(n)
f = 1
i = 1
while i <= n:
    f = f * i
    i = i + 1
print (n, "! = ", f)
```

4

Допустим $n = 5$, то есть надо вычислить $5!$. При первом проходе в теле цикла `while` переменной f присваивается 1×1 . После второго прохода – 1×2 , затем – 2×3 , 6×4 и 24×5 . В шестой раз цикл `while` не будет выполняться, так как значение переменной i станет равным 6 и условие $i \leq n$ не будет выполнено.

Заметим, что по договоренности принято: $0! = 1$ и $1! = 1$.

Перевод числа из десятичной системы счисления в двоичную. Запись заданного числа A в двоичной системе счисления означает, что оно представлено в виде следующей суммы:

$$A = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

Итак, чтобы получить двоичное представление числа A , необходимо найти коэффициенты a_0, a_1, \dots, a_n . Для этого сначала разделим число A на 2. Понятно, что остаток будет равен a_0 . Потому что в показанной выше развернутой записи числа A все суммы, кроме последней, целиком делятся на 2. Затем разделим частное, полученное при делении числа A на 2, еще раз на 2. Новый остаток, полученный в этот раз, будет равен a_1 . Если продолжить этот процесс, то мы найдем все цифры a_0, a_1, \dots, a_n в двоичном представлении числа A .

Перевод числа из 10-й системы счисления в 2-ную

Напишите программу, переводящую заданное натуральное число из десятичной системы счисления в двоичную систему счисления.

Решение. Процесс получения двоичного числа из десятичного числа можно описать в виде алгоритма:

- 1) Данное десятичное натуральное число делится на 2 (на основании двоичной системы счисления).
- 2) Частное в виде целого числа присваивается одной переменной, а остаток в виде строки – другой (если нет остатка, записывается 0).
- 3) Если частное не равно нулю, то оно делится на 2 и результат присваивается переменной, в которой было записано частное (предыдущее значение переменной стирается). А в начало строковой переменной, предусмотренной для хранения остатков, добавляется новый остаток.
- 4) Шаг 3 повторяется до тех пор, пока остаток не будет равен нулю.
- 5) Размещенные в строковой переменной остатки и будут двоичной записью десятичного числа. Заметим, что остатки (0 или 1) располагаются в строковой переменной справа налево в порядке их вычисления.

5

```
x = int(input("Введите натуральное число: "))
n = ""
while x > 0:
    y = str(x % 2)
    n = y + n
    x = int(x / 2)
print (n)
```


Числа Фибоначчи. Элементы последовательности, начинающейся с 1, 1, в которой каждое последующее число, начиная с третьего, равно сумме двух предыдущих чисел, называют **числами Фибоначчи**: 1, 1, 2, 3, 5, 8, 13 и т.д.

Эту последовательность чисел можно представить в виде формулы так:

$$F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2}$$

Пример: $F_3 = F_2 + F_1 = 1 + 1 = 2$

$$F_5 = F_4 + F_3 = 3 + 2 = 5$$

$$F_4 = F_3 + F_2 = 2 + 1 = 3$$

$$F_6 = F_5 + F_4 = 5 + 3 = 8$$

Нахождение n-го числа в последовательности Фибоначчи

Напишите программу, которая находит n -е число последовательности Фибоначчи для заданного числа n .

Решение. Алгоритм вычисления n -го числа в последовательности Фибоначчи:

- 1) Вводится номер определяемого элемента n .
- 2) Проверяется: если $n < 3$, то в качестве результата выводится 1.
- 3) Если $n \geq 3$, то
- 4) Первым двум членам последовательности $fi\ b1$ и $fi\ b2$ присваиваются начальные значения (1 и 1).
- 5) Затем, начиная с номера 3 до n (включительно):
 - Вычисляется сумма $fi\ b1$ и $fi\ b2$ и результат присваивается третьей переменной $fi\ b_sum$.
 - Производятся замены $fi\ b1 = fi\ b2$ и $fi\ b2 = fi\ b_sum$.
- 6) Выводится результат.

6

```

fi b1 = 1
fi b2 = 1
n = int(input("Номер числа в послед-ти Фибоначчи: "))
if n < 3:
    print(1)
else:
    fi b1 = 1
    fi b2 = 1
    i = 3
    while i <= n:
        fi b_sum = fi b2 + fi b1
        fi b1 = fi b2
        fi b2 = fi b_sum
        i = i + 1
    print (fi b_sum)

```

Странный алгоритм. Рассмотрим алгоритм, который принимает на вход целое положительное число n . Если число четное, то по алгоритму оно делится на 2, если нечетное, умножается на 3 и прибавляется 1. Процесс повторяется до тех пор, пока значение n не станет равным 1. Например, для $n = 3$ получается следующая последовательность:

$$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1.$$

Необходимо написать программу, моделирующую работу этого алгоритма для любого заданного числа n .

Эта задача связана со знаменитой *гипотезой Коллатца*: согласно этой гипотезе описанный выше алгоритм завершается результатом, равным 1 при любом n . Это остается недоказанным и по сей день. Это простая задача моделирования, не требующая особых размышлений. Программу можно написать на Python следующим образом:

7

```
n = int(input("Введите натуральное число: "))
while n > 1:
    print (int(n), end = " ")
    if n % 2 == 0:
        n /= 2
    else:
        n = n*3 + 1
print (int(n))
```

Программа сначала считывает число n из входного потока, затем моделирует алгоритм и после каждого шага передает значение n на выход. Легко проверить, например, для случая $n = 3$, что эта программа правильно выполняет алгоритм, заданный в условии задачи.

Проверьте себя

1. Напишите программу, находящую наибольший общий делитель трех чисел.
2. Используя формулу $\text{НОК} = ab / \text{НОД}(a, b)$, напишите программу вычисления наименьшего общего кратного (НОК) двух натуральных чисел a и b .
3. Напишите программу вычисления по формуле для заданных натуральных чисел n и k ($k < n$).

$$C_n^k = \frac{n!}{k!(n-k)!}$$

4. Напишите программу вычисления произведения $1 \times 3 \times \dots \times (2n-1)$ для заданного натурального числа n .
5. Напишите программу нахождения числа единиц в двоичной записи десятичного числа 123456.
6. Напишите программу, определяющую, чего больше в двоичной записи десятичного числа 135797: единиц или нулей.
7. Напишите программу нахождения числа единиц в двоичной записи натурального числа n в интервале $[1, 100]$.
8. Напишите программу, выводящую на экран первые n чисел последовательности Фибоначчи.
9. Напишите программу, выводящую на экран n_1 -й и n_2 -й элемент последовательности Фибоначчи, а также их сумму.
10. Напишите программу, определяющую, является ли вводимое с клавиатуры число числом Фибоначчи.
11. Дано четырехзначное натуральное число. Определите, является ли это число палиндромом. Натуральное число, которое читается одинаково слева направо и справа налево, называется числом-палиндромом. Например, 232, 2332, 2341432 — числа-палиндромы.
12. Необходимо определить, является ли данное трехзначное число числом Армстронга. Если сумма кубов цифр трехзначного числа равна самому этому числу, такое число называется числом Армстронга. Например, 153 — это число Армстронга: $153 = 1^3 + 5^3 + 3^3$.
13. Дано четырехзначное натуральное число. Определите, делится ли это число на все свои цифры. Например, число 1326 делится на все свои цифры (1, 3, 2, 6).
14. Дано трехзначное натуральное число. Найдите наибольшую цифру в этом числе. Например, самая большая цифра в числе 263 — это 6.
15. Дано четырёхзначное натуральное число. Определите, есть ли в этом числе цифра 3.

ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

1. Какие математические операции обозначаются символами `//` и `%`?
2. Какие значения примут переменные целого типа `x` и `y` после выполнения фрагмента программы?

```
x = 823
y = 4
x = x // y
y = x % y
```

3. Каким будет последнее значение переменной `c`, если во время выполнения следующего фрагмента программы с клавиатуры ввести числа 40 и 10?

```
a = int(input())
b = int(input())
b = -a / 2 * b
if a < b :
    c = b - a
else:
    c = a - 2 * b
```

4. Составьте программу для определения делимости заданного целого числа на 4.
5. Составьте программу, вводящую с клавиатуры пословицу "Быстрее мысли ничего нет" и выводящую:
 - a) слово "мысли"
 - b) количество букв "е"
 - c) целиком пословицу из заглавных букв
6. Какой метод используется для удаления заданного элемента из списка?
7. Составьте программу, выводящую в обратном порядке буквы заданного предложения.
8. Опишите алгоритм нахождения длины самого короткого слова в предложении.

9. Для каких целей используют подпрограммы?
10. При каком наименьшем входном значении переменной s следующая программа выведет число 16?

```
s = int(input())
n = 64
while s > 121 :
    s = s - 25
    n = n // 2
print (n)
```

11. При каком наименьшем входном значении переменной s следующая программа выведет число 64?

```
s = int(input())
n = 1
while s < 51 :
    s = s + 5
    n = n * 2
print (n)
```

12. Изучите следующий программный код и определите, какую задачу он решает.

