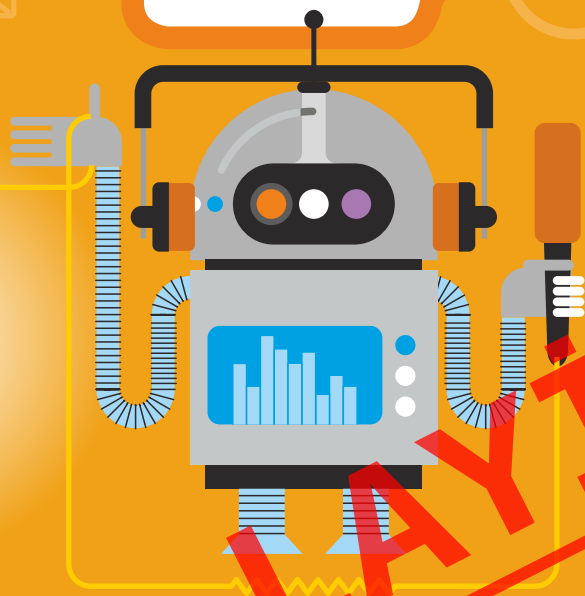


ИНФОРМАТИКА

9



1001



ЛАЙНЭ



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 qadیرiz!
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!

LAYİHƏ



ГЕЙДАР АЛИЕВ
ОБЩЕНАЦИОНАЛЬНЫЙ ЛИДЕР
АЗЕРБАЙДЖАНСКОГО НАРОДА

LA VİNƏ

LAYIHƏ

РАМИН МАХМУДЗАДЕ
ИСМАИЛ САДЫГОВ
НАИДА ИСАЕВА

УЧЕБНИК

по предмету

ИНФОРМАТИКА

для **9**-го класса общеобразовательных школ

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

В

А

К

І



Н

Ә

Ş

Р

LAYIHƏ

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. ПРИКЛАДНЫЕ ПРОГРАММЫ

10. Работа со сложными документами	44
11. Мультимедийные презентации	48
12. Работа с электронными таблицами	52
13. Диаграммы в электронной таблице	56
Обобщающие вопросы и задания	60

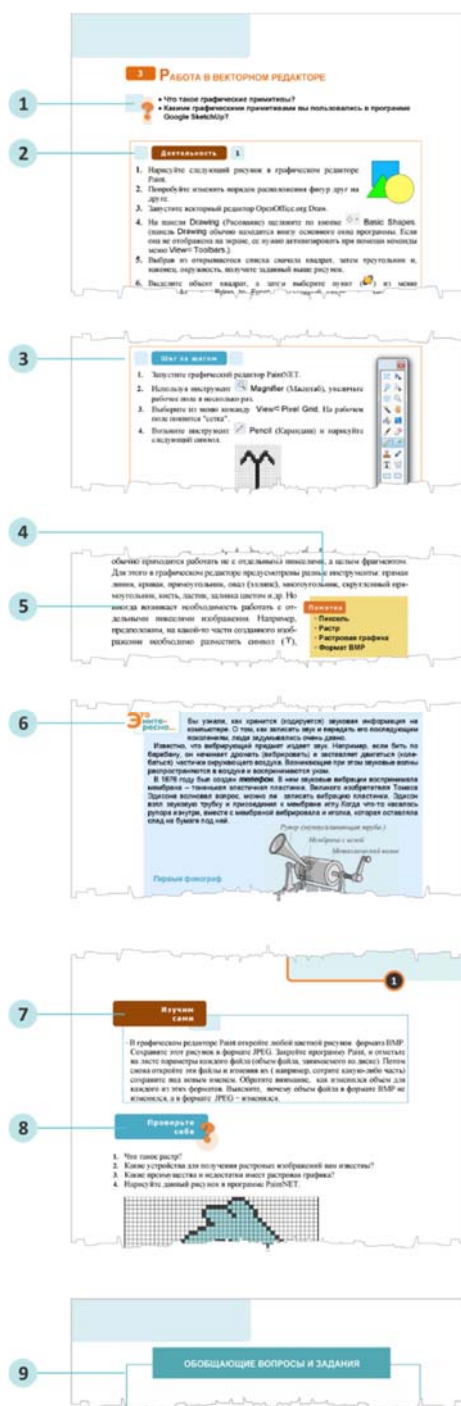
4. ПРОГРАММИРОВАНИЕ

14. Работа с числами	62
15. Строки.....	66
16. Списки.....	71
17. Функция	77
18. Классификация языков программирования.....	81
Обобщающие вопросы и задания	84

5. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

19. Информационная модель на графах	86
20. Задачи, связанные с графами.....	91
21. Компьютерные сети.....	96
22. Передача информации.....	100
23. Создание веб-сайтов.....	104
24. Разработка сайта, посвященного информационному обществу	108
25. Адресация в Интернете	113
Обобщающие вопросы и задания	116
Известные алгоритмы	117
Словарь терминов.....	124

ЛАЙНЭ



ЗНАКОМСТВО С УЧЕБНИКОМ

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



1

КОДИРОВАНИЕ

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

LAYІНӘ

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.

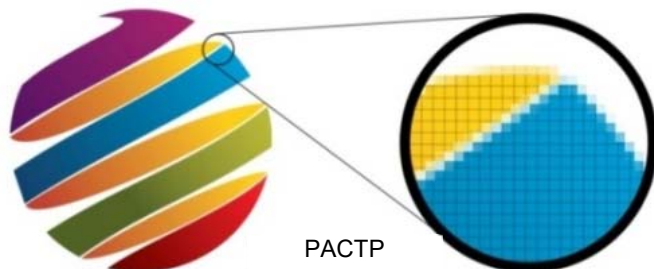


Обсудим:

- 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, формирующих 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 (сеть), 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).

Памятка

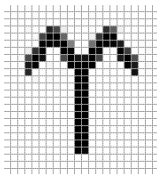
- Пиксель
- Растр
- Растровая графика
- Формат BMP

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⇒Gridlines (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 для сохранения растровых изображений в операционной системе Windows является формат *Windows Bitmap*. Файлы этого формата имеют расширение **BMP**. Для растровых изображений используют также форматы GIF и JPEG (особенно в Интернете). В последнее время получил широкое распространение формат PNG. В указанных форматах при заметном уменьшении размера файла качество изображения практически не ухудшается. Для сохранения высококачественного растрового изображения используют формат TIFF.

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

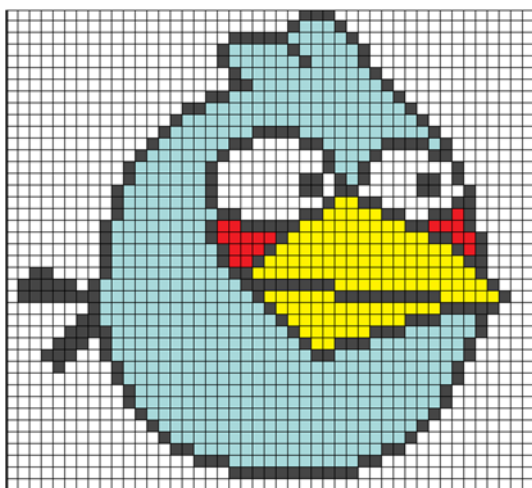
объем. Во-вторых, появляются трудности при увеличении или уменьшении растровых изображений. При увеличении масштаба видны элементы растра, и в результате изображение искажается.

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.



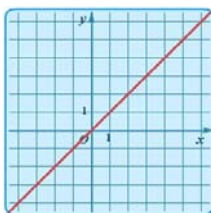


• Какие формы представления функций в математике вам известны?

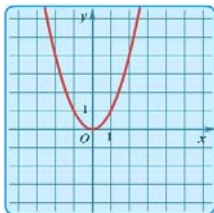
Деятельность

Определите соответствие формул представленным графикам.

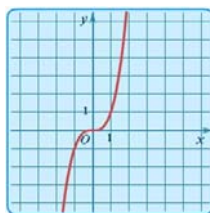
$$y = \sin x$$



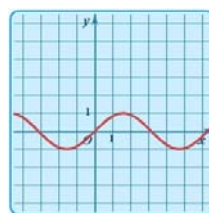
$$y = x^2$$



$$y = kx$$



$$y = x^3$$



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.

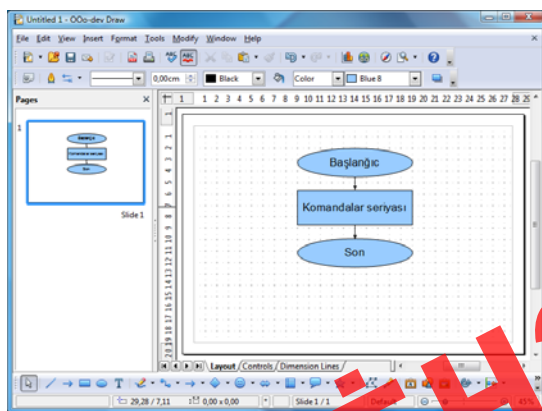
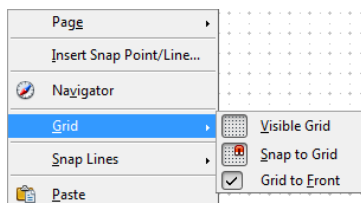
Памятка

- Векторная графика
- Формула
- Инструмент выделения
- Формат ODG

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 (Ellipse), 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 (Rectangle) и снова эллипс (Ellipse), впишите в них, соответственно, слова **Последовательность команд** и **Конец**.
7. Щелкните по кнопке Line Ends with 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 **File⇒Save As**. В открывшемся соответствующем диалоговом окне укажите тип файла *ODF Drawing* и присвойте файлу имя **alqoritm.odg**.
10. Выберите команду меню **File⇒Export**. Укажите в открывшемся соответствующем диалоговом окне формат JPEG. Блок-схема сохранится как растровое изображение.
11. Закройте программу OpenOffice.org Draw.

Изучим сами

Перейдите в папку, где вы сохранили файлы **alqoritm.odg** и **alqoritm.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.

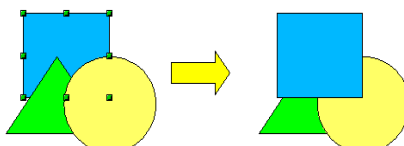
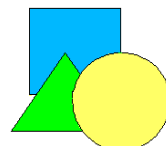
ЛАЙНЭ



- 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⇒Toolbars.).
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.

Памятка

- Слой
- Группировать
- Разгруппировать

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.

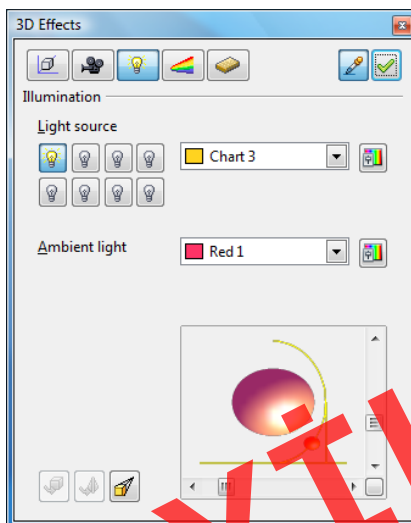
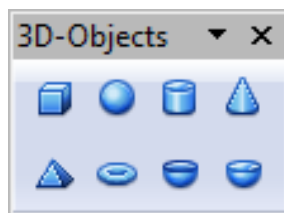
В векторном редакторе, сгруппировав (объединив) нарисованные примитивы, можно создать цельный объект. С действием **Группировать** (команда Group) вы познакомились на прошлом уроке. Действия, которые проводят над группой, относятся ко всем составляющим его объектам. Но иногда возникает необходимость 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.

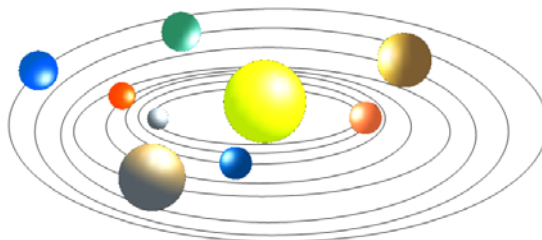
Деятельность

2

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⇒Toolbars⇒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).



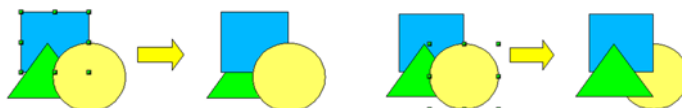
- Используя эти возможности редактора, создайте трехмерную модель Солнечной системы.



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

Изучим сами

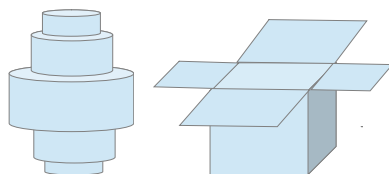
- Выясните назначение кнопок на панели Drawing в программе OpenOffice.org Draw.
- Применяя другие пункты меню **Arrange**, обратите внимание на то, как будет меняться положение выделенного объекта по отношению к другим объектам.



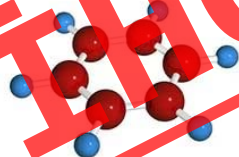
Проверьте себя



- 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.



LAYING

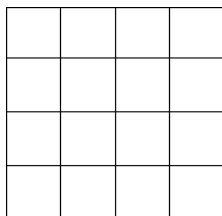
КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ



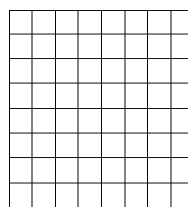
- Какие виды информации вы знаете?
- Как кодируется текстовая информация на компьютере?

1. На листе в клетку начертите прямоугольники с соответствующим числом клеток по горизонтали и вертикали.

0110
1001
1001
0110



00111100
01111110
11100111
11000011
11000011
11100111
01111110
00111100



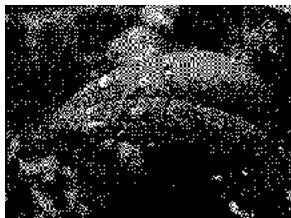
2. Найдите клетки, соответствующие "1", и закрасьте их.
3. Сравните полученные изображения.

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 (растр или вектор) 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**. Как вам известно, при помощи 1 бита можно закодировать 2 цвета (2^1), 2 бит – 4 цвета (2^2), 4 бит – 16 цветов (2^4), 8 бит – 256 цветов (2^8). В цифровом видео для глубины цвета достаточно 15 бит. Для того чтобы на фотографии цвет был естественным, необходимо как минимум 24 бита.



1 бит (2 цвета)



4 бита (16 цветов)



24 бита (16 777 216 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:

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

Пример

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.

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

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

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

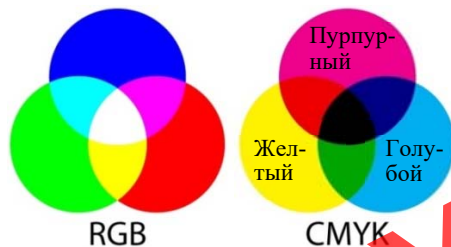
Памятка

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

На экране монитора любой цвет получается смешением **трех основных цветов**: *красного* (Red), *зеленого* (Green) и *синего* (Blue). Такая система кодирования называется **RGB** (читается как "er-ci-bi"). 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. Кодирование цветных изображений при помощи 24-битных двоичных чисел называют **полным цветным** кодированием (True Color).



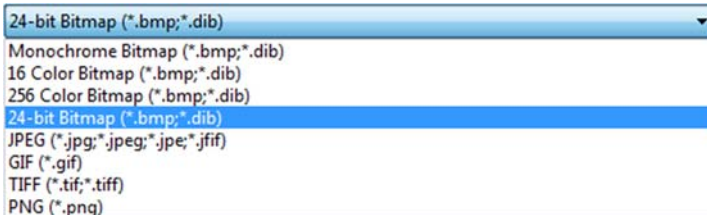
При печати приемлемо использовать не эти три основных цвета, а **дополняющие цвета**. Дополняющие цвета следующие: *голубой* (Cyan), *пурпурный*

(Magenta) и *желтый* (Yellow). Так как вещества краски несовершенны, 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 К.

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 Color 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?

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



- 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:

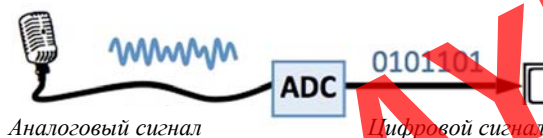
– Как эти устройства связаны друг с другом в компьютере?

Компьютер, как и большинство современных электронных устройств (фото- и видекамеры, 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*.



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**.

Памятка

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

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



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.

Первый фонограф



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**.

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 видеоинформации 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 Магомеда-гусейна Шахрияра "Heydərbabaya 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. По какой формуле вычисляется объем графической информации?
4. Что означает понятие "глубина цвета" для графических изображений?
5. Установите соответствие.

а) Растровый графический редактор	1) .bmp
б) Векторный графический редактор	2) .odg
в) Растровый формат	3) PaintNET
г) Векторный формат	4) OpenOffice Draw
6. Какое минимальное количество бит потребуется для кодирования растрового изображения, если при его создании использовано 1025 различных цветов?
7. Размер черно-белого изображения составляет 640×320 пикселей. Каков информационный объем данного изображения?
8. Какой файл займет в памяти компьютера больше места: текст из 2056 символов в кодировке Unicode или 4-цветное растровое изображение размером 10×10 пикселей?
9. Каково назначение звуковой карты компьютера?
10. Какие форматы звуковых файлов чаще используются на современных мобильных телефонах?



КОМПЬЮТЕР

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

LAYIHE

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.

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 представлении, 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 (напри-

мер, от производительности видеосистемы). Помимо этого, современные компьютеры выполняют некоторые операции (в том числе отдельные команды) *параллельно*, то есть одновременно.



Если бы процессор компьютера работал не с современной скоростью, а в привычном для человека ритме, то:

- сигналы от клавиатуры он получал бы один раз в десять лет;
- перемещение указателя мыши из одного угла экрана в другой заняло бы тысячелетие.

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.

Памятка

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



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.

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 величина меньше 70 кадров в секунду, глаз человека (и его мозг) ощущает мерцание 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 "/". Например, на рисунке указаны следующие параметры ноутбука: модель процессора (Intel Core i3-3217U), его тактовая частота

(1,7 GHz – 1,7 ГГц). Затем перечислены объем оперативной памяти, ее модель (4GB DDR3L) и объем жесткого диска (500 GB HDD).



HP 250 G3

Intel Core i3-3217U 1,7GHz, 4GB DDR3L, 500GB HDD, DVD+/-RW, Intel HD 4400, 802.11n WLAN, BT, 15.6 HD w/CAM display, Çanta, Dr.Web Antivirus, Mouse, Win 7 Pro

HP 3500 MT PC

Intel Pentium G2030, 500GB HDD 7200 SATA, DVD+/-RW, 2GB PC3-10600+ w/ HP W2072a 20-inch Widescreen LCD Monitor Dr.Web Antivirus, Win 7 Pro

Развитие аппаратного обеспечения компьютеров происходит высокими темпами, их средние характеристики ежегодно меняются. В 2015 году тактовая частота персональных компьютеров составляла 4 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 i7 2600k 3.4/Asus P4PE-2x/16 GB DDR3/1,5 TB HDD/Video Asus Radeon AX600 Pro 128 MB TV-out/DVD+RW NEC/ATX



- 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.

Памятка

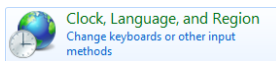
- Панель управления
- Раскладка клавиатуры

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...

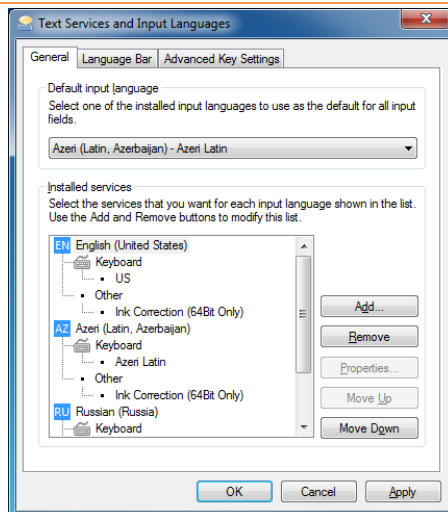
(Изменить клавиатуру).

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.

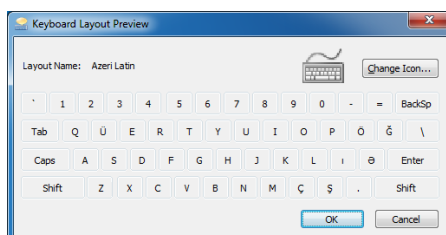
5. В разделе **Installed services** (Установленные службы) щелкните по кнопке **Add** (Добавить).

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?

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.

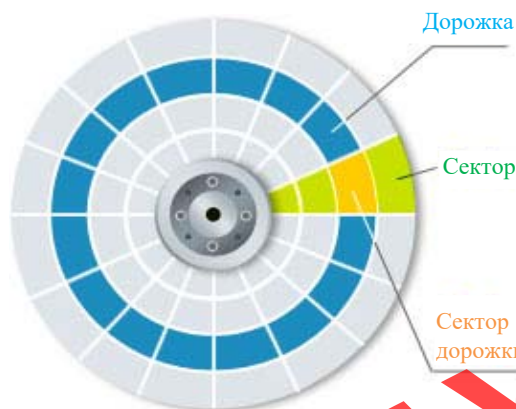
Функции служебных программ тесно связаны с возможностями операционной системы, поэтому большинство этих программ входит в ее состав. Но есть и самостоятельные служебные программы.

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*.

Памятка

- Служебные программы
- Сектор
- Дорожка
- Фрагментация
- Дефрагментация



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.

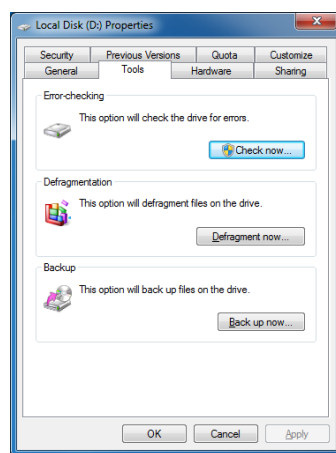
Деятельность

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?



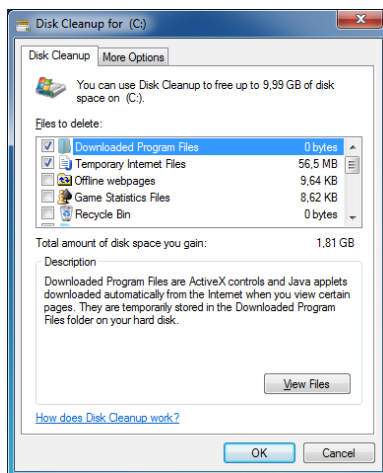
- Имеются ли на вашем персональном компьютере ненужные вам программы или файлы? Что вы делаете, обнаружив их?
- Что означает зависание программы или компьютера? Что вы делаете в такой ситуации?

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. Выберите команду All Programs⇒Accessories⇒System Tools⇒Disk Cleanup. Откроется соответствующее диалоговое окно.



3. Если в системе больше одного диска, выберите диск, который следует очистить, в ином случае перейдите к следующему пункту. Для выбора из списка нужного диска используйте кнопку со стрелкой.
4. Щелкните по кнопке ОК. Программа подсчитает и укажет, сколько места на диске она сможет очистить.
5. Отметьте файлы, которые хотите удалить, и щелкните по кнопке ОК.

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

Памятка

- **Disk Cleanup**
- **Task Manager**

Программы, установленные когда-то на вашем компьютере, через определенное время могут оказаться ненужными, и вы захотите их удалить.

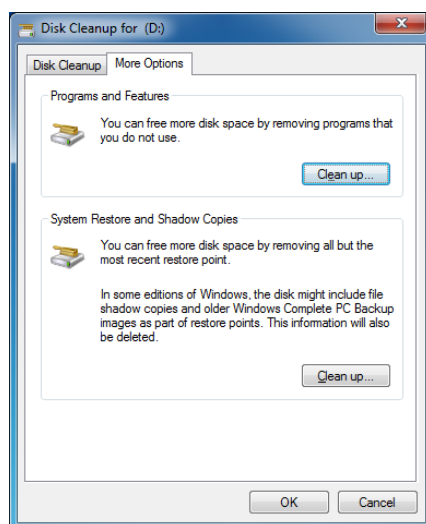
Внимание !!! Не удаляйте программы, назначение которых точно не знаете!

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**.

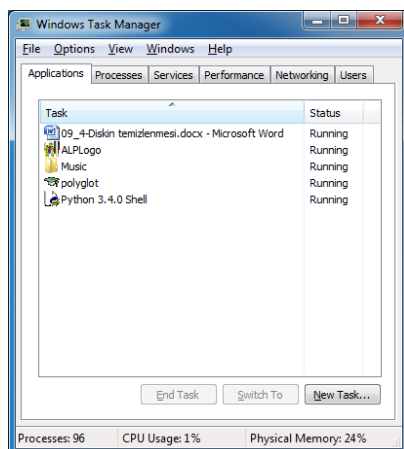
Иногда программа, с которой вы работаете, зависает, то есть перестает отвечать на ваши команды. Нажатие разных клавиш клавиатуры не дает никаких результатов. Операционная система пытается сама найти и решить проблему. Но если у вас нет времени ждать, то при помощи служебной программы **Task Manager** 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	2% CPU	19% Memory	0% Disk	0% 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. Какую роль играет Панель управления в операционной системе Windows? Может ли компьютер не иметь Панели управления? Объясните свой ответ.
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. Для чего предназначена программа Task Manager?
6. К какому виду программного обеспечения относятся программы Task Manager и Disk Cleanup?
7. Если у вас дома имеется компьютер, проведите анализ фрагментации одного из его дисков. Определите полный объем диска, свободный объем памяти, процент свободного места, размер кластера.
8. Какие утверждения неверны?
 - a) Частота тактов процессора измеряется в герцах.
 - b) Кэш-память находится на жестком диске компьютера.
 - c) Объем оперативной памяти больше объема жесткого диска.
 - d) Служебные программы относятся к системным программам компьютера
 - e) С помощью программы Task Manager можно остановить работу программы.
 - f) Программа ScanDisk очищает компьютер от ненужных файлов.
9. Сравните параметры двух компьютеров. Параметры какого из них предпочтительнее? Поясните свой ответ.