```
n = abs(int(input()))
a = (n // 100) % 10
b = (n // 10) % 10
c = n % 10
print(a, b, c)
```

13. Используя алгоритм Евклида для нахождения наибольшего общего делителя двух чисел, напишите программу для вычисления НОД N заданных чисел.
14. Даны N целых чисел. Найдите среди них такие два числа, чтобы их НОД был наибольшим.
15. Даны N целых чисел. Найдите среди них такие два числа, чтобы их НОК был наименьшим.
16. Напишите программу вычисления для заданного натурального числа n произведения $2 \times 4 \times \dots \times (2n)$.

17. Напишите программу нахождения наименьшего натурального числа n , при котором $n!$ делится на 990.
18. Напишите программу, определяющую по заданному числу n количество единиц, присутствующих в числе $n!$.
19. Напишите программу нахождения числа единиц в двоичной записи натурального числа n в интервале $[1, 100]$.
20. Напишите программу нахождения количества натуральных чисел в интервале $[a, b]$, в двоичной записи которых имеется только цифра 1.
21. Напишите программу, определяющую количество чисел Фибоначчи в заданном интервале $[a, b]$.
22. Напишите программу, находящую сумму всех чисел Фибоначчи, находящихся в интервале $[0, 100]$.
23. Алгоритм вычисления значения функции $F(n)$ (где n – натуральное число) задается следующими соотношениями:

$F(n) = 1$, когда $n = 1$;

$F(n) = n + F(n - 1)$, когда n четное;

$F(n) = 2 \times F(n - 2)$, когда $n > 1$ и n нечетно.

Каково значение функции $F(17)$?

24. Что делает эта программа??

```
n = 2
s = input("Введите предложение: ")
l = len(s)
s = s.split(' ')
print(s[n-1])
```

- A) выводит второе слово введенного предложения
- B) выводит первое слово введенного предложения
- C) выводит второй символ введенного предложения
- D) выводит первый символ введенного предложения
- E) выводит последний символ введенного предложения



ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

19. Информационная модель на графах
20. Задачи связанные с графами
21. Топологии сети
22. Веб-сайт
23. Шаблоны веб-сайтов
24. Языки веб-программирования
25. Адресация в Интернете



ключевые слова

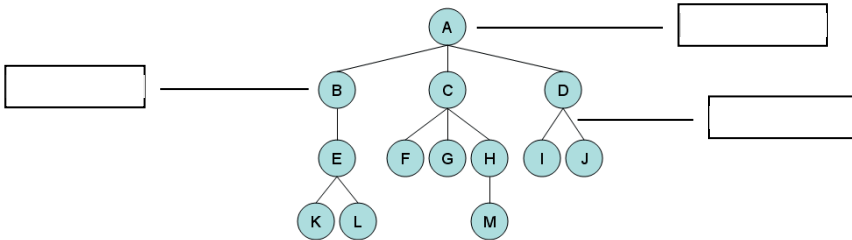
- Граф
- Вершина
- Ребро
- Список смежности
- Матрица смежности
- Связный граф
- Ориентированный граф

19. Информационная модель на графах

- Что такое информационная модель?
- Может ли у одного объекта быть несколько видов информационных моделей?

Деятельность

Назовите указанные элементы данной структуры.

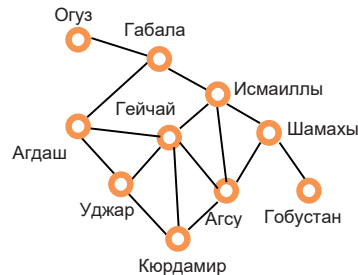


Обсудим:

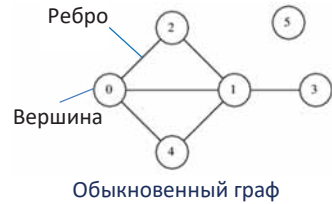
- К какому виду информационных моделей относится данная структура?
- Что такое путь и сколько путей имеется между двумя произвольными вершинами этой структуры?

В 7-м классе вы познакомились с табличной информационной моделью, в 8-м классе – с моделью в виде дерева. А теперь вы познакомитесь еще с одним видом информационной модели – *графом* или *сетью*.

На представленном фрагменте карты показаны автомобильные дороги между районными центрами одного из регионов Азербайджана. Если изобразить районные центры крупными точками (или небольшими кружочками), а дороги – соединяющими их линиями, то можно получить следующую схему.

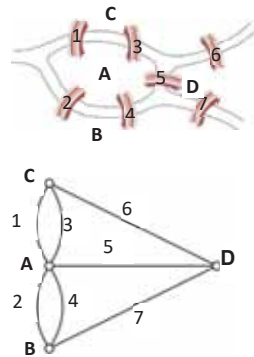


Структура, состоящая из определенного числа объектов, связанных между собой произвольным образом, называется **графом**. В программировании иногда вместо термина "граф" используют термин "**сеть**". Объекты, составляющие граф, называются **вершинами**, а линии, связывающие объекты, называются **ребрами**. Между двумя произвольными вершинами графа соединение может отсутствовать. Чаще всего вершины графа нумеруют или обозначают буквами.



Поводом для создания **теории графов** послужило решение одной занимательной задачи известным математиком Леонардом Эйлером во время посещения им в 1736 году Кёнигсберга (современного Калининграда).

Протекающая по городу Кёнигсбергу река делила его на четыре части, и эти части были соединены между собой семью мостами. На упрощенном плане города мосты указаны цифрами, а части города – буквами. Известная задача того времени формулировалась так: как пройти по всем мостам, не проходя ни по одному из них дважды?



Эйлер обозначил части города соответствующими точками A, B, C и D, а мосты – с помощью линий, соединяющих эти точки (схема справа). Таким образом, задача стала эквивалентной такому заданию: можно ли, не отрывая ручку от бумаги, нарисовать данную фигуру, пройдя по каждой линии только один раз? Эйлер доказал, что эта задача не имеет решения.

Обычно графы представляют двумя способами: **списком смежности** и **матрицей смежности**. В **списке смежности** перечисляются вершины, соединенные с каждой из данных вершин. В этом способе представления имеются повторения, например, если соединены вершины A и B, то в списке смежности вершины A будет вершина B, а в списке смежности вершины B будет вершина A.

Во втором способе представления – **матрице смежности**, граф из количества вершин n представлен в виде таблицы с n -м количеством строк и n -м количеством столбцов (матрица размерности $n \times n$). Если между любой вершиной x и вершиной y есть ребро, то элемент a_{xy} будет равен 1, в противном случае – 0: список смежности и матрица смежности для графа, представленного выше, будут такими:

Вершина	Список смежности
0	1, 2, 4
1	0, 2, 3, 4
2	0, 1
3	1
4	0, 1
5	

	0	1	2	3	4	5
0	0	1	1	0	1	0
1	1	0	1	1	1	0
2	1	1	0	0	0	0
3	0	1	0	0	0	0
4	1	1	0	0	0	0
5	0	0	0	0	0	0

Для сохранения графов в памяти компьютера используют как эти, так и другие способы. Например, если список смежности данного графа в языке Python назовем `adjacency_list`, а количество его вершин `num_vertices`, то:

```
adjacency_list = [[1, 2, 4],
                  [0, 2, 3, 4],
                  [0, 1],
                  [1],
                  [0, 1],
                  [],
                  ]
num_vertices = len(adjacency_list)
```

Если для этого графа обозначить матрицу смежности как `adjacency_matrix`, то:

```
adjacency_matrix = [[0, 1, 1, 0, 1, 0],
                    [1, 0, 1, 1, 1, 0],
                    [1, 1, 0, 0, 0, 0],
                    [0, 1, 0, 0, 0, 0],
                    [1, 1, 0, 0, 0, 0],
                    [0, 0, 0, 0, 0, 0],
                    ]
num_vertices = len(adjacency_matrix)
```

Следует заметить, матрица смежности обыкновенного (не ориентированного) графа всегда симметрична относительно главной диагонали. *Главная диагональ* матрицы проходит от верхнего левого угла к нижнему правому.

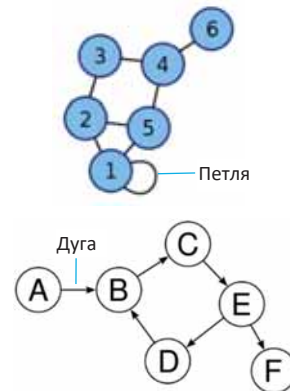
Графы можно хранить и в форме списка ребер. В этом случае каждое ребро обозначается двумя числами – номером начальной и конечной вершины. Количество вершин графа хранится в отдельной переменной, так как *изолированные* (не связанные ни с одной из вершин) вершины не попадают в список ребер.