Notebook X1/1920×1080/Intel® Core™ i7/2.5 GHz /RAM 16Gb/ HDD 256 Gb/Veb-kamera 1.3 Mpiksel/Windows 8.1 (64-bit)/25.4×38.1×3.3 sm/2.7 kg

Notebook X2/1366×768/Intel® Celeron®/2.16 GHz/ RAM 4Gb/HDD 1500Gb/Veb-kamera 0.3 Mpiksel/Windows 8.1 (64-bit)/38.1×25.9×2.54 sm/2.6 kg



ПРИКЛАДНЫЕ ПРОГРАММЫ

- 10. Работа со сложными документами
- 11. Мультимедийные презентации
- 12. Работа с электронными таблицами
- 13. Диаграммы в электронной таблице

LAYІНӘ



- Что называется текстовым документом?
- Какие объекты могут быть в текстовом документе?

Шаг за шагом

1

*Количество теплоты, переданное для плавления кристаллического вещества, взятого при температуре плавления, называется **теплотой плавления**.*

Теплота плавления зависит от массы и вида кристалла:

$$Q = \lambda t.$$

Здесь Q — теплота плавления, t — масса тела, λ — **удельная теплота плавления**. Удельная теплота плавления вещества — это физическая величина, используемая для определения разницы затрачиваемой теплоты во время полного плавления твердых тел разных видов массой 1 кг при температуре плавления.

Формула	Единица измерения
$\lambda = \frac{Q}{t}$	$[\lambda] = 1 \frac{\text{Дж}}{\text{кг}}$
Определение	Определение
Удельная теплота плавления вещества — это физическая величина, равная количеству теплоты, необходимого для плавления 1 кг кристаллического вещества, взятого при температуре плавления.	1 джоуль на килограмм — это такая удельная теплота плавления, когда для плавления 1 кг кристаллического вещества затрачивается 1 Дж количества теплоты.

Подготовьте такой документ в текстовом редакторе (Microsoft Word или OpenOffice.org Writer). Для этого выполните следующий алгоритм:

1. Запустите программу Microsoft Word.
2. Введите и отформатируйте следующий текст.

*Количество теплоты, переданное для плавления кристаллического вещества, взятого при температуре плавления, называется **теплотой плавления**.*

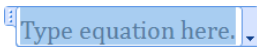
Теплота плавления зависит от массы и вида кристалла:

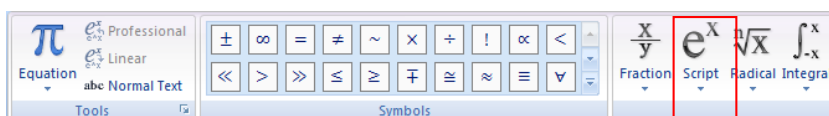
3. Перейдите на новую строку и введите формулу $Q = \lambda m$. Чтобы ввести букву греческого алфавита λ (лямбда) выберите из списка шрифтов **Symbol** и нажмите на клавиатуре букву <L> (в английской раскладке клавиатуры). С помощью кнопки  расположите формулу по центру строки.

4. Введите следующий абзац предложенного документа.

5. Для ввода таблицы используйте в меню **Insert** кнопку . Постройте таблицу, состоящую из двух столбцов и четырех строк.

6. Заполните ячейки таблицы так, как показано на рисунке.

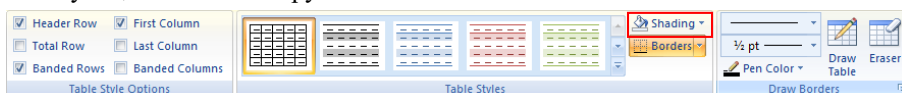
7. Для ввода формулы выберите меню **Insert** и щелкните по кнопке . Откроется панель инструментов **Equation tools**, и в документе появится поле  для ввода формулы.



8. Для ввода дробей щелкните по кнопке **Fraction** в панели инструментов **Equation tools**. Выберите из списка вариант для простой дроби. Поместив курсор соответственно в числитель и знаменатель дроби, введите нужные символы формулы.

9. Расположите записи в ячейках таблицы соответственно образцу: записи первых трех строк выровните по центру, записи последней строки выровните по левому краю. Отформатируйте соответствующие записи, выделив их жирным шрифтом.

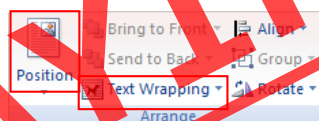
10. Измените цвет ячеек первой и третьей строк таблицы. Для этого щелкните по кнопке **Table Tools**, появившейся в строке заголовка окна. Откроется соответствующая панель инструментов.



11. Выберите из списка **Shading** соответствующий цвет. Выбранный цвет заполнит активную ячейку (на которой находится курсор) таблицы.

12. Сохраните документ, задав файлу соответствующее имя (например, **Теплота плавления**).

Иногда размещение рисунка в тексте может вас не устраивать. Например, вы хотите, чтобы рисунок был размещен не между двумя абзацами, а обтекался текстом или находился в правом верхнем углу страницы. Наверняка в учебниках или других печатных изданиях вы встречались с таким размещением. Для расположения изображений в нужном месте документа в текстовом процессоре Word предусмотрены два



инструмента Position и Text Wrapping на панели инструментов для рисования (Picture Tools).



Для отображения панели инструментов для рисования необходимо произвести двойной щелчок на графическом изображении.

В текстовом редакторе имеется возможность добавлять в документ декоративные тексты с особыми текстовыми эффектами.



Для этого предусмотрен инструмент WordArt. С его помощью можно повернуть текст, наклонить его, придать ему тень и применить другие эффекты. В меню Insert, щелкнув по этой кнопке, можно выбрать в открывшемся окне любой понравившийся шаблон.

Шаг за шагом

2

Для подготовки данного образца выполните заданные шаги.

Наземные млекопитающие

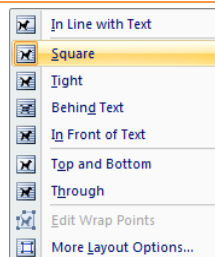
Наземные млекопитающие, в основном, населяют леса и открытые пространства. У них пропорционально сложенное сильное тело, хорошо развитые длинные конечности, мускулистая шея. Передвигаются они ходьбой, бегом и прыганьем.

Большую часть наземных зверей составляют растительноядные животные, например представители парнокопытных – олени, антилопы, жирафы, козлы, овцы, коровы и др. К ним относятся и представители непарнокопытных (лошадь, зебра, осел) и зайцеобразные.



1. Создайте новый файл в текстовом редакторе.
2. В меню Insert щелкните по кнопке WordArt и выберите в открывшемся окне любой шаблон.
3. В открывшемся окне впишите название текста (**Наземные млекопитающие**) в поле Text и щелкните по кнопке ОК.
4. С новой строки наберите следующий текст:
Наземные млекопитающие, в основном, населяют леса и открытые пространства. У них пропорционально сложенное сильное тело, хорошо развитые длинные конечности, мускулистая шея. Передвигаются они ходьбой, бегом и прыганьем.
Большую часть наземных зверей составляют растительноядные животные, например, представители парнокопытных – олени, антилопы, жирафы, козлы, овцы, коровы и др. К ним относятся и представители непарнокопытных (лошадь, зебра, осел) и зайцеобразные.
5. Найдите рисунок, соответствующий тексту (например, из Интернета) и вставьте его в документ.

6. Переместите указатель мыши на рисунок и щелкните правой кнопкой.
7. Из открывшегося контекстного меню выберите строку **Text Wrapping**, а из его подменю – пункт **Square**.
8. Переместите рисунок в нужное место документа.
9. Сохраните файл под соответствующим именем.



Изучим сами

Ознакомьтесь в учебнике с заданием, размещенным в блоке "Деятельность" темы "Передача информации" (стр.100). Создайте в текстовом редакторе одностраничный документ по этой теме. Вставьте в него заданные рисунки.

Для того чтобы воспользоваться данными рисунками, скачайте из Интернета (с сайта derslik.edu.az или informatik.az) электронную версию этого учебника, либо учебника информатики для 6-го класса. Найдите страницы, на которых находятся эти рисунки, сохраните их любым способом и расположите каждый рисунок в соответствующем месте текста.

Проверьте себя

1. Как разместить рисунки в разных частях текста?
2. В каких случаях используют инструмент WordArt?
3. Используя возможности текстового редактора WordArt, подготовьте один из документов – Грамоту, Диплом или Сертификат.
4. Подготовьте в текстовом редакторе данный документ и выполните задание, взятое из учебника математики.

Записав произведение в виде степени, заполните таблицу:

$(-3,2x)^2 \cdot (-3,2x)^4 =$	$3^7 \cdot 3^5 \cdot 3^2 =$	$(-0,6)^4 \cdot (-0,6) =$
$(a-b)^5 \cdot (a-b)^8 =$	$7^2 \cdot 7^4 \cdot 7 \cdot 7^6 =$	$16^2 \cdot 16 \cdot 16 \cdot 16^5 =$
$(x+2y)^9 \cdot (x+2y)^{10} =$	$12^4 \cdot 12 \cdot 12^9 =$	$2,3 \cdot 2,3^8 \cdot 2,3^6 \cdot 2,3 =$
$\left(\frac{3}{4}x\right)^{11} \cdot \left(\frac{3}{4}x\right)^8 \cdot \left(\frac{3}{4}x\right)^9 =$	$x^{11} \cdot x^8 \cdot x \cdot x^3 =$	$(1,(5))^3 \cdot \left(\frac{14}{9}\right)^8 =$



- Что такое мультимедиа?
- Как добавляют звук и видео в презентацию, созданную в программе OpenOffice Impress?

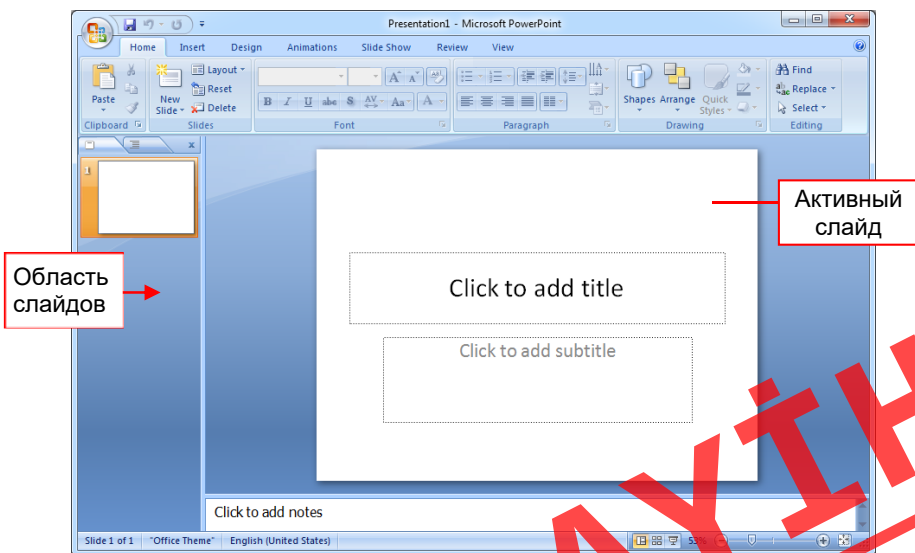
В младших классах вы научились создавать электронные презентации, размещать на слайдах текст, графику, рисунки, звуковые и видеофайлы, таблицы, диаграммы и другие объекты, добавлять к объектам на слайдах эффекты анимации. Эту работу вы выполняли в основном в программе **OpenOffice Impress**. А теперь вы закрепите свои умения в другой широко распространенной презентационной программе **PowerPoint**, входящей в пакет Microsoft Office.

Шаг за шагом

Создание мультимедийной презентации в программе Microsoft PowerPoint

Перед началом создания презентации выполните следующее: создайте на компьютере новую папку "**Поэты Азербайджана**"; найдите в Интернете и скачайте изображения с портретами пяти известных поэтов Азербайджана, небольшие отрывки их стихов и сохраните все в соответствующей папке. Найдите в Интернете или на диске с видеофайлами видеоролик об одном из поэтов и сохраните в этой же папке.

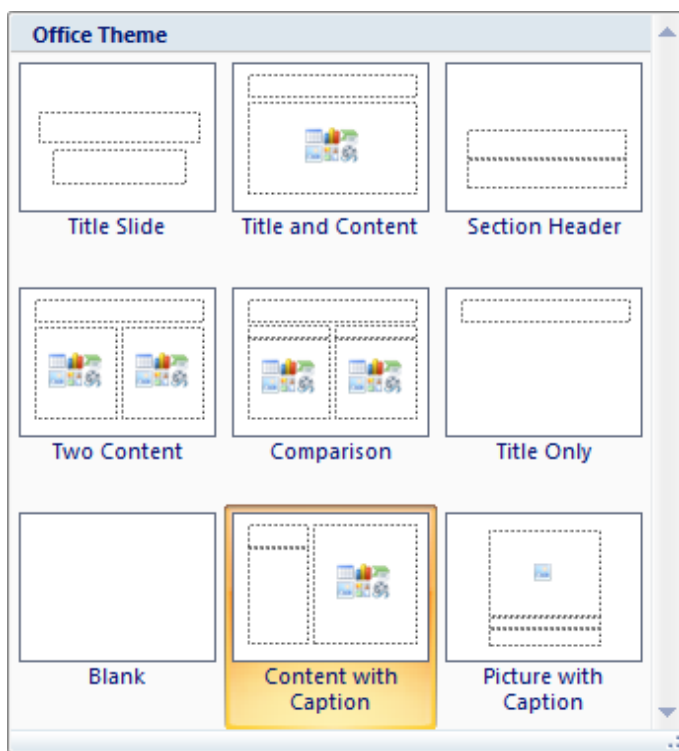
1. Запустите программу Microsoft PowerPoint. Откроется главное окно программы, а вместе с ним новый файл, состоящий из одного слайда. Как правило, в середине окна размещается **активный слайд**. Все слайды (и этот, и новые, которые вы будете создавать потом) можно увидеть в левой части окна в **области слайдов**.



2. Используя клавишу <Enter> или воспользовавшись кнопкой  на панели инструментов в блоке Slides, создайте несколько (пять-шесть) новых слайдов.
3. Активируйте первый слайд. Обычно этот слайд является титульным. На нем размещается название презентации, информация об ее авторе. Укажите в поле Click to add title заголовок презентации (**Поэты Азербайджана**). Если вы хотите, чтобы название было более привлекательным, воспользуйтесь инструментом WordArt. К примеру, заголовок может получиться такой:

Поэты Азербайджана

4. В поле Click to add subtitle разместите информацию о себе.
5. Активируйте второй слайд и выберите для него другой макет. Для этого на панели инструментов в блоке Slides щелкните по кнопке . Будут отображены различные макеты слайдов.



6. Выберите макет  Content with Caption. Обычно этот макет используют для размещения на одном слайде текста и рисунка.
7. В поле Click to add title укажите название слайда – имя одного из поэтов. Запишите в поле Click to add text один куплет стихотворения этого поэта.

8. Чтобы добавить на слайд рисунок с портретом поэта, следует воспользоваться

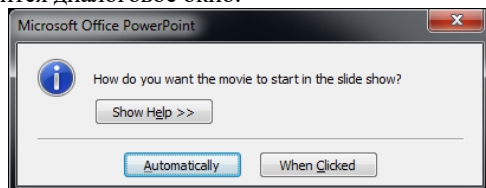


группой кнопок и выбрать из них кнопку . Перейдите в папку с сохраненными графическими изображениями (Поэты Азербайджана) и выберите соответствующий рисунок.

9. Активизируйте третий слайд. Выберите для него макет, аналогичный второму слайду. Этот слайд посвятите поэту, о котором имеется видео. Введите в название слайда имя этого поэта, а в поле, предусмотренное для стихотворения, поместите

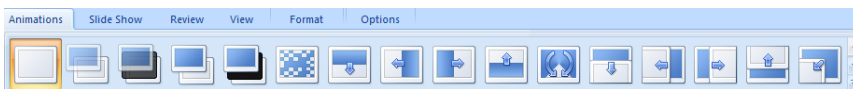


соответствующее видео. Для этого в меню **Insert** щелкните по кнопке **Movie**. Найдите в соответствующей папке необходимый видеофайл и добавьте на слайд. На экране появится диалоговое окно.



Если вы хотите, чтобы видео при открытии слайда запускалось автоматически, выберите вариант **Automatically**. Если вы выберете вариант **When Clicked**, для запуска видео необходимо будет воспользоваться левой кнопкой мыши. Обычно во время демонстрации презентации предпочтение отдают второму варианту. На слайде будет отображен кадр из видеофайла в виде рисунка. Размеры рисунка и его место на слайде отрегулируйте по своему желанию.

10. Аналогично заполните остальные слайды.
11. Для того чтобы применить эффект перехода от одного слайда к другому, выберите меню **Animations**. Пройдитесь указателем мыши по предлагаемым эффектам и, чтобы заранее посмотреть, каким будет эффект, немного подождите. Выберите понравившийся эффект перехода, и если желаете применить его ко всем слайдам, щелкните на этой же панели по кнопке **Apply To All**.



12. В меню **Insert** щелкните по кнопке  **Header & Footer**. В открывшемся диалоговом окне отметьте запись **Slide Number** и щелкните по кнопке **Apply to All**. Слайды в презентации пронумеруются последовательно, начиная с первого.

13. Перейдите в меню **Slide Show** и щелкните по кнопке  **From Beginning**. Начнется демонстрация подготовленной презентации с первого слайда.

14. Для того чтобы завершить демонстрацию слайдов и перейти в обычный режим, нажмите клавишу <Esc>.

15. Задайте файлу соответствующее имя и сохраните его в папке "Поэты Азербайджана".

После создания презентации или во время работы над ней иногда возникает необходимость в смене порядка следования слайдов. Работа над слайдами ведется, как правило, в режиме обычного просмотра (Normal View). Но для полного представления о презентации этого режима недостаточно. Для этого можно воспользоваться режимом сортировки слайдов (Slide Sorter View). Для того чтобы перейти в этот режим, в нижней части окна программы из кнопок на панели статуса  следует выбрать кнопку Slide

Памятка

- Область слайдов
- Активный слайд

Sorter View или в меню View щелкнуть по кнопке  Slide Sorter. В этом режиме можно с легкостью менять местами слайды, добавлять новые слайды и удалять некоторые из них.

- **Изменение порядка следования слайдов.** Для изменения места слайда в презентации установите на нем указатель мыши и, удерживая левую кнопку, переместите указатель в новое место. Затем отпустите кнопку мыши.
- **Удаление слайда.** Для удаления слайда выделите его и нажмите клавишу <Delete> или <Backspace>. Для удаления слайда можно использовать также кнопку  на панели инструментов.
- **Добавление нового слайда.** Для добавления в презентацию нового слайда сначала выделите слайд, после которого будет вставлен новый, и щелкните по кнопке **New Slide**. Для редактирования нового слайда перейдите в обычный режим.

Изучим сами

Для того чтобы сделать презентацию более привлекательной и передавать аудитории информацию по частям и в определенной последовательности, для некоторых объектов на слайде применяют эффекты анимации. В прошлом году вы научились делать это в программе OpenOffice Impress. Естественно, аналогичную работу можно проделать и в программе Microsoft PowerPoint. Щелкните в меню **Animations** по кнопке  и изучите одноименную панель, которая появится в правой части окна программы. Примените разные эффекты к объектам на слайдах созданной вами презентации **Поэты Азербайджана**.

Проверьте себя

1. Что подразумевается под мультимедийными объектами в презентации?
2. Как в презентацию добавить звук и видеофайл?
3. Создайте презентацию на любую тему. Примените к слайдам презентации эффекты Motion Paths.

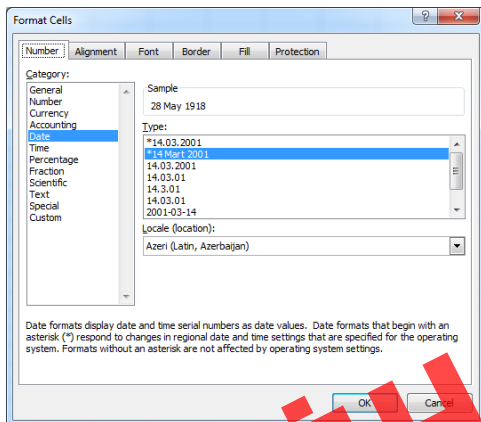


- Для каких целей используют электронные таблицы?
- Как записывается формула в электронной таблице?

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 43248 и установить формат ячейки как текстовый, то содержимое ячейки будет воспринято как последовательность символов 4, 3, 2, 4, 8. 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 2018 года (число дней с 01.01.1900 до 28.05.2018 составляет 43248).

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

Назначение электронной таблицы в первую очередь состоит в автоматизации вычислений над данными. Для этого в ячейки таблицы вводятся **формулы**. В формулы могут входить числовые данные, знаки операций, различные функции, а также адреса объектов таблицы. Формулы с адресами ячеек можно сравнить с математическими уравнениями: в уравнениях вместо адресов ячеек используют переменные.

В записи формулы адрес ячейки или диапазона ячеек указывается при помощи **ссылки**. Ссылки позволяют связывать любые ячейки электронной таблицы друг с другом и проводить необходимую обработку данных таблицы.

Памятка

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

	B1	f_x	=A1*A1+23
	A	B	C
1	12	=A1*A1+23	
2	4		
3	43		
4			

Часто при решении однотипных задач используют одинаковые формулы. Для того чтобы каждый раз заново не вводить одну и ту же формулу в разные ячейки таблицы, можно создать ее копию. Существует удобный способ копирования содержимого ячейки в ячейки одной строки или одного столбца. Для этого ячейку, содержимое которой будет скопировано, следует сделать активной (появится жирная черная рамка вокруг ячейки). Потом необходимо переместить указатель мыши на небольшой квадратик в правом нижнем углу рамки (указатель примет вид $+$) и, удерживая нажатой левую кнопку мыши, протянуть указатель вдоль строки или столбца. Затем отпустить кнопку мыши. В результате формула активной ячейки будет скопирована во все выделенные ячейки. Аналогичным способом содержимое произвольной ячейки можно легко и быстро скопировать в другие ячейки.

При копировании формулы в электронной таблице из одной ячейки в другую адреса ячеек, используемые в формулах (ссылки), автоматически изменяются. Такие ссылки называют **относительными ссылками**. Относительная ссылка записывается в обычной форме, например: F3, E7.

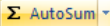
Иногда нет необходимости в изменении ссылки при копировании формулы из одного места в другое. В таких случаях в электронных таблицах используют **абсолютные ссылки**. В абсолютных ссылках перед буквой столбца и номером строки ставится символ \$, например: \$F\$3, \$E\$7.

Если при копировании формулы может изменяться только какая-то одна часть ссылки – либо буква столбца, либо номер строки, то используется **смешанная ссылка**. При этом символ \$ ставится перед той частью ссылки, которая должна остаться неизменной, например: \$F3, E\$7.

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.

	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

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.

Иногда при вычислениях используют различные стандартные функции (математические, логические, финансовые, статистические и др.). С этими функциями можно ознакомиться в диалоговом окне, если щелкнуть по кнопке  на панели формул.

Изучим сами

Ознакомьтесь с записью математических функций $|x|$ (ABS), $\sqrt{x}$ (SQRT), $\sin x$ (SIN), $\cos x$ (COS) в программе Excel. Выделите столбец А электронной таблицы для аргумента (x), а столбцы В–Е для значений соответствующих функций. Вычислите значение функций для 10 различных значений аргумента.

Проверьте себя

- Какой тип данных используется в электронных таблицах?
- Что такое абсолютная ссылка в формуле и как она обозначается?
- Создайте таблицу по данному образцу. Для этого
 - в ячейке А1 укажите текущий год;
 - запишите в соответствующие ячейки таблицы имена, фамилии, даты рождения ваших друзей;
 - 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	2017			
2				
3	Фамилия	Имя	Год рождения	Возраст
4	Аббаслы	Эльхан	2002	15
5	Ализаде	Гюнай	2001	
6	Ягублу	Рустам	2003	

- Укажите в ячейке А2 скорость автомобиля, в ячейке В2 время, которое он затрачивает на путь. Вычислите в ячейке С2 с помощью формулы путь, пройденный автомобилем ($s=vt$). Меняя значение скорости (содержимое ячейки А2) и времени (содержимое ячейки В2), проследите, как меняется пройденный путь (содержимое ячейки С2).
- Дан один фрагмент электронной таблицы. Каким будет значение в ячейках С2 и С3 при копировании в них формулы из ячейки С1?

	A	B	C
1	45	110	=A\$1+B1
2	50	23	
3	12	70	



- 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 A1, а саму таблицу разместите в диапазоне ячеек A3:J5.