```
num_vertices = 6          # Количество вершин
edges_list = [[0, 1],    # Список ребер
              [0, 2],
              [0, 4],
              [1, 2],
              [1, 3],
              [1, 4],
              ]
```

Познакомимся еще с некоторыми понятиями, имеющими отношение к графам. Последовательность ребер, в которой последняя вершина каждого ребра является началом другого ребра, за исключением последней, называется *путем*. Замкнутый путь называют *циклом*. Так, в верхнем примере ребра, соединяющие вершины 1, 2, 3, образуют путь, а путь между вершинами 1, 2, 5 – это цикл.

Как было отмечено, совсем не обязательно соединение всех вершин графа между собой. Если в графе существует путь между любыми двумя его вершинами, то его называют **связным** графом.

Если ребро соединяет вершину саму с собой, то такое ребро называют **петлей**. Если ребро имеет определенное направление (например, ребро идет не от вершины В к вершине А, а из вершины А к вершине В), то такое ребро называют **дугой**. То есть ребро соединяет две вершины графа, а дуга начинается из одной вершины и заканчивается в другой. Граф, все ребра которого являются дугами, называют **ориентированным графом**, или **диграфом**.



Ориентированный граф

Для сохранения ориентированных графов в памяти можно использовать (немного изменив предыдущие способы) следующие способы представления:

- в списке смежности для каждой вершины хранить те вершины, к которым ведут ребра;
- в матрице смежности, если имеется ребро от i до j , принять $adjacency_matrix[i][j] == 1$, а если в графе нет ребра в обратном направлении, то $adjacency_matrix[j][i] == 0$;
- в списке ребер каждое ребро хранить в виде [начало, конец].

Дерево (древовидная структура) является одним из видов графа – это связный граф без цикла. То есть между любыми двумя вершинами дерева есть путь, но в дереве нет замкнутых путей.

Информационная модель на графах (сеть) широко используется во многих сферах нашей жизни. Например, новостройки, здания и другие постройки в населенных пунктах можно представить как вершины графа, а пути между ними, линии электропередачи, водопроводы, линии связи и другие коммуникации – как ребра графа. На таких графах можно планировать оптимальные маршруты транспорта, определять кратчайшие пути между объектами.

Изучим сами

Узнайте в Интернете (например, на сайте www.gomap.az) расстояние между населенными пунктами Азербайджана, указанными на схеме в начале темы. На основе полученных данных постройте граф. Определите на нем кратчайший путь между Агдашем и Шамахой.

Проверьте себя

1. Что такое граф и из каких элементов он состоит?
2. Как называется связный граф без дуг?
3. В какой форме хранятся графы в памяти компьютера?
4. Постройте граф региона, в котором вы живете, и составьте на его основе соответствующую матрицу смежности.

- Взвешенный граф
- Весовая матрица

20. Задачи связанные с графами

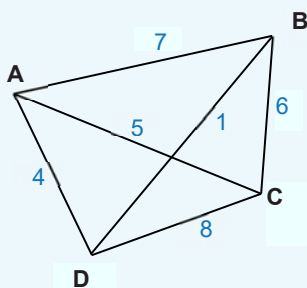
- Как называется ребро, имеющее направление?
- Как называется граф, у которого все ребра имеют направление?

При решении некоторых практических задач (например, нахождении кратчайшего пути между пунктами) имеют значение не только связи между вершинами, но и числа, соответствующие этим связям. Этими числами могут быть, например, расстояния между городами или стоимость проезда. В теории графов числовое значение, поставленное в соответствие каждому ребру графа, называют его **весом**, а такой граф – **взвешенным графом**.

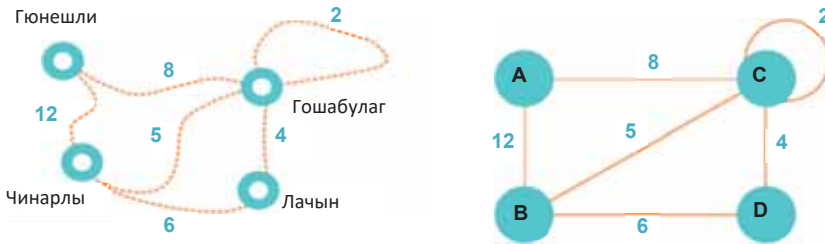
Из каждого из пунктов А, В, С и D имеется путь в остальные пункты, расстояния между которыми известны: $AB = 7$, $AC = 5$, $AD = 4$, $BC = 6$, $BD = 1$, $CD = 8$. Необходимо, начиная от одного из этих пунктов и побывав в каждом из пунктов только один раз, вернуться в исходный пункт. Какой маршрут надо выбрать, чтобы путь оказался кратчайшим?

Решение. Соответствующие пункты и схему путей между ними можно показать при помощи взвешенного графа. Как видите, здесь имеется 6 возможных циклов: ABCDA, ACBDA, ABDCА, ACDBA, ADBCА, ADCBA. Их длина, соответственно, равна: 25, 16, 21, 21, 16, 25. Таким образом, самыми короткими будут маршруты ACBDA и ADBCА.

1



Во взвешенном графе вместо матрицы смежности используют **весовую матрицу**. В ячейках весовой матрицы указывается вес ребер, и если между двумя вершинами нет ребра, соответствующая ячейка остается пустой. На рисунке показана схема, на которой указаны длины путей, соответствующий ей граф и весовая матрица.



	A	B	C	D
A		12	8	
B	12		5	6
C	8	5	2	4
D		6	4	

Что можно определить при помощи весовой матрицы? Во-первых, можно узнать, имеется ли ребро между двумя вершинами, и если имеется, то какова его длина (вес). Для этого достаточно посмотреть в соответствующую ячейку. Например, между вершинами B и C есть ребро и его вес равен 5. Во-вторых, если предположить, что вес ребер указывает расстояние, можно определить длину пути. Например, длина пути **ABCD** равна сумме длин ребер **AB**, **BC** и **CD**: $12 + 5 + 4 = 21$. И наконец, при помощи весовой матрицы можно начертить сам граф.

2

В галактике Млечный Путь на планете Нептун 6 городов, и они последовательно пронумерованы, начиная с 1. Некоторые города соединены дорогами. Император галактики Максимус принимает решение составить список этих дорог на планете. Но так как он слаб в математике, просит у вас помощи.

Решение. Если представить города планеты Нептун с помощью вершин, а дороги между ними – с помощью ребер, то получим обыкновенный граф. В задаче требуется найти количество ребер. Матрица смежности, соответствующая этому графу, может выглядеть приблизительно так:

```

0 1 1 0 1 0
1 0 1 1 1 0
1 1 0 0 0 0
0 1 0 0 0 1
1 1 0 0 0 1
0 0 0 1 1 0

```

Здесь 1 на пересечении i -й строки и j -го столбца указывает на наличие дороги между соответствующими городами. Если такая дорога имеется, следовательно, на пересечении j -го столбца и i -строки тоже будет 1. То есть, если из города i в город j есть дорога, следовательно, из города j в город i тоже есть дорога. Значит, для решения поставленной задачи необходимо в матрице смежности найти количество единиц и результат разделить на 2. Таким образом, в соответствии с матрицей смежности, данной выше, на планете Нептун 8 дорог.

Программу решения этой задачи на языке Python можно представить так:

```
i = 1
w = 0 # w - количество дорог
while i <= 6:
    s = input() # Вводится одна строка матрицы.
    w = w + s.count('1') # Подсчитывается число 1 в этой
                        # строке и это число добавляется
                        # к общему числу дорог