Распределение численности населения по возрасту (на начало 2015 года)

	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70 и выше	Всего
Численность населения, тыс. человек	1489,1	1427,5	1866,5	1465,1	1305,9	1170,7	473,1	395,1	9593,0
Численность населения, в процентах	15,6	14,9	19,4	15,3	13,6	12,2	4,9	4,1	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?

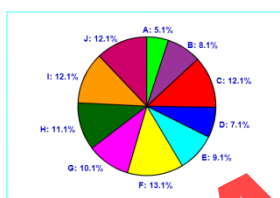
Электронные таблицы предназначены в основном для быстрой и качественной обработки однотипных числовых данных большого объема, но представление данных в графической форме делает их более наглядными и понятными. Основным средством представления табличных данных в графической форме являются диаграммы.

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

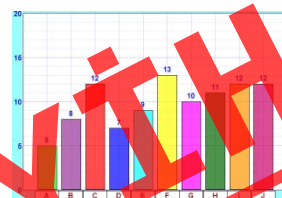
В электронных таблицах можно строить линейные, круговые, столбчатые и другие диаграммы.



Линейная диаграмма



Круговая диаграмма



Гистограмма

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.

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

В электронной таблице данные, расположенные на одной строке или в одном столбце, называют **рядом** данных. При построении диаграммы сначала необходимо выделить ряд данных, а потом выбрать тип диаграммы. Диаграмма – сложный объект, который состоит из следующих элементов: *ряда, оси, заголовка, легенды, области построения*.

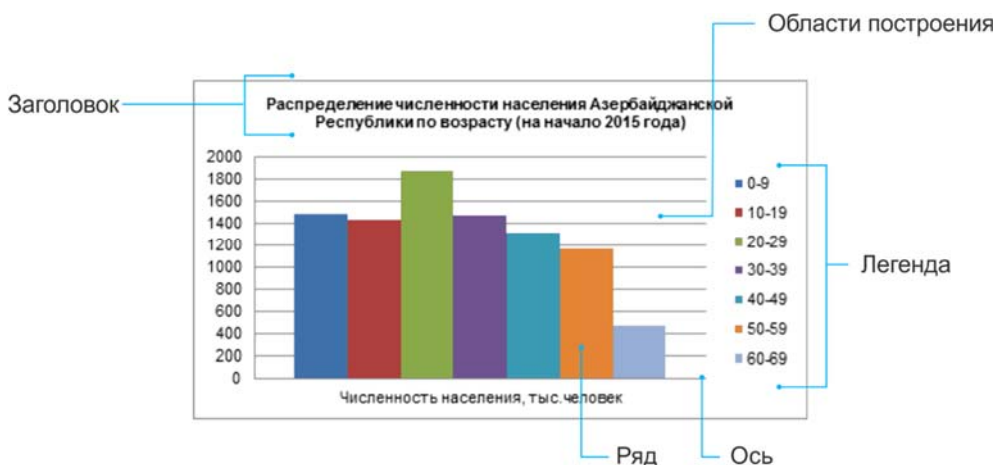


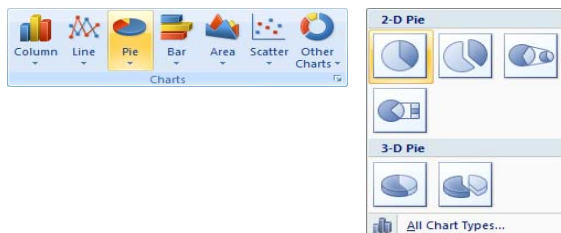
Диаграмма может быть построена как по одному ряду, так и по нескольким рядам. Для выделенного диапазона ячеек построение диаграммы ведется по нескольким рядам данных. В этом случае в качестве каждого ряда принимается соответствующая строка или столбец выделенного диапазона. **Заголовком** служит текст, 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 наноситься сетка.

Памятка

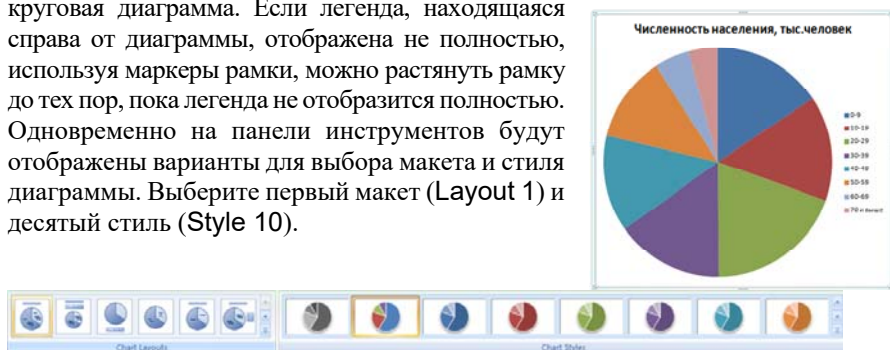
- **Диаграмма**
- **Линейная диаграмма**
- **Круговая диаграмма**
- **Гистограмма**
- **Легенда**

Шаг за шагом

1. Откройте файл электронной таблицы **"Население Азербайджана"**.
2. Для построения диаграммы на основе таблицы выделите диапазон ячеек A3:I4.
3. В меню Insert в блоке Charts щелкните по кнопке Pie. 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.
5. Одновременно на панели инструментов будут отображены варианты для выбора макета и стиля диаграммы. Выберите первый макет (Layout 1) и десятый стиль (Style 10).



6. В каждом секторе диаграммы будут указаны возраст группы и доля этого населения в процентах. Щелкните по заголовку **"Численность населения (тысяч человек)"** и введите в это поле текст **"Распределение численности населения Азербайджанской Республики по возрасту (на начало 2015 года)"**.



7. Сохраните файл под соответствующим именем.

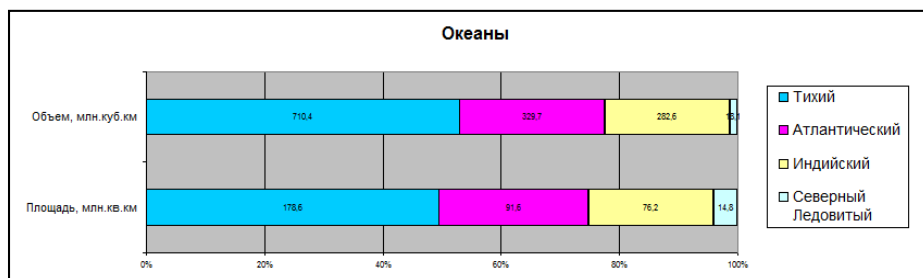
Изучим сами

Сравните процентные значения последней диаграммы со значениями в строке "Численность населения, тыс. человек", приведенными в таблице в начале урока. Попробуйте выяснить, почему округлены значения в диаграмме, и попытайтесь восстановить дробную часть чисел.

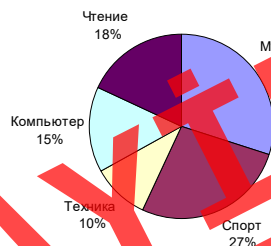
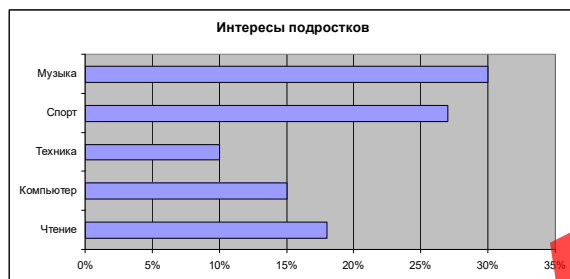
Проверьте себя

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?



ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

1. Как вы понимаете понятие "объекты текстового документа"?
2. Как поместить рисунок в различных местах относительно текста?
3. Объясните назначение данных кнопок.



4. Как в мультимедийной презентации можно задать один и тот же эффект перехода для всех слайдов?
5. В чем разница между относительным, абсолютным и смешанным адресами?
6. Что следует записать в ячейку E3 электронной таблицы, чтобы вычислить объем комнаты?

	A	B	C	D	E
1					Объем комнаты
2	Длина	7 метров			V
3	Ширина	3,5 метра			
4	Высота	3,5 метра			

7. Дан фрагмент электронной таблицы. В ячейку C1 записана формула =A1-\$B\$1. Определите значения в ячейках C2 и C3 после копирования в них формулы из ячейки C1.

	A	B	C
1	117	29	
2	98	-12	
3	45	23	

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



ПРОГРАММИРОВАНИЕ

- 14. Работа с числами
- 15. Строки
- 16. Списки
- 17. Функция
- 18. Классификация языков программирования

LAYIHE

14

РАБОТА С ЧИСЛАМИ



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

Деятельность

1

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 – это пояснения для человека, читающего текст программы. 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. При этом чем глубже уровень вхождения, тем больше отступ. Необходимо иметь в виду, что в пределах одной группы операторов отступ должен быть одинаковым.

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

Памятка

- Комментарии
- Отступ
- Арифметические операции
- Знаки сравнения

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

$$m = n // 10$$

Деятельность 2

Наберите программу в среде Python и выполните ее.

```
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!')
```

Обсудим:

- Что делает программа?
- Задавая различные числа при вводе, проверьте, правильно ли работает программа.
- Запишите операторы `print` на одном уровне с оператором `if` и проследите за изменением результата выполнения программы.

Пример Нахождение суммы и произведения цифр числа. Задано положительное целое число. Найти сумму и произведение его цифр.

Решение. Сопоставим заданному числу переменную `n`, сумме его цифр – переменную `s`, произведению его цифр – переменную `p`. Тогда алгоритм решения задачи можно описать так:

Алгоритм

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.

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

Изучим сами

Ознакомьтесь с программой и определите, для чего она предназначена. Заданная во второй строке функция `abs()` служит для нахождения *модуля числа*. Как известно из математики, модуль числа – это величина числа без знака. Например:

$$\text{abs}(4) = 4, \text{abs}(-4) = 4.$$

```
m = int(input('Введите трехзначное целое число: '))
n = abs(m)

a = (n // 100) % 10
b = (n // 10) % 10
c = n % 10
k = 100 * c + 10 * b + a

if m < 0:
    print(-k)
else:
    print(k)
```

Проверьте правильность работы программы, вводя произвольные положительные и отрицательные трехзначные числа.

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.
 - Введите произвольное двузначное число.
 - Умножьте его само на себя.
 - Сложите полученное число с произведением введенного числа и 4.
 - Прибавьте к полученному результату число 3.
 - Разделите полученный результат на число, которое на 3 больше введенного числа.
 - Вычтите выбранное вами число из полученного частного.
 - Выведите результат на экран.



- 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 = 'л'
s3 = 'п'
print (s1 + s2 + s3)
print ("a", "л", "п")
```

Обсудим:

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

Памятка

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

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:

```
>>> a = 'АЗЕРБАЙДЖАН'
>>> a[4]
'Б'
```



Гвидо ван
Россум
(1956)

Создатель
языка Python

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

```
>>> a[0:4]
'АЗЕР'
```

Как видите, несмотря на то, что вторым индексом диапазона указано число 4, 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.

```
>>> a[:5]
'АЗЕРБ'
>>> a[7:]
'ДЖАН'
```

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

```
>>> len(a)
11
```

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.
```

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 символов.
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.

Как вам известно, любой объект имеет определенные характеристики. В информатике различают два вида характеристик объекта: **свойства объектов** 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('o', 5, 10))
8
```

Пример

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

Описание переменных, используемых в программе:

`s` – вводимая строка;

`lastname, firstname` – первое и второе слово в строке; `k` – позиция пробела между словами.

Алгоритм решения задачи:

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.

```
s = input()
p = s.find(' ')
lastname = s[:p]
firstname = s[p + 1:]
s = firstname + ' ' + lastname
print(s)
```

первое слово – фамилия
второе слово – имя

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

Метод	Описание
<code>s1.count(s0)</code>	Определяет количество вclusions подстроки <code>s0</code> в строку <code>s1</code> . Например: <pre>>>> s1 = 'abrakadabra' >>> s1.count('ab') 2</pre>
<code>s1.strip()</code>	Удаляет пробелы в начале и в конце строки. Например: <pre>>>> s1 = ' abc ' >>> s1.strip() 'abc'</pre>
<code>s1.replace(s2, s3)</code>	Заменяет фрагмент строки <code>s2</code> на фрагмент <code>s3</code> . Например: <pre>>>> s1 = 'Кыргызстан' >>> s1.replace('Кыргыз', 'Казах') 'Казахстан'</pre>
<code>s1.capitalize()</code>	Преобразует первую букву в заглавную. Например: <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>

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.

LAYINƏ

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`.

Памятка

- Список
- Элементы списка
- Индекс
- Срез
- Цикл

```
>>> 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']
```

Вы знакомы с **циклами** как по программной среде ALPLogo, так и по Python. 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>	Определяет число элементов списка. Например: <pre>>>> lst = [1, 2, 3] >>> len(lst) 3</pre>
<code>lst1 + lst2</code>	Элементы списка <code>lst2</code> добавляются в конец списка <code>lst1</code>. Например: <pre>>>> lst1 = [1, 2, 3] >>> lst2 = ['один', 'два', 'три'] >>> lst1 + lst2 [1, 2, 3, 'один', 'два', 'три']</pre>
<code>min(lst)</code>	Определяет минимальный элемент списка. Например: <pre>>>> lst1 = [1, 2, 3, 85, -42, 33, 84] >>> min(lst1) -42</pre> В величинах строкового типа за основу берется порядок в алфавите. <pre>>>> lst2 = ['семь', 'восемь', 'девять'] >>> min(lst2) 'восемь'</pre>
<code>max(lst)</code>	Определяется максимальный элемент списка. Например: <pre>>>> lst = [1, 2, 3, 85, -42, 33, 84] >>> max(lst) 85</pre>
<code>del lst[i]</code>	Из списка удаляется элемент с номером <code>i</code>. Например: <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> . Например: <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> . Например: <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. Например: <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> . Например: <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. Например: <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). Например: <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. Например: <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()`.

Например:

```
>>> 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']
```

Пример

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.

Описание переменных, используемых в программе:

`s` – вводимый текст (строка);
`lst` – список, соответствующий тексту;
`i` – индекс элементов списка.

Алгоритм решения задачи:

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 'а' 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.

LAYITHE



- 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

Памятка

- Подпрограмма
- Функция
- Вызов функции
- Формальные параметры
- Фактические параметры

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 функции 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

```
def privet(s):  
    print(s)  
privet('Доброе утро!')  
privet('Добрый вечер!')
```

Формальный параметр

Фактический параметр

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**, передаваемые функции. Во время выполнения операторов функции формальные параметры заменяются фактическими значениями.

Проследим все сказанное на примере одной программы. Эта программа переводит введенный год мусульманского календаря Хиджры в год по Григорианскому (европейскому) календарю и наоборот, а также выводит результаты на экран. При этом вычисления производятся по формулам $Q = X - X / 33 + 622$ и $X = Q - 622 + (Q - 622) / 32$. Здесь переменная Q указывает год по Григорианскому календарю, X – год Хиджры.

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

```
def miladi_hicri(S):
    Q = int(S)
    X = Q - 622 + (Q - 622) // 32
    print('год Хиджры: ', X)

def hicri_miladi(S):
    X = int(S)
    Q = X - (X // 33) + 622
    print('Григорианский год: ', Q)

S = input('Григорианский год: ')
miladi_hicri(S)

S = input('год Хиджры: ')
hicri_miladi(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 вызвавшей ее программе. Для этой цели в программе Python предусмотрена команда `return`. Эта команда записывается в конце функции и после нее в скобках 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              # Значение переменной возвращается

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

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):
    Q = int(S)
    X = Q - 622 + (Q - 622) // 32
    return (X)

def hicri_miladi(S):
    X = int(S)
    Q = X - (X // 33) + 622
    return (Q)

S = input('Григорианский год: ')
X = miladi_hicri(S)
print('год Хиджры: ', X)

S = input('год Хиджры: ')
Q = hicri_miladi(S)
print('Григорианский год: ', Q)
```

Проанализируйте работу программы.

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.

LAYINƏ

КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ



- Какой язык программирования и по какой причине вам нравится больше: ALPLogo или Python?
- С какого из этих языков вы посоветуете начинать обучение программированию кому-то младше вас?

Деятельность

Даны фрагменты программ на языках программирования BASIC и Pascal. Посмотрите фрагменты и постарайтесь определить, что они делают.

BASIC

```
S = 0
FOR I = 1 TO 100
    S = S + I
NEXT I
```

Pascal

```
a := 5;
b := 10;
if a > b then write(a)
else write(b);
```

Обсудим:

- Какие операторы вам знакомы?
- Как можно записать данные фрагменты программ на языке Python?

Тот, кто начинает изучать программирование, рано или поздно задается вопросом: какой из языков программирования лучше? Несмотря на то, что имеется более тысячи языков программирования, нет единого языка, который устраивал бы всех программистов. У каждого языка программирования есть свои сторонники и противники.

Перед представителями разных сфер деятельности – инженерами, банкирами, военными – стоят разные задачи, при решении которых предпочтение отдается тому или иному языку программирования. Язык FORTRAN широко применяется при решении математических задач, язык COBOL ("кобол") – в банковском деле, языки PROLOG, LISP используются в сфере искусственного интеллекта. Программисты, создающие программы для Интернета, обычно отдают предпочтение языку JAVA.

Все перечисленные языки программирования – это языки специального назначения. В каждом из этих языков имеются такие операторы, при помощи которых можно более просто решать специфические задачи. Обычно эти языки изучают непосредственно на рабочих местах, так как заранее их осваивать не имеет смысла.

Памятка

- Языки специального назначения
- Языки общего назначения

Помимо специальных языков программирования имеются и **языки общего назначения**. С их помощью можно решать практически любую задачу. Такие языки можно изучать заранее – они всегда могут понадобиться. Среди них наиболее популярны Basic, Pascal, C++, Python.

В настоящее время в мире существует более 4000 языков программирования, многие из которых сегодня не используются. Помимо этого, наиболее важная часть программ выполняется не на языках программирования общего назначения, а с использованием специальных инструментов программирования (например, Visual Basic) или специальных типов программного обеспечения (например, Maple).

Хронограф популярных языков программирования

Название	Год	Автор	География создания
FORTRAN	1954	Джон Бэкус	США
LISP	1958	Джон Маккарти	США
Algol 60	1960	Питер Наур и др.	Различные страны
COBOL	1960	Коллектив авторов	Различные страны
Simula	1962	Кристен Нигаард и др.	Европа
BASIC	1963	Джон Кемени и др.	США
PL/I	1964	Джордж Радин	Различные страны
Algol 68	1968	Аад ван Вайнгартен и др.	Различные страны
Pascal	1970	Никлаус Вирт	Европа
FORTH	1970	Чарльз Мур	США
C	1972	Деннис Ритчи	США
Smalltalk	1972	Алан Кей	США
Prolog	1973	Алан Кольмеро и др.	Европа
Ada	1980	Джин Ишбиа и др.	США
C++	1984	Бьорн Страуструп	США
Python	1991	Гвидо ван Россум	США
Java	1995	Джеймс Гослинг	США

По назначению языки программирования можно сгруппировать приблизительно так:

1. Языки программирования общего назначения для создания больших, сложных программ: PL/I, C, C++, Pascal, Python, Modula-2, Ada, Java, C#.

2. Языки программирования общего назначения для создания небольших программ: BASIC, Visual Basic, Pascal, Python.
3. Математические вычисления, наука и инженерия: FORTRAN, APL, Maple и языки общего назначения, названия которых указаны выше.
4. Коммерческая обработка данных: COBOL, RPG.
5. Задачи искусственного интеллекта и задачи со сложной логикой: LISP и Prolog.
6. Работа со строками и сценариями: SNOBOL, REXX, Awk, Perl, Python, VBSCRIPT, JavaScript.

Изучим сами

Некоторые языки программирования созданы для решения неординарных задач. К примеру, программа, написанная на языке **Piet**, выглядит как абстрактный рисунок. Рисунок, представленный внизу, – это программа "Hello World!" ("Здравствуй, мир!"), написанная на этом языке.



Выясните, откуда появилось название этой программы. Соберите информацию о других экзотических языках программирования, таких как **Chef**, **Ook!**, **Malbolge**.

Проверьте себя

1. Какой язык программирования лучше?
2. К какому языку относится Python: языку программирования специального назначения или общего назначения?
3. Кем и когда был создан язык Python?
4. Какие языки используют, в основном, для математических вычислений?
5. Какие языки программирования можно отнести к категории специальных языков?

ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

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

```
s = input()
l = len(s)
n = int(input("Введите порядковый номер слова в предложении: "))
s = s.split(' ')
print(s[n-1])
```
9. Для каких целей используют подпрограммы?



ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- 19. Информационная модель на графах
- 20. Задачи, связанные с графами
- 21. Компьютерные сети
- 22. Передача информации
- 23. Создание веб-сайтов
- 24. Разработка сайта, посвященного
информационному обществу
- 25. Адресация в Интернете

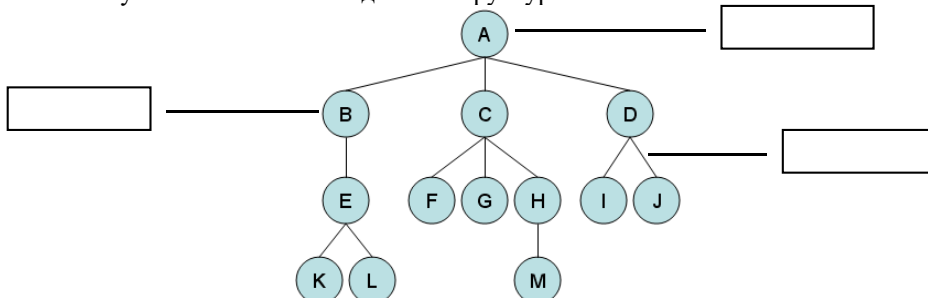
LAYIHE



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

Деятельность

Назовите указанные элементы данной структуры.



Обсудим:

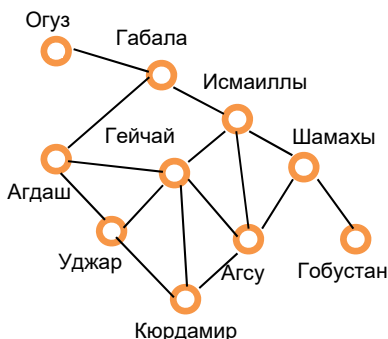
- К какому виду информационных моделей относится данная структура?
- Что такое путь и сколько путей имеется между двумя произвольными вершинами этой структуры?

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

Памятка

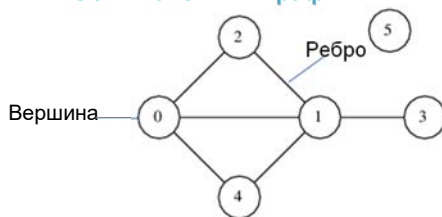
- Граф
- Вершина
- Ребро
- Список смежности
- Матрица смежности
- Связный граф
- Ориентированный граф

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

Обыкновенный граф



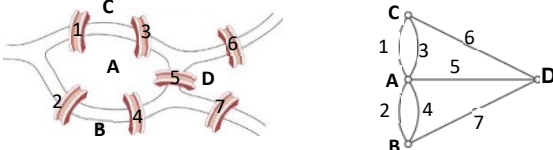
ЭТО
интересно...



Леонард
Эйлер
(1707-1783)

Поводом для создания **теории графов** стало 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 была такой: 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** перечислены вершины, соединенные с каждой из данных вершин. В этой форме 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 $a_{x,y}$ будет равен 1, в противном случае – 0. Например, 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

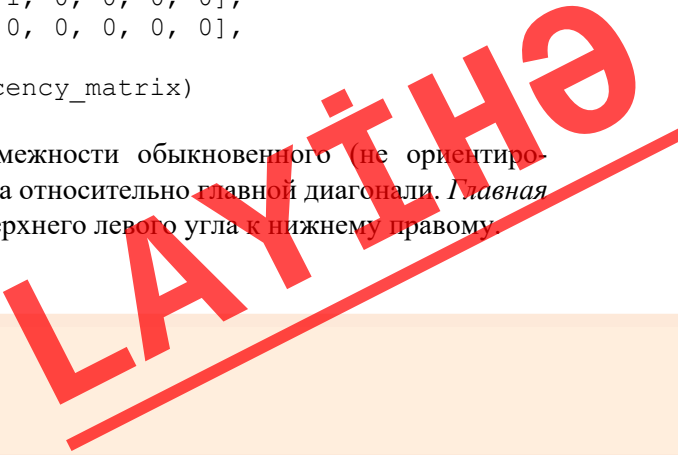
Для сохранения графов в памяти компьютера используют как эти, так и другие способы. Например, если список смежности данного графа в языке Python назовем `adjacency_list`, а количество его вершин `num_vertices`, то:

```
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.



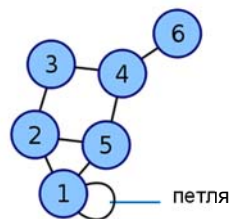
Графы можно хранить и в форме списка ребер. В этом случае каждое ребро обозначается двумя числами – номером начальной и конечной вершины.

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**.

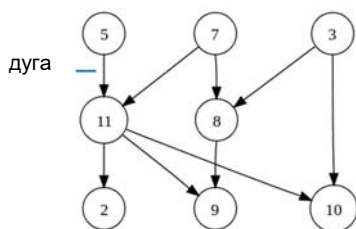
Для сохранения ориентированных графов в памяти можно использовать (немного изменив предыдущие способы) следующие способы представления:

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

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

Оrientированный граф



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. Как называют ребро, которое выходит из одной вершины и возвращается в эту же вершину?
4. В какой форме хранятся графы в памяти компьютера?
5. Постройте граф региона, в котором вы живете, и составьте на его основе соответствующую матрицу смежности.