    i = i + 1
w = w // 2
print(w)
```

Шахматный турнир проводится по круговой системе, при которой каждый участник встречается с каждым ровно один раз. В турнире принимают участие 7 школьников. Известно, что Ариф провел шесть партий, Бякир – пять, Джейхун и Дадаш – каждый по три, Эльхан и Али – каждый по две, а Илькин сыграл одну партию. С кем сыграл Джейхун?

3

Решение. Построим граф отражающий встречу игроков. Вершины этого графа отметим числами от 1 до 7 и зададим такое соответствие: 1 – Ариф, 2 – Бякир, 3 – Джейхун, 4 – Дадаш, 5 – Эльхан, 6 – Али, 7 – Илькин. G ,



Так как степень вершины 1 равна 6 (Ариф провел 6 игр), то эта вершина соединена со всеми остальными вершинами. Так как степень вершины 7 равна 1, смежной с ней будет только вершина 1.

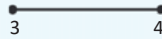
Рассмотрим подграф H_1 , состоящий из множества вершин $\{2, 3, 4, 5, 6\}$. Этот подграф можно образовать удалением вершин 1, 7 и выходящих из них ребер из графа G . Поэтому в графе H_1 , состоящем из пяти вершин, степени вершин будут такими:

$$d(2) = 4, d(3) = d(4) = 2, d(5) = d(6) = 1.$$

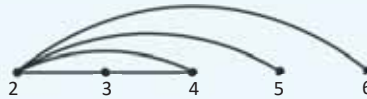
В графе H_1 вершина 2 имеет смежность со всеми вершинами, вершины 5 и 6 смежны только с двумя вершинами.

Теперь рассмотрим подграф H_2 , состоящий из множества вершин $\{3, 4\}$. Этот граф получается при удалении из графа H_1 вершин 2, 5, 6 и ребер, выходящих из них.

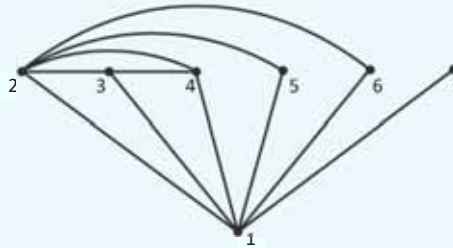
В графе H_2 $d(3) = d(4) = 1$, то есть это граф, представленный на рисунке:



Если вернуть удаленные вершины 2, 5, 6, получится граф H_1 :



Теперь вернем удаленные вершины 1 и 7 и получим необходимый граф G :



Этот граф отражает встречу школьников во время соревнований.

Из этого графа видно, что Джейхун (3-я вершина графа) сыграл с Арифом, Бекиром и Дадашем, которые соответствуют вершинам 1, 2 и 4. Понятно, что при помощи этого графа нетрудно определить, с кем играли остальные участники соревнований.

Изучим сами

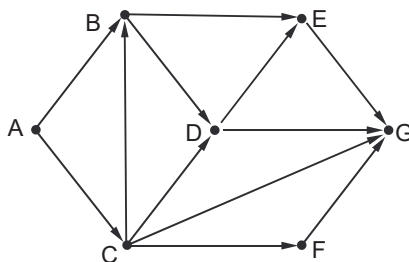
В галактике Млечный Путь на планете Нептун имеется N городов, которые пронумерованы последовательно, начиная с 1. Некоторые города соединены дорогами. Император галактики Максимус принимает решение составить список этих дорог на планете. Но так как он слаб в математике, просит помочь ему посчитать количество дорог.

Проверьте себя

1. Как называется граф, каждому ребру которого соответствует определенное число?
2. Какие свойства обязательно будут в графе, у которого весовая матрица не симметрична относительно главной диагонали: имеет цикл; взвешенный; ориентированный; нет цикла; связный?
3. Сколько ребер в данной весовой матрице? Чему равен вес ребра, соединяющего вершины A и E?

	A	B	C	D	E
A		5	2		6
B	5			5	
C	2			2	
D		5	2		3
E	6			3	

4. Если в весовой матрице числа показывают расстояние между пунктами, чему будет равна длина пути A-B-D-E?
5. На рисунке дана схема дорог, соединяющих города A, B, C, D, E, F, G. По каждой дороге можно перемещаться только в указанном направлении. Сколько маршрутов ведет из города A в город G?



ключевые слова

- Локальная сеть
- Глобальная сеть
- Сеть "клиент-сервер"
- Одноранговая сеть
- Топология
- Топология "шина"
- Топология "кольцо"
- Топология "звезда"

21. Топологии сети

- Что такое компьютерная сеть?
- Что подразумевается под локальной сетью?

Деятельность

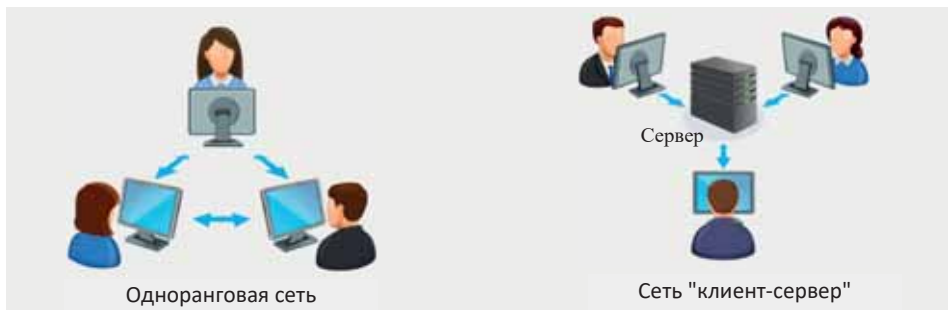
Постройте граф, соответствующий компьютерной сети вашей школы. Для этого каждый компьютер отметьте как **вершину графа**. Если нет сети, то представьте, что в классе перед каждым учащимся имеется компьютер, который связан с другими компьютерами класса с помощью кабеля.

Обсудим:

- Какой граф вы начертили для компьютерной сети – обыкновенный или ориентированный? Ответ обоснуйте.
- Если вы удалите из графа какое-нибудь ребро (то есть прервете связь между двумя компьютерами), как это отразится на работе всей компьютерной сети?

Как вы знаете, компьютерная сеть может состоять из нескольких компьютеров, но может объединять и миллионы компьютеров. По территориальному признаку компьютерные сети делят на несколько видов, среди которых наиболее распространены **локальные** и **глобальные сети**. Как правило, специалисты называют их сокращенными именами – соответственно **LAN** (Local Area Networks) и **WAN** (Wide Area Networks). Как видно из названий, локальные сети охватывают ограниченные, а глобальные – более широкие географические территории.

Хотя существуют локальные сети разных типов, в основном используются два из них: сеть **"клиент-сервер"** и **одноранговая сеть**.

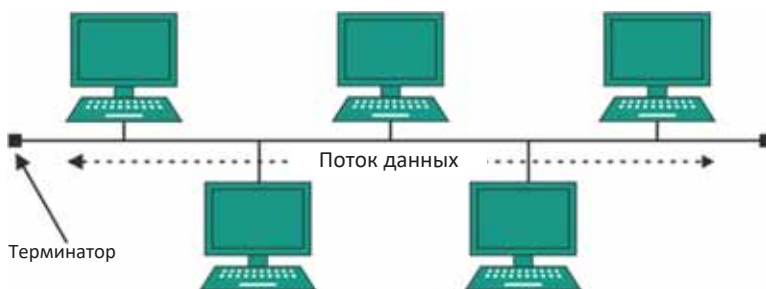


Сеть "клиент-сервер" (client-server network). В локальной сети такого типа для максимального использования возможностей рабочих станций и серверов управление разделено между ними. В этой архитектуре обработка прикладных программ распределена между клиентом-компьютером и компьютером-сервером. Как правило, в качестве сервера используется мощный компьютер. Сервер дает клиенту традиционные преимущества работы в многопользовательской среде. Это управление данными, коллективная работа с информацией, сетевое администрирование данных и возможности защиты их.

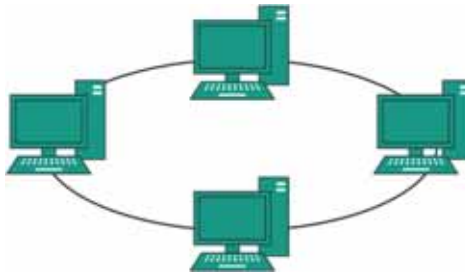
Одноранговая сеть (peer to peer network). В сети такой архитектуры все компьютеры равноправны между собой. Здесь ни один компьютер не выступает в роли сервера. Пользователи в сети сами определяют, какими файлами, имеющимися в компьютере, они будут делиться с другими пользователями. Устанавливать такие сети и управлять ими очень легко. Небольшие офисы, как правило, отдают предпочтение одноранговым сетям.

При создании сети большое значение имеет **топология**, то есть схема расположения и соединения сетевых устройств и кабелей сети. Необходимо выбрать такую структуру (граф), чтобы сеть работала надежно и эффективно и можно было бы с легкостью управлять потоком информации. Большинство сетей строится на основе трех топологий:

1. **Топология "шина"**. В этой топологии все компьютеры локальной сети подключены к одному общему кабелю. Данные, отправленные в такую сеть, передаются всем компьютерам в сети. Каждый компьютер проверяет, ему ли направлена информация. Если информация направлена ему, он принимает ее и обрабатывает. Эту топологию называют также **линейной топологией**. Она очень проста и обходится дешево (используется мало кабеля), но имеет некоторые недостатки. Неисправность какой-то части этой сети останавливает всю ее работу. Именно из-за этого недостатка очень популярная ранее топология "шина" сейчас практически не используется.



2. **Топология "кольцо"**. В этой топологии каждый компьютер соединен с двумя другими компьютерами, принимает информацию от одного компьютера и передает ее другому. Последний компьютер подсоединен к первому компьютеру, и таким образом кольцо замыкается. У этой топологии тоже больше недостатков, чем преимуществ. Например, сложно вести поиск неисправностей в сети. Выход из строя одного компьютера оказывает влияние на работу подсоединенных к нему компьютеров, а в итоге и всей сети. Поэтому популярные ранее кольцевые сети сейчас можно встретить очень редко.



3. **Топология "звезда"**. В топологии, которая возникла в первые годы компьютерных технологий и называлась *активной звездой*, все пользователи в сети были подключены к мощному центральному компьютеру. В такой конфигурации все потоки данных передавались только через центральный компьютер; Центральный компьютер также отвечал за управление обменом информацией между всеми частями сети. В такой сетевой организации центральный компьютер был настолько перегружен, что обычно обслуживал только сеть. Его сбой вызвал отключение всей сети, однако сбой или отключение любого периферийного компьютера не повлияло на работу остальной части сети. В настоящее время такие сети очень редки.



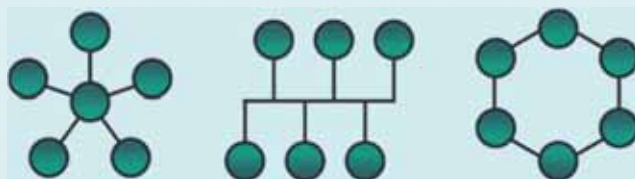
В настоящее время очень распространен похожий тип сетей – *звезда-шина* или *пассивная звезда*. Здесь периферийные компьютеры подключены к пассивному концентратору (hub), а не к центральному компьютеру. В отличие от центрального компьютера, концентратор не отвечает за обмен данными, он только восстанавливает входящие сигналы и отправляет их на другие компьютеры и подключенные к нему устройства.

Несмотря на большой расход кабеля, характерный для сетей типа "звезда", эта топология имеет существенные преимущества перед остальными, что и обусловило ее широчайшее применение в современных сетях. Эти сети надежны – подключение к центральному концентратору и отключение компьютеров не отражается на работе остальной сети. Обрывы кабеля влияют только на единичные компьютеры. Такие сети также просты в обслуживании и устранении неполадок. Поэтому эта топология широко используется в современных сетях.

Современные компьютерные сети постоянно расширяются и модернизируются. Поэтому такие сети почти всегда бывают *гибридными*, то есть состоящими из комбинации нескольких базовых топологий.

Изучим сами

Определите, каким сетевым топологиям соответствуют следующие графы.



На какой из этих графов похож граф, который вы построили в блоке "Деятельность"? Для построенных графов создайте весовую матрицу, принимая за вес каждого ребра его длину.

Проверьте себя

1. На какие виды принято подразделять компьютерные сети по территориальному признаку?
2. Откуда произошли названия "клиент-сервер" и "одноранговая сеть"?
3. Что такое топология сети и какие топологии используют в компьютерных сетях?
4. Если в вашей школе есть компьютерная сеть, определите, по какой топологии она устроена.
5. Все ли виды топологии сетей являются графами?

22. Веб-сайт

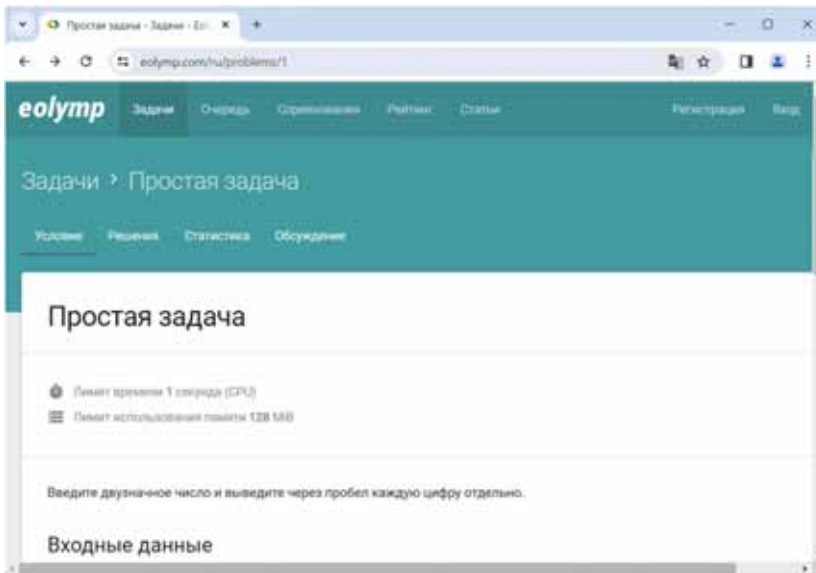
- Что такое веб-страница и чем она отличается от веб-сайта?
- Как называются программы для демонстрации веб-страниц?

ключевые слова

- Веб-сайт
- Веб-страница
- HTML
- Тег
- Главная страница

Деятельность

1. Запустите произвольный веб-браузер.
2. Перейдите на сайт `eolymp.com`.
3. Откройте список **Сложность** в меню **Задачи** и выберите пункт **Простая**. Выберите и откройте из списка любую задачу и ознакомьтесь с ее условием. Затем зайдите в подраздел **Статистика** и посмотрите, кто решил выбранную задачу.
4. Откройте меню **Рейтинг**. Нажмите на имя пользователя с самым высоким рейтингом в таблице лидеров и ознакомьтесь на открывшейся странице с его результатами.



Обсудим:

- Какой язык программирования наиболее популярен среди пользователей сайта?
- Какой процент пользователей программирует на языке Python?
- Сколько человек правильно решили задачу под именем «Простая задача»?

Выход компьютера в Интернет часто осуществляется посредством телефонной сети. К сожалению, скорость передачи информации по телефонной линии весьма невелика. Поэтому в Интернете для уменьшения трафика, то есть количества символов, передаваемых за единицу времени, пересылают не сам документ, а его описание на особом языке. Для описания веб-документов используется язык **HTML** (HyperText Markup Language, язык разметки гипертекста) (произносится как "аш-ти-эм-эл"). Получив описание на этом языке, браузер воссоздает исходный документ, формируя текст и расставляя иллюстрации в нужных местах. Каждый раз, открывая ту или иную веб-страницу, мы на самом деле открываем документ, написанный на языке HTML. Все веб-документы отформатированы с помощью языка HTML.