LAYINƏ



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

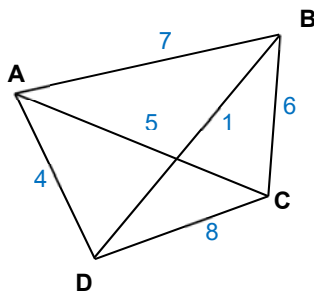
При решении некоторых практических задач (например, нахождении кратчайшего пути между пунктами) имеют значение не только связи между вершинами, но и числа, соответствующие этим связям. Этими числами могут быть, например, расстояния между городами или стоимость проезда. В теории графов числовое значение,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

Задача

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, ABDCA, ACDBA, ADBCA, ADCBA. Их длина, соответственно, равна: 25, 16, 21, 21, 16, 25.

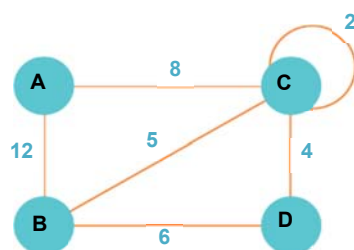
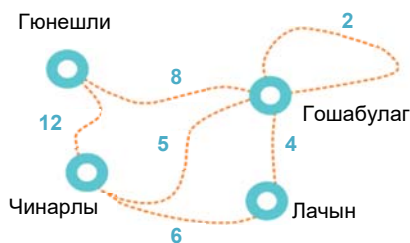
Таким образом, самыми короткими будут маршруты ACBDA и ADBCA.



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.

Памятка

- Взвешенный граф
- Весовая матрица



	A	B	C	D
A		12	8	
B	12		5	6
C	8	5		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.

Задача

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
while i <= 6:
    s = input()

    w = w + s.count('1')

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

w - количество дорог
Вводится одна строка
матрицы.
Подсчитывается число 1 в этой
строке и это число добавляется
к общему числу дорог

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. (Источник: *informatika.edu.az*)

Задача

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?

Решение. Построим граф G , 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.



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:

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

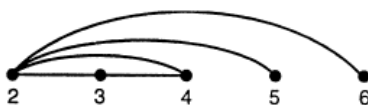
В графе H_1 вершина 2 имеет смежность со всеми вершинами, вершины 5 и 6 смежны только с двумя вершинами.

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.

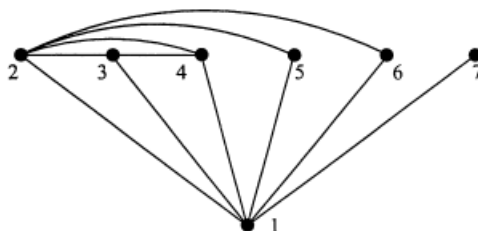
В графе H_2 $d(3) = d(4) = 1$, то есть это граф, представленный на рисунке:



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



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



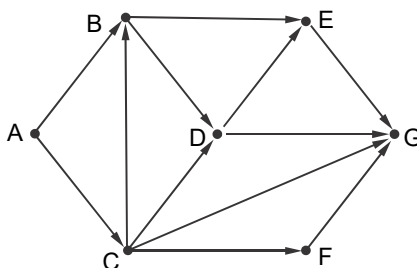
Этот граф отражает встречу школьников во время соревнований. Из этого графа видно, что Джейхун (3-я вершина графа) сыграл с Арифом, Бекиром и Дадашем, которые соответствуют вершинам 1, 2 и 4. Понятно, что при помощи этого графа нетрудно определить, с кем играли остальные участники соревнований.

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?





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

Деятельность

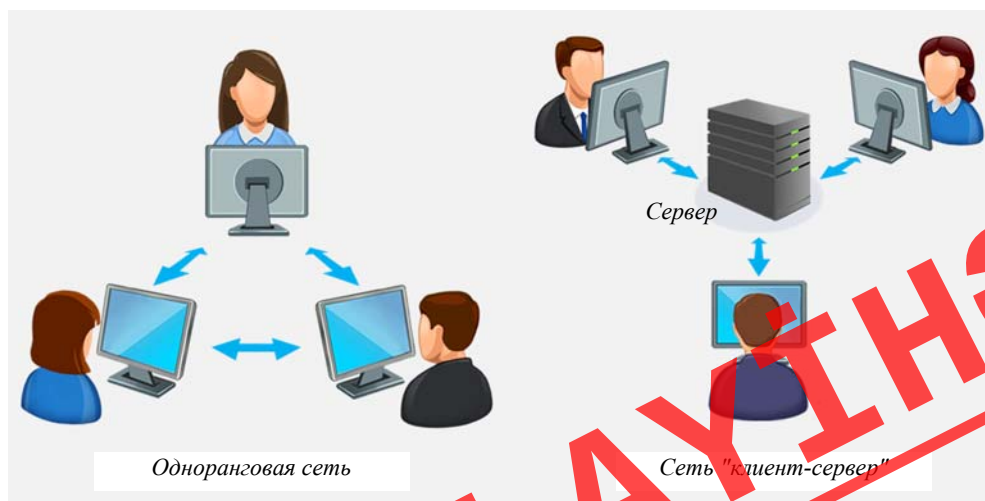
Постройте граф, соответствующий компьютерной сети вашей школы. Для этого каждый компьютер отметьте как вершины графа. Если нет сети, то представьте, что в классе перед каждым учащимся имеется компьютер, который связан с другими компьютерами класса с помощью кабеля.

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.

Памятка

- Локальная сеть
- Глобальная сеть
- Сеть "клиент-сервер"
- Одноранговая сеть
- Топология
- Топология "Шина"
- Топология "Кольцо"
- Топология "Звезда"

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.

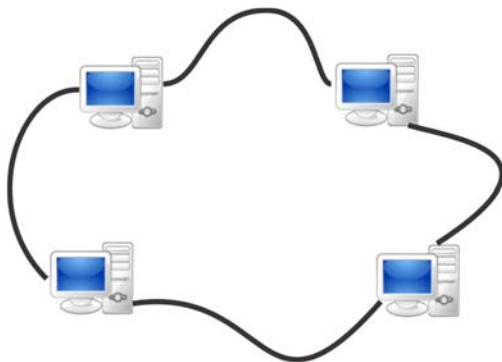
При создании сети большое значение имеет **топология**, то есть схема расположения и соединения сетевых устройств и кабелей сети. Необходимо выбрать такую структуру (граф), чтобы сеть работала надежно и эффективно и можно было бы с легкостью управлять потоком информации. Большинство сетей строится на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

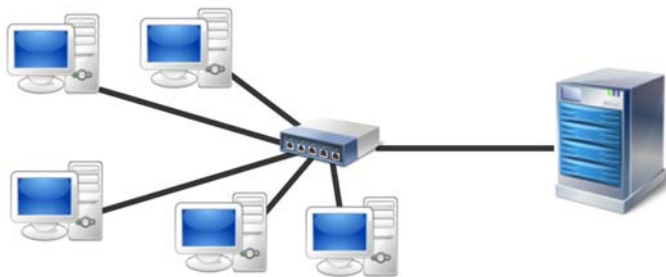


2. **Топология "кольцо".** В этой топологии каждый компьютер соединен с двумя другими компьютерами, принимает информацию от одного

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.

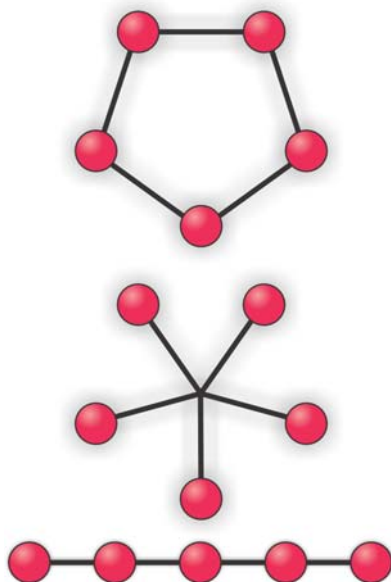


3. **Топология "звезда".** В этой топологии, появившейся еще на начальных этапах развития вычислительной техники, все компьютеры подсоединены к мощному центральному компьютеру. В такой конфигурации все данные передаются только через центральный компьютер, и за обмен информацией между всеми частями сети отвечает именно он.



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?



- Как вы представляете путь, который проходит информация при разговоре с помощью обычного и мобильного телефонов?
- Как передается информация посредством Интернета?

Деятельность

Определите порядок следования рисунков с исторической точки зрения. Опишите способы передачи информации в каждом из этих случаев.



Обсудим:

- Какой вид информации передается при помощи каждого из этих средств?
- Как по-вашему, какие еще способы передачи информации имели большое значение в развитии общества?

Если бы в обществе не было обмена информацией, то общество не развивалось бы. Распространение накопленной информации, полученных знаний, передача их из поколения в поколение является важнейшим фактором развития человеческого общества.

Любой процесс, связанный с передачей информации, можно представить с помощью следующей простой схемы.

Источник информации



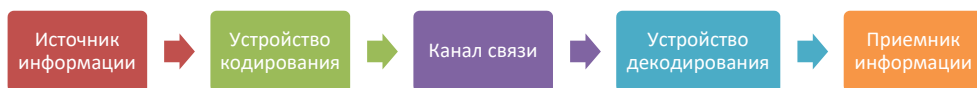
Канал связи



Получатель информации

Информация, вышедшая из **источника информации**, пройдя по **каналам связи**, доходит до **приемника информации**. Каждая часть этой схемы имеет свои свойства, то есть параметры, и качество передачи зависит от этих параметров.

Схема передачи информации, данная выше, упрощенная. В действительности, этот процесс происходит следующим образом:



Источником информации может быть как живое существо, так и техническое устройство (например, компьютер). Информация от источника поступает на **устройство кодирования** (например, модем). Это устройство преобразовывает поступившую к нему информацию в форму, удобную для передачи. Информация, пройдя по каналам связи (например, телефонной линии), попадает в устройство **декодирования** (например, модем), который преобразовывает информацию в форму, понятную для получателя.

Для передачи информации с одного компьютера на другой необходимо связующее звено – **канал связи**, по которому будут передаваться данные. В качестве примера такого канала можно привести телефонную сеть, играющую сегодня важную роль в нашей жизни. Телефонная система объединяет все телефонные сети – локальные, удаленные, мобильные, спутниковые. Хотя телефонная сеть предусмотрена для голосовой связи, сейчас её широко используют для передачи и других данных (факс, электронная почта, живое аудио и видео).

Для передачи данных по каналам необходимо иметь либо физические, либо беспроводные средства связи.

Физические средства. Для передачи данных существует несколько видов физических средств.

Кабель "витая пара". Используемый в большинстве телефонных систем, этот кабель самый дешевый. Он состоит из двух изолированных и переплетенных друг с другом проводов. Один из проводов несет сигнал, а второй заземляется. Заземленный провод (конец закопан в землю) проглатывает радиопомехи и тем самым защищает сигнал, передаваемый другим проводом.

Коаксиальный кабель. Этот вид кабеля, используемый в основном в телевизионной отрасли, широко применяется в компьютерной сети. Коаксиальный кабель состоит из двух проводов: экрана цилиндрической формы и центрального изолированного провода внутри него.

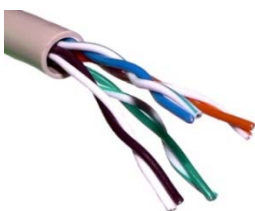
Памятка

- Физические средства
- Беспроводные средства
- Кабель "витая пара"
- Коаксиальный кабель
- Фиброоптический кабель
- Радиосигналы
- Микроволны
- Средства спутниковой связи

Благодаря этой изоляции сигналы, которые проходят в коаксиальном кабеле, не подвержены влиянию помехам, которые создают другие передатчики и электронные устройства. Несмотря на указанные преимущества, этот кабель несколько дороже кабеля витая пара.

Фиброоптический кабель. Фиброоптический кабель (волоконно-оптический кабель) изготавливается из тонких эластичных стеклянных трубочек. Этот кабель обладает некоторыми преимуществами перед традиционными металлическими линиями связи. Имея широкую полосу пропускания, он может передавать намного больше данных, значительно легче металлических проводов и менее подвержен помехам. Основным недостатком фиброоптического кабеля – его хрупкость и дороговизна.