Для того чтобы показать браузеру или другой программе, как выводить информацию на экран, в языке HTML используют набор *тегов*. *Теги* – это команды, определяющие правила отображения текста в браузере. Тег всегда начинается открывающейся угловой скобкой (знак <) и заканчивается закрывающей угловой скобкой (знак >). С помощью тегов в документе можно указать разделы, абзацы, списки, рисунки, таблицы, колонтитулы и другие объекты. В пределах каждого блока можно изменить шрифт, размер, цвет символов, выделить курсивом или сделать жирным текст. Создателем языка HTML считается Тим Бернерс-Ли (Tim Berners-Lee).



Для создания веб-страниц или веб-сайтов можно использовать несколько методов:

1. Создать документ HTML в простом текстовом редакторе (например, Notepad или WordPad).

2. Создать документ в текстовом процессоре (например, Microsoft Word или OpenOffice.org Writer) и сохранить веб-страницу в формате HTML.
3. Использовать особые программы – визуальные **HTML-редакторы** (например, Adobe Dreamweaver, Microsoft FrontPage, Nvu) или воспользоваться **конструкторами сайтов** (например, uCoz).

Независимо от того, какой метод будет использован при создании сайта, первоначально нужно определить его *содержание* и *структуру*.

Предположим, вы посещаете кружок "Юный программист", и представление структуры сайта, отображающего деятельность данного кружка, в форме графа дает возможность наглядно представить его содержание, а также помогает организовать переход с одной страницы сайта на другую.



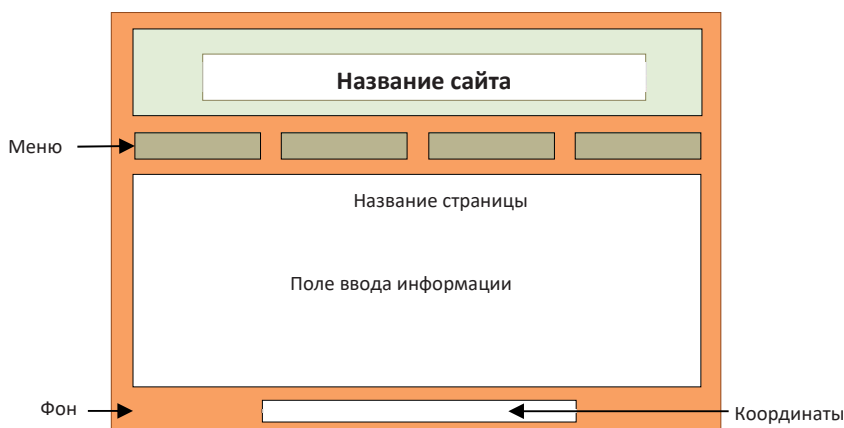
При загрузке любого сайта сначала открывается его **главная (домашняя) страница (home page)**. Эта страница в первую очередь отвечает на вопрос: "О чем этот сайт?". Для этого на главной странице размещается название сайта, графическое изображение, соответствующее теме сайта, краткий текст, отражающий содержание сайта, а также главное меню для перехода к основным разделам сайта. На главной странице может быть размещено имя

создателя, контактная информация для связи с ним, новости и другие информационные блоки.

На каждой странице сайта, как правило, имеются несколько постоянных элементов, которые размещаются в одних и тех же местах:

- *заголовок сайта*, размещенный в самом верху страницы;
- главное меню, расположенное, как правило, в верхней или левой части страницы (количество пунктов меню, их названия, порядок следования неизменны на всех страницах сайта).

Рекомендуется разрабатывать все страницы сайта в одном и том же стиле. Для сохранения стиля желательно заранее разработать шаблон одной страницы сайта и отразить на ней элементы, характерные для всех страниц.



Изучим сами

Выясните структуру сайта ict.az и представьте ее в форме графа. Почему для структуры сайта приемлема не древовидная информационная модель, а информационная модель в форме графа?

Проверьте себя

1. Почему для описания веб-документов используют специальный язык?
2. Что такое тег и как он представляется в документе HTML?
3. Какими способами создают веб-сайты?
4. Что такое "Главная страница" и что располагается на ней?

ключевые слова

- Информационное общество
- Электронное правительство

23. Шаблоны веб-сайтов

На прошлом уроке вы узнали, что существует много разных способов для создания веб-сайтов. Один из них – воспользоваться услугой, которую предлагают определенные сайты в Интернете. Эти сайты позволяют пользователям, которые не знают языка HTML и не имеют опыта создания сайтов, за короткий период без финансовых вложений создать свой собственный сайт.

Сейчас вы узнаете, как на одном из таких сайтов с помощью готовых шаблонов разработать сайт, посвященный информационному обществу. Для этого вначале ознакомьтесь с материалами, которые даны после блока "Шаг за шагом", и определите структуру сайта, который будете создавать.

Шаг за шагом

Разработка веб-сайта на основе шаблонов

1. Введите в адресную строку браузера адрес www.simplosite.com и нажмите клавишу <Enter>. Откроется главная страница сайта.



2. Выберите из меню пункт **Make a free website or blog**. Откроется следующая страница.



3. Щелкните по кнопке **Personal website or a blog**.
4. Выберите на открывшейся странице цветовую схему сайта и щелкните по кнопке **Next: Images**.

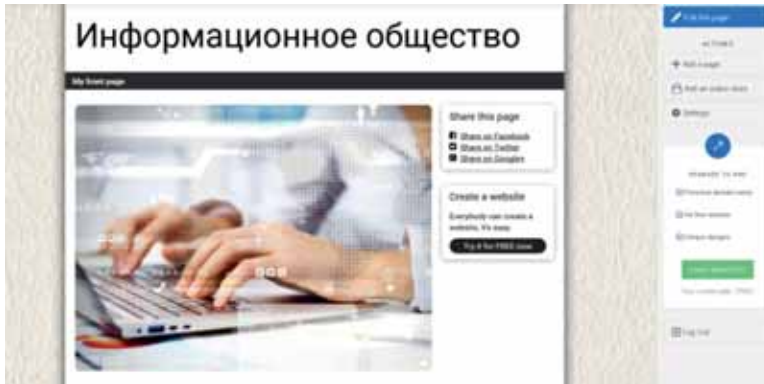


5. Для главной страницы будут предложены различные изображения. Выберите то, которое соответствует вашей теме, и щелкните по кнопке **Next: Backgrounds**.
6. Выберите из предложенных одно изображение для фона страницы сайта.



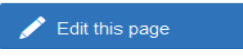
7. Щелкните по кнопке **Next: Headline**. Введите заголовок сайта на открывшейся новой странице. Например, **Информационное общество**.
8. Щелкните по кнопке **Save and continue**.
9. На открывшейся странице для задания имени сайта введите соответствующее название в указанное поле (например: **info-obshestvo**). Если введенное имя уже используется кем-то, то появится сообщение об этом.
10. После задания имени сайта будет выведено сообщение об успешном создании сайта.
11. На следующей странице задайте пароль для администрирования сайта.
12. Указав адрес электронной почты, щелкните по соответствующей кнопке.
13. Если все шаги будут выполнены правильно, браузер отобразит главную страницу вашего сайта.

14. Для подтверждения адреса сайта ознакомьтесь с письмом, пришедшим на ваш электронный адрес, и перейдите по указанной ссылке. Адрес вашего сайта будет состоять из имени **simplesite** и имени, которое вы выбрали сами: **info-obshchestvo.simplesite.com**.



15. Для размещения на сайте материалов по теме можно создать дополнительные страницы. Для этого на панели, находящейся справа, щелкните по кнопке **Add a page**. Откроется окно с образцами страниц разного типа. Выберите вариант **Standard page**.



16. Для размещения материала на странице и его редактирования щелкните по кнопке  , расположенной в верхнем правом углу. Ознакомьтесь в открывшемся окне с инструментами редактора и выясните функцию каждого инструмента.
17. Разместите информацию об информационном обществе на странице.
18. Соберите в Интернете информацию об основных направлениях формирования информационного общества и разместите ее на второй странице сайта.
19. Создайте страницы **Фотоальбом** и **Видео** и, отыскав рисунки и видеоматериалы по соответствующей теме, разместите их на этих страницах.

Во второй половине прошлого века для хранения, передачи и обработки информации в большом объеме стало неизбежным использование современных средств. Большая часть населения Земли перешла из сферы материального производства в сферу работы с информацией. Появился новый социальный слой населения, непосредственно не производящий материальные ценности. Возрастание доли умственного труда увеличило и число людей, выбирающих сферы деятельности, связанные с этим трудом. Началось формирование нового общества, которое ученые назвали "информационным обществом".

Информационное общество – это такое общество, большая часть членов которого занята производством, хранением, обработкой и использованием информации.

Вот некоторые характерные особенности информационного общества:

1. *Объемы информации стремительно возрастают и человек привлекает для ее обработки и хранения специальные технические средства.*
Объем информации увеличился настолько, что человек не может сам обработать ее, и поэтому привлекает к этой работе особые технические средства. Компьютеры создают возможность для использования надежных информационных источников, уменьшают непродуктивную деятельность, ускоряют принятие оптимальных решений и автоматизируют обработку информации.
2. *Стремительно развиваются компьютерная техника, компьютерные сети, информационные технологии.* Количество пользователей сети Интернет исчисляется миллионами. Использование современных мультимедийных систем, объединяющих в себе функции различных устройств (компьютера, телевизора, радио, телефона и т.д.), приводит к универсализации информационных технологий. А устройства, хранящие информацию, постоянно уменьшаясь, уже умещаются на ладони. В этих устройствах умещаются и личные, и универсальные справочники объемом в несколько энциклопедий. Подсоединив эти устройства к сети Интернет, можно получить оперативную информацию, например, о погоде или пробках на дорогах.
3. *Происходит переоценка ценностей, уклада жизни и изменяется культурный досуг.*
Уже сегодня компьютерные игры занимают большую часть свободного времени человека. Сейчас они трансформируются в сетевые игры с участием нескольких удаленных партнеров. Растет время, проведенное в Интернете, где можно путешествовать по образовательным сайтам, виртуальным музеям, читать литературу и т.д. Большой популярностью пользуются службы общения онлайн, которые позволяют общаться с людьми на расстоянии в режиме реального времени.
4. *В домах у людей появляются различные электронные приборы и компьютеризированные устройства.*

Система "умного дома" оснащается вместо системы проводов одним силовым и одним информационным кабелем, который возьмет на себя все информационные связи, включая каналы кабельного телевидения и выход в Интернет. Специальный электронный блок контролирует бытовую технику и все системы жизнеобеспечения. К "умным зданиям" добавятся "умные автомобили", в которых кроме компьютера, следящего за технической частью автомобиля, будет функционировать система, связанная с городскими информационными службами. Такой автомобиль будет связан с «умным домом» и даже сможет им управлять.

5. *Производством энергии и материальных продуктов будут заниматься машины, а человек – главным образом, обработкой информации.*

На производстве происходит сокращение людей на сборочных линиях, и внедряются роботы и манипуляторы.

6. *В сфере образования создается система непрерывного образования.*

Человек получит возможность учиться всю жизнь, чтобы не отстать от времени, иметь возможность сменить профессию, занять достойное место в обществе.

7. *Дети смогут обучаться на дому с помощью компьютерных программ и телекоммуникаций.*

В связи с этим в процессе обучения меняются его формы, и возникают проблемы с воспитательным аспектом обучения.

8. *Появляется и развивается рынок информационных услуг.*

Информация становится товаром и особым видом услуг, которые можно как обычный товар купить или продать.

Изучим сами

Зайдите на Портал электронного правительства (<https://www.e-gov.az/>) и ознакомьтесь с услугами, которые предлагают различные государственные структуры. Зарегистрируйтесь на портале и, используя услугу Министерства науки и образования "Электронная справка о месте получения образования учащихся общеобразовательных школ", получите электронную справку о себе..

Проверьте себя

1. Какие преимущества имеет создание сайтов с помощью готовых шаблонов?
2. Как разместили текст на созданном вами сайте?
3. Что вы узнали об информационном обществе?
4. Назовите основные направления формирования информационного общества.

- HTML
- CSS
- JavaScript
- Атрибут стиля
- Свойство
- Значение

24. Языки веб-программирования

В сфере веб-программирования наряду с HTML широко используются такие языки, как **CSS** и **JavaScript**.

Язык программирования CSS. Часто используется в сочетании с HTML. Он позволяет изменять внешний вид HTML-элементов, отображаемых в браузере. Хотя CSS (Cascading Style Sheets) обычно переводится как «каскадные таблицы стилей», иногда его сокращенно именуют как «*таблицы стилей*» или просто «*стили*». С помощью CSS можно изменить размер и форму HTML-элементов и сделать их красочными. При этом элементы можно перемещать по экрану. Кроме того, CSS позволяет вносить и другие изменения во внешний облик веб-страницы.

Чтобы изменить HTML-элемент с помощью CSS, необходимо добавить в открывающий тег **атрибут стиля**. Его можно добавить ко всем тегам:

```
<p style="CSS-property: value;">В библиотеке одного учёного была обнаружена новая рукопись «Китаби-Деде Коркуд».</p>
```

Атрибут стиля

CSS

Значение атрибута стиля задается с помощью знака равенства (=) и заключено в двойные кавычки (" "). Чтобы применить CSS к тегу HTML, необходимо присвоить CSS-значение атрибуту style.

CSS – очень простой язык. В нем все разделено на две части: *свойство* и *значение*. **Свойство** сообщает браузеру, какую часть HTML-элемента следует изменить. В **значении** указано, какие изменения потребуются:

CSS	Что это значит?	Примеры значений
свойство	То, что вы хотите изменить	background-color; height;
значение	Как вы хотите это изменить	red; 200px;

В CSS между свойством и значением ставится двоеточие (:). После значения ставится точка с запятой (;). Если в CSS нужно указать более одного слова, между ними ставится дефис (-). Браузер не поймет инструкцию, если ее синтаксис неверен. **Синтаксис** – это набор правил, определяющих способ структурирования и написания кода.

Атрибут стиля всегда следует писать следующим образом:

```
style="CSS-property: value;"
```

Дефис

Двоеточие

Точка с запятой

В браузер встроены сотни различных CSS-свойств и значений, которые разместив в HTML-тегах можно изменить внешний вид страниц.

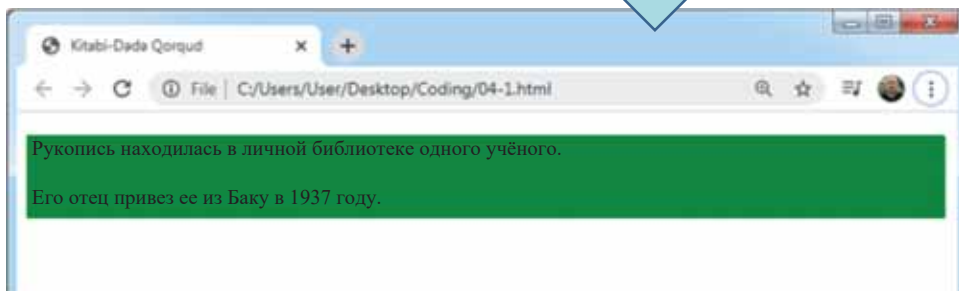
Рассмотрим процесс добавления CSS в теги <div>. Чтобы изменить цвет фона одного раздела понадобится следующий CSS и HTML-код:

Атрибут стиля

Свойство CSS

Значение CSS

```
<body>
  <div style="background-color: green;">
    <p>Рукопись находилась в личной библиотеке одного учёного.</p>
    <p>Его отец привез ее из Баку в 1937 году.</p>
  </div>
</body>
```



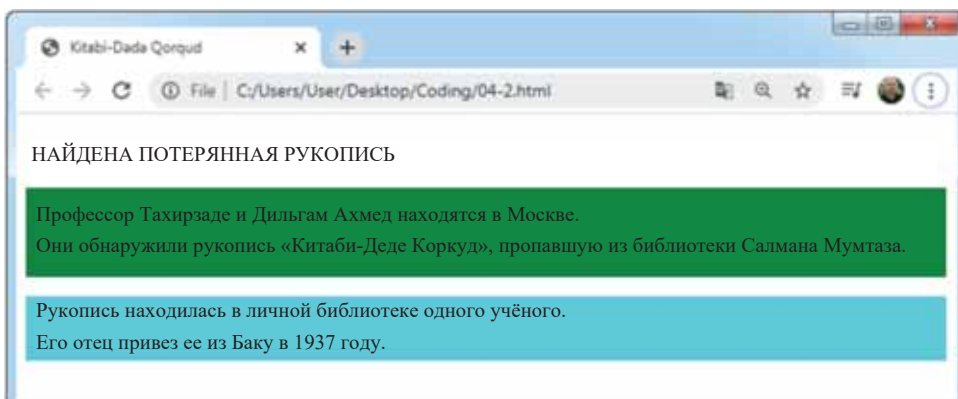
CSS-свойство background-color. Имея представление о CSS, можно добавить цвет на страницу, используя CSS-свойства.

Теги <div> можно использовать для изменения цвета различных областей страницы. Для начала, атрибуты стиля добавляются ко всем открывающим тегам <div>. CSS-свойство background-color помещается внутри этих

атрибутов, и отображается цвет CSS-значения. В результате будет получен такой код:

```
<body>
  <p>НАЙДЕНА ПОТЕРЯННАЯ РУКОПИСЬ</p>
  <div style="background-color: green;">
    <p>Профессор Тахирзаде и Дильгам Ахмед находятся в
      Москве.<br/>
      Они обнаружили рукопись «Китаби-Деде Коркуд», пропавшую
      из библиотеки Салмана Мумтаза.</p>
  </div>
  <div style="background-color: cyan;">
    <p>Рукопись находилась в личной библиотеке одного
      учёного.</p>
    <p>Его отец привез ее из Баку в 1937 году.</p>
  </div>
</body>
```

Свойство background-color



Язык программирования JavaScript. Язык HTML недостаточен для обеспечения интерактивности веб-страниц, для этого используются дополнительные инструменты. Одним из таких инструментов является *язык программирования JavaScript*. С помощью этого языка создают программы (например: анкеты, заполняемые пользователем или формы регистрации) и встраивают их в HTML-код. Используя JavaScript, можно изменить страницу, изменить стиль элементов, удалить или добавить новые теги. В то же время этот язык позволяет отслеживать все «перемещения», совершаемые пользователем на странице (нажатие любой клавиши или кнопки мыши, перелистывание страницы, увеличение и уменьшение масштаба рабочей области экрана и т. д.). Многие другие операции можно выполнить с помощью JavaScript.

Программы, написанные на языке JavaScript, можно размещать в любом месте HTML-файла с помощью тега `<script>`. Код JavaScript необходимо размещать между открывающим и закрывающим тегами `<script>`, иначе код не будет работать. HTML-документ может содержать любое количество тегов `<script>`, и эти теги помещаются в теги `<head>` или `<body>`. Открывающий и закрывающий теги `<script>` и инструкции, расположенные между ними, называются блоком `<script>`.

```
<!DOCTYPE HTML>
<html >
<head>
  <title>Использование JavaScript</title>
</head>
<body>
  <p>Начало документа...</p>
  <script>
    alert( 'Hello World!' );
  </script>
  <p>... Конец документа</p>
</body>
</html >
```

В этом примере используются элементы `<script> ... </script>`. Тег `<script>` содержит исполняемый код. Браузер:

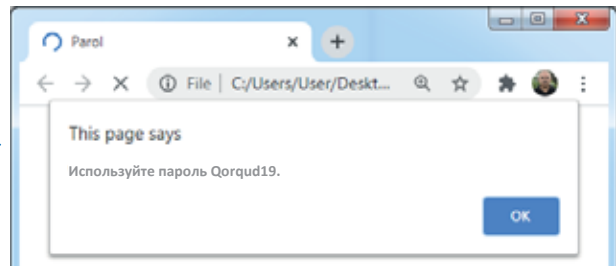
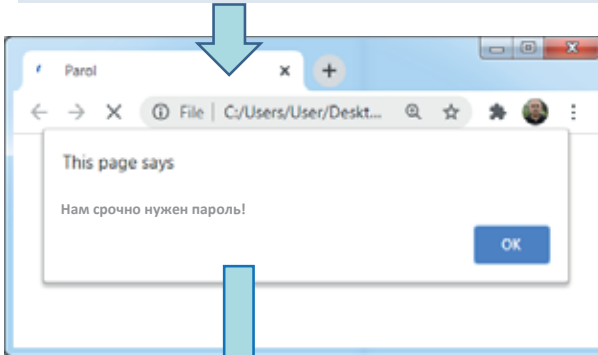
1. Отображает страницу до тега `<script>`.
2. Когда он видит тег `<script>`, он переключается в режим JavaScript и выполняет инструкцию внутри него.
3. По завершении выполнения он возвращается в HTML-режим и отображает остальную часть документа.

Команды, которые сообщают браузеру, что делать, в JavaScript называются **директивами**. Обычно инструкция начинается с ключевого слова, которое сообщает, что делать. Точка с запятой (;) всегда ставится в конце инструкции. Браузер запускает запросы в том порядке, в котором они появляются.

```
<!DOCTYPE html >
<html >
<head>
  <title>Parol </title>
</head>
<body>
  <script>
    alert("Нам срочно нужен пароль!");
    alert("Используйте пароль Qorqud19. ");
  </script>
</body>
</html >
```

Точка с запятой

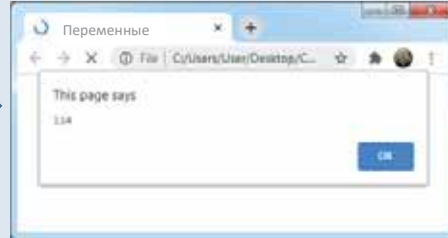
Инструкции



Две инструкции, заключенные между открывающим и закрывающим тегами `<script>`, запускаются одна за другой, в результате чего открываются два окна сообщений.

Переменные. Переменные являются важной частью языка JavaScript.

```
<html >
<head>
  <ti tle>Переменные</ti tle>
</head>
<body>
  <scri pt>
    var bookPages = 114;
    al ert(bookPages);
  </scri pt>
</body>
</html >
```



В этом примере между открывающим и закрывающим тегами `<scri pt>` создается переменная с именем `bookPages`, которой присваивается значение 114. Затем встроенная в браузер функция `al ert()` отображает значение переменной `bookPages`.

Изучим сами

JavaScript часто путают с языком программирования Java, но это разные языки. Узнайте разницу этих языков, используя Интернет.

Проверьте себя

1. Какой тег используется для размещения программы, написанной на JavaScript, в HTML-файле?
2. Используя свои знания Python, определите, какое значение переменной `x` будет напечатано на экране после выполнения следующего фрагмента кода.

```
<scri pt>
  var x = 10;
  whi le (x > 0) {
    x = x - 3;
  }
  al ert(x);
</scri pt>
```

КЛЮЧЕВЫЕ СЛОВА

- Доменное имя
- Система доменных имен
- IP-адрес
- DNS-адрес
- Протокол
- Провайдер
- Хостинг

25. Адресация в Интернете

- Какому количеству разных объектов можно дать адрес при помощи 8 последовательных бит? А при помощи 32 бит?
- Чем отличаются компьютеры в сети друг от друга?

Деятельность

1. Наберите в адресной строке браузера <http://www.speedguide.net/ip/> и нажмите клавишу <Enter> .
2. В поле Search IP address or hostname открывшейся страницы наберите **ict.az** и нажмите клавишу <Enter>. Откроется информация о том, где находится сервер сайта ict.az и карта размещения.

Search IP address or hostname:
Your IP address: 94.20.147.66

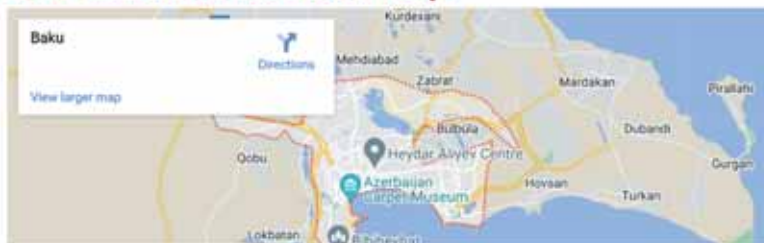
62.212.230.48 IP address Information

The IP address 62.212.230.48 was found in Baku, Azerbaijan. It is allocated to Delta Telecom Ltd. Additional IP location information, as well as network tools are available below.

IP address: **62.212.230.48**
hostname: 62.212.230.48
ISP: Delta Telecom Ltd
ASN: AS29049
City: Baku
Country: Azerbaijan (AZ)
latitude: 40.3771
longitude: 49.8875

[traceroute](#) [check latency](#) [whois](#) [BGP routing info](#) [blacklist check](#)

62.212.230.48 Location Map



Обсудим:

- На что указывает аббревиатура ISP, являющаяся сокращением слов "Internet Service Provider"?
- Где находится сервер сайта ict.az?

Каждый компьютер в сети имеет свой уникальный *IP-адрес* (произносится как "ай-пи адрес"). Каждый адрес – это 32-битное двузначное число. Для удобства это число представляют четырьмя десятичными числами с точками между ними. Каждое из этих десятичных чисел может получать значение от 0 до 255.

172 . 16 . 254 . 1
 ↓ ↓ ↓ ↓
 10101100 . 00010000 . 11111110 . 00000001

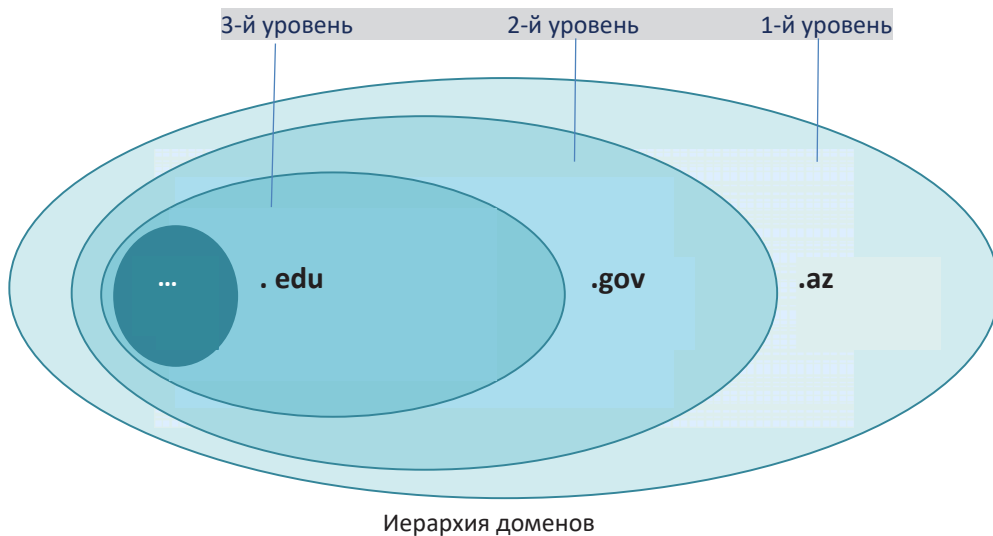
IP-адрес состоит из двух частей: *номера сети* и *номера узла*. Кроме того, адреса в сети могут выбираться администратором из адресных блоков, зарезервированных специально для таких сетей (10.0.0.0/8, 172.16.0.0/12 или 192.168.0.0/16). Если сеть будет работать в составе Интернета, то адрес сети дает провайдер или региональный реестр Интернета (Regional Internet Registry, RIR).

Понятно, что такой набор цифр запомнить нелегко. Поэтому была создана *доменная система имен* – *DNS* (Domain Name Service), которая позволяет пользователям использовать вместо IP-адресов более понятные названия (например, `www.ict.az`). То есть когда пользователь набирает в адресной строке соответствующее имя, DNS-серверы автоматически преобразовывают его в IP-адрес (например, 62.212.230.48). Затем идет обращение к веб-серверу по этому IP-адресу.



Все имена в сети разделены на *домены* разного уровня. Для каждой страны выделен домен первого уровня. Например, домен ".az" принадлежит Азербайджану, домен ".de" – Германии, домен ".ru" – России. Для так называемых территориальных доменов характерны и другие домены первого уровня. Например, домен ".com" объединяет коммерческие организации, домен ".gov" государственные организации. Доменов первого уровня мало, и новые домены появляются редко. Каждый домен первого уровня может быть

адресом для очень многих доменов более низкого уровня. Владелец каждого домена может создавать поддомены своего домена.



Каждое физическое или юридическое лицо, которое желает быть представлено в Интернете, должно уделить особое внимание выбору **доменного имени**. Доменные имена должны подчиняться определенным принципам:

- выбранное имя должно соответствовать сути сайта;
- имя, по возможности, не должно быть очень длинным;
- надо постараться, чтобы в имени домена не было букв, которых нет в английском алфавите.

Как каждый компьютер, который имеет адрес в Интернете, так и каждый ресурс – веб-страница имеет свой адрес. Этот адрес называют **URL-адресом** (произносится как “ю-эр-эл”), (аббревиатура URL образована от сокращения английской фразы **Uniform Resource Locator** и означает **единый указатель ресурсов**). Браузер использует этот адрес для поиска любого Интернет-ресурса – документа или его части.

URL-адрес однозначно определяет местонахождение ресурсов в Интернете. Каждый такой адрес состоит из двух частей: первая представляет собой *идентификатор ресурса*, вторая указывает на *местоположение* данного ресурса. Типичный URL-адрес выглядит следующим образом:

<https://ict.az/az/announce/406>

Первая часть адреса – <http://> является протоколом и показывает, что ресурс является веб-сайтом. **Протокол** – это правила или набор стандартов, обуслав-

ливающих порядок взаимосвязи и обмена информацией между компьютерами с минимальными ошибками. Для того чтобы хорошо понять оставшуюся вторую часть, разделим ее на две части. Первая часть (`ict.az`) – это доменное имя компьютера, где размещен ресурс, вторая часть (`/az/announce/406`) указывает на название конкретного веб-ресурса (веб-страницы).

Предположим, что вы создали сайт и определили для него доменное имя. А как разместить этот сайт в Интернете? Сначала вы должны зарегистрировать доменное имя. Имейте в виду, что найти соответствующее доменное имя – это только часть дела, вполне возможно, что кто-то уже зарегистрировал выбранное вами название на свое имя.

Регистрацию доменных имен проводят частные компании. Например, раздачей доменных имен “.az” занимается компания "İntrans". Там регистрируются домены второго уровня. Например, `azerbaijan.az`, `ict.az`, `oxunabiliq.az` являются такими доменами.

Размещением сайтов на любом сервере сети Интернет и оказанием им услуг также занимаются частные компании – **провайдеры**. Такие услуги называют **хостингом**. Услуги хостинга платные, но есть и компании, оказывающие бесплатные услуги.

Изучим сами

При помощи существующей системы IP-адресации (IPv4) можно дать 4 294 967 296 (256×256×256×256) компьютерам неповторимые адреса. Почему возникла необходимость перехода к 6-й версии IP-адресации? Проведите исследование и подготовьте презентацию о значении системы IPv6-адресации.

Проверьте себя

1. В каком формате представлены адреса компьютеров в сетях?
2. Какой IP-адрес у официального сайта Министерства образования?
3. Что показывает URL-адрес?
4. В чем заключается услуга хостинга?
5. Представьте 32-битный IP-адрес 1100110110100010000101001111010 в виде четырех десятичных чисел, разделенных точками.
6. Укажите IP-адрес 145.120.30.56 в 32-битной форме, состоящий из 4-х десятичных чисел.

ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

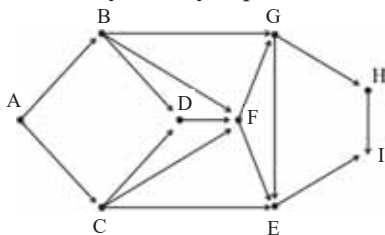
1. Для чего нужен IP-адрес?
2. Принадлежность к каким группам и структурам доменных имен указывает домен ". gov"?
3. Как называется топология сети, в которой все компьютеры сети подключены к одному центральному устройству?
4. Ветвь (подсеть) компьютерной сети образует связный неориентированный граф, и этот граф задается в виде списка смежности. Определить топологию этой сети.

Вершина	Список смежности
1	2, 5
2	1, 3
3	2, 4
4	3, 5
5	4, 6
6	5, 1

5. В таблице указаны расстояния (в километрах) между населенными пунктами А, В, С, D и Е. Определите длину кратчайшего пути между пунктами В и Е?

	А	В	С	Д	Е
А	–	13	3	–	9
В	13	–	11	11	–
С	3	11	–	9	7
Д	–	11	9	–	2
Е	9	–	7	2	–

6. Дана схема дорог, соединяющих города А, В, С, D, Е, F, G, H, I. По каждой дороге можно двигаться в одном направлении, указанном стрелкой. Сколько существует различных путей от А до I?



Литература

1. R.Mahmudzadə, İ.Calallı. İNFORMATİKA. Qəbul imtahanlarına hazırlaşanlar, yuxarı sinif şagirdləri və müəllimlər üçün ƏL KİTABI, 2 hissədə – Bakı, "Bakı" nəşriyyatı, 2022.
2. İ.Calallı. İnformatika terminlərinin izahlı lüğəti, Bakı, "Bakı" nəşriyyatı, 2017
3. C.Vorderman, C.Steele, C.Quigley, D.McCafferty, M.Goodfellow. Coding Games in Python – New York, DK Publishing, 2018
4. Computer Literacy BASICS: A Comprehensive Guide to IC3 – Boston, "Thomson Course Technology", 2005
5. Charles Petzold. Code: The Hidden Language of Computer Hardware and Software – Microsoft Press, 2000.
6. Anany Levitin. Introduction to the design & analysis of algorithms – Pearson, 2012
7. Programming BASICS Using Microsoft Visual Basic, C++, HTML, and Java – Boston, "Thomson Course Technology", 2002
8. Энциклопедия школьной информатики – Москва, "БИНОМ. Лаборатория знаний", 2011
9. Проблемы школьного учебника: XX век: Итоги – Москва, "Просвещение", 2004
10. Таненбаум Э., Уэзеролл Д. Компьютерные сети – СПб, "Питер", 2012
11. Пол Мак-Федрис. Microsoft Windows 7. Полное руководство – Москва, "Вильямс", 2012

BURAXILIŞ MƏLUMATI

İNFORMATİKA – 9

Ümumi təhsil müəssisələrinin 9-cu sinifləri üçün informatika fənni üzrə dərslik
Rus dilində

Tərtibçi heyət:

Müəlliflər	Ramin Əli Nazim oğlu Mahmudzadə İsmayıl Calal oğlu Sadıqov Naidə Rizvan qızı İsayeva
Nəşriyyat redaktoru	Kəmalə Abbasova
Bədii redaktor	Taleh Məlikov
Texniki redaktor	Zeynal İsayev
Dizayner	Pərviz Məmmədov
Rəssam	Elmir Məmmədov
Korrektor	Aqşin Məsimov

© Azərbaycan Respublikası Elm və Təhsil Nazirliyinin qrif nömrəsi: 2024-057

Müəlliflik hüquqları qorunur. Xüsusi icazə olmadan bu nəşri
və yaxud onun hər hansı hissəsini yenidən çap etdirmək, surətini çıxarmaq,
elektron informasiya vasitələri ilə yaymaq qanuna ziddir.

Hesab-nəşriyyat həcmi 7,4. Fiziki çap vərəqi 8,5.
Şriftin adı və ölçüsü: Times New Roman qarnituru, 10-11 pt. Səhifə sayı 136.
Kağız formatı 70 × 1001/16. Kəsimdən sonra ölçüsü 16,5 × 24,0
Sifariş __. Tiraj 7880. Pulsuz. Bakı – 2024

Əlyazmanın yığma verildiyi və çapa imzalandığı tarix: 07.05.2024

Çap məhsulunu hazırlayan:
“**Bakınəşr**” (Bakı, H.Seyidbəyli k., 30)

Çap məhsulunu istehsal edən:
“İndiqo” mətbəəsi (Şəfayət Mehdiyev, 133)

Pulsuz



Əziz məktəbli !

Bu dərslik sizə Azərbaycan dövləti tərəfindən bir dərs ilində istifadə üçün verilir. O, dərs ili müddətində nəzərdə tutulmuş bilikləri qazanmaq üçün sizə etibarlı dost və yardımçı olacaq.

İnanırıq ki, siz də bu dərsliyə məhəbbətlə yanaşacaq, onu zədələnmələrdən qoruyacaq, təmiz və səliqəli saxlayacaqsınız ki, növbəti dərs ilində digər məktəbli yoldaşınız ondan sizin kimi rahat istifadə edə bilsin.

Sizə təhsildə uğurlar arzulayırıq!