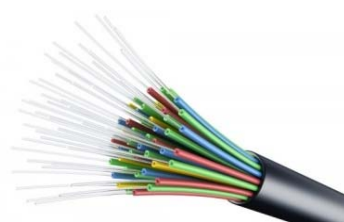
Физические средства для передачи данных



Кабель "витая пара"



Коаксиальный кабель



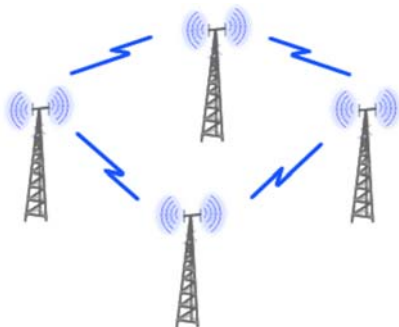
Фиброоптический кабель

В настоящее время наряду с физическими средствами широко используют и **беспроводные средства связи**.

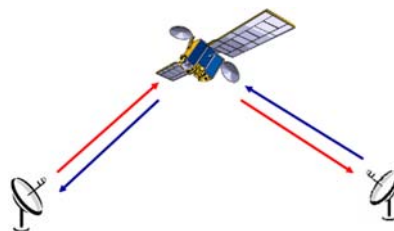
- **Радиосигналы.** Во время передачи данных при помощи радиосигналов, между антеннами передатчика и получателя должна быть прямая видимость, то есть сигнал из одного пункта в другой должен идти по прямой линии. Для радиовещания необходимы радиопередатчик, посылающий сигналы, и радиоприемник, принимающий сигналы.
- **Микроволны.** Микроволновые сигналы распространяются в пространстве в форме электромагнитных волн. Как и радиосигналы, они передаются по прямой от одной микроволновой станции до другой. Во избежание помех, большинство микроволновых станций устанавливается на возвышенностях или на высоких зданиях.
- **Средства спутниковой связи.** Спутниковые антенны находятся на орбите Земли на расстоянии десятков тысяч километров от ее поверхности. Такая высота позволяет спутникам, вращаясь со скоростью вращения Земли, сохранять стабильное положение в одной точке над её поверхностью. Оборудование на спутнике усиливает сигналы, посылаемые с Земли, и снова возвращает их к нам на Землю.

При конструировании сети и выборе одного из средств коммуникации необходимо учитывать несколько факторов – вид сети, ее размеры (масштаб) и ее стоимость.

Беспроводные средства для передачи данных



Микроволновые вышки



Спутниковая связь

Изучим сами

Соберите информацию из Интернета о технологиях Wi-Fi и Wi-Max.
Какова максимальная скорость передачи информации для этих технологий?
Какие преимущества и недостатки они имеют?

Проверьте себя



1. Какова общая схема передачи данных?
2. Что такое каналы связи и как их можно классифицировать?
3. Какие беспроводные средства связи используют для передачи данных?
4. Какой вид кабеля использован в компьютерной сети вашей школы?
5. Как вы думаете, при помощи каких сигналов передается информация в мобильных телефонах?

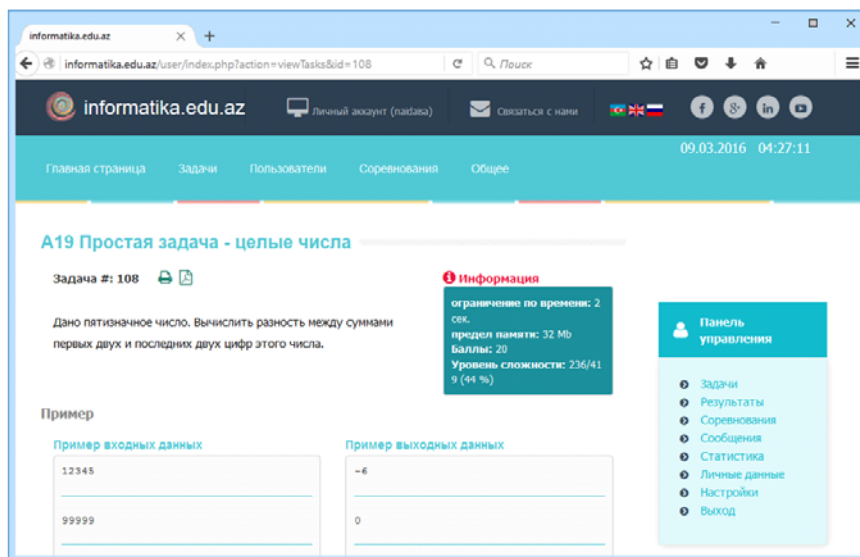
ЛАЙІНЭ



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

Деятельность

1. Запустите произвольный веб-браузер.
2. Перейдите на сайт *informatika.edu.az*.
3. Для перехода на русский язык выберите в верхней части сайта значок с соответствующим флагом.
4. Ознакомьтесь с назначением сайта, выбрав из меню **Общее** пункт **О проекте**.
5. Выберите в меню **Задачи** пункт **Весь список задач**. Ознакомьтесь с условиями нескольких задач. Выберите пункт **Статистика**. Рассмотрите подробнее диаграммы.
6. Щелкните по имени первого пользователя в списке 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 выполнили задачу A50 Простая задача - целые числа?

Выход компьютера в Интернет часто осуществляется посредством телефонной сети. К сожалению, скорость передачи информации по телефонной сети весьма невелика. Поэтому в Интернете для уменьшения количества символов, передаваемых за единицу времени, то есть трафик, 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.

Памятка

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

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 могут указываться 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).



Тим
Бернерс-Ли
(1955)

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Sample HTML Page</title>
  <meta http-equiv="Content-type" content="text
  <meta property="og:type" content="website" />
  <meta property="og:url" content="http://www.s
  <meta name="robots" content="index, follow" /
  <meta name="author" content="http://www.somed
  <link href="http://www.somedomain.com/" rel="
  <link href="http://www.somedomain.com/" rel="
```

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 – визуальный **HTML-редактор** (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 *informatika.edu.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?

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 "Деятельность", и определите структуру сайта, который будете создавать.

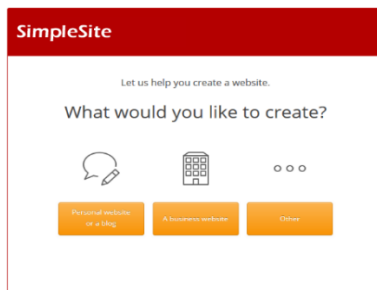
Деятельность

Разработка веб-сайта на основе шаблонов

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

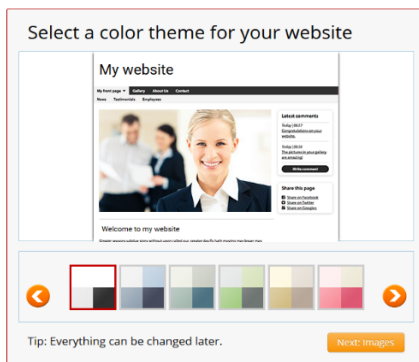


2. Выберите из меню пункт **Make a free website or blog**. Откроется следующая страница.

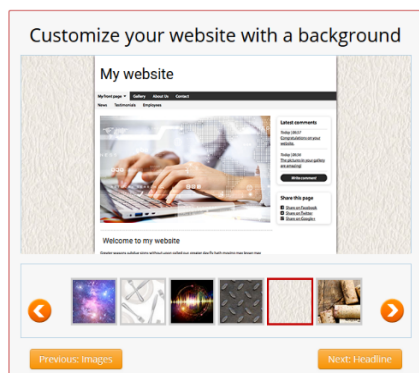


3. Щелкните по кнопке **Personal website or a blog**.

4. Выберите на открывшейся странице цветовую схему сайта и щелкните по кнопке Next: Images.

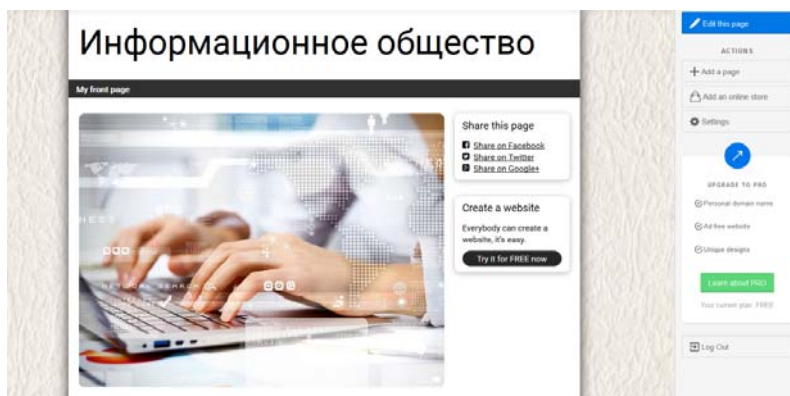


5. Для главной страницы будут предложены различные изображения. Выберите то, которое соответствует вашей теме, и щелкните по кнопке Next: Backgrounds.
6. Выберите из предложенных одно изображение для фона страницы сайта.

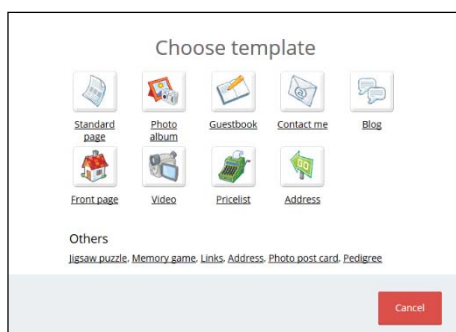


7. Щелкните по кнопке Next: Headline. Введите заголовок сайта на открывшейся новой странице. Например, **Информационное общество**.
8. Щелкните по кнопке Save and continue.
9. На открывшейся странице для задания имени сайта введите соответствующее название в указанное поле (например: **info-obshestvo**). 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  **Edit this page**, 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 называли "информационным обществом".

Памятка

- Информационное общество
- Электронное правительство

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 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.

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 informatika.edu.az и нажмите клавишу <Enter>. Откроется информация о том, где находится сервер сайта informatika.edu.az и карта размещения.

Search IP address or hostname:
Your IP address: 188.72.183.59

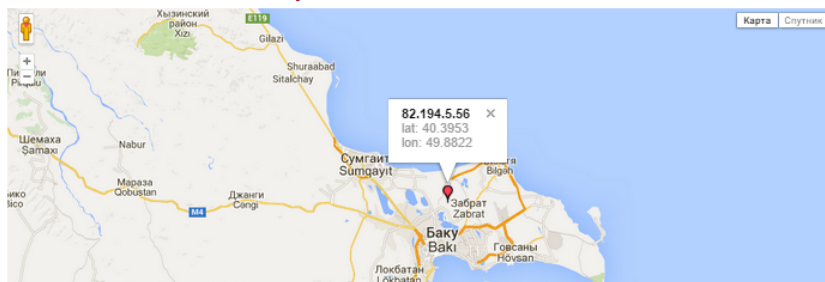
82.194.5.56 IP address Information

The IP address 82.194.5.56 was found in Baku, Baki, Azerbaijan. It is allocated to Azedunet LLC. Additional IP location information, as well as network tools are available below.

IP address: 82.194.5.56
ISP: Azedunet LLC
City: Baku
Region: Baki
Country: Azerbaijan (AZ)
latitude: 40.3953
longitude: 49.8822

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

82.194.5.56 Location Map



Обсудим:

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

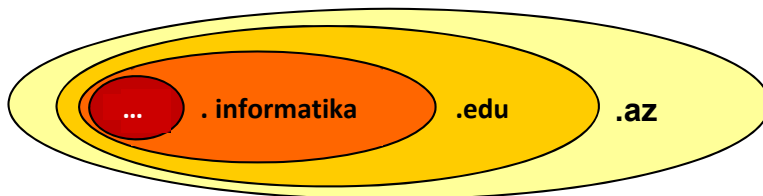
Каждый компьютер в сети имеет свой уникальный **IP-адрес** (произносится как "ай-пи адрес"). Каждый адрес – это 32-битное двузначное число. Для удобства это число представляют четырьмя десятичными числами с точками между ними. Каждое из этих десятичных чисел может получать значение от 0 до 255.

172 . 16 . 254 . 1
↓ ↓ ↓ ↓
10101100.00010000.11111110.00000001

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.informatika.edu.az*). То есть когда пользователь набирает в адресной строке соответствующее имя, DNS-серверы автоматически преобразовывают его в IP-адрес.

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 **U**niform **R**esource **L**ocator 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:

<http://informatika.edu.az/tasks.php?action=result>

Первая часть адреса <http://> является протоколом и показывает, что ресурс является веб-сайтом. **Протокол** – это правила или набор стандартов, обуславливающих 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 (*informatika.edu.az*) – это доменное имя компьютера, где размещен ресурс, вторая часть (*/tasks.php?action=result*) 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 "Intrans". Там регистрируются домены второго уровня. Например, *azerbaijan.az*, *ict.az*, *ayna.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.

Памятка

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

Изучим сами

При помощи существующей системы IP-адресации (IPv4) можно дать **4 294 967 296** ($256 \times 256 \times 256 \times 256$) 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. Запишите состоящий из четырех десятичных чисел IP-адрес 145.120.30.56 в 32-битном виде.

ОБОБЩАЮЩИЕ ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите способы получения и передачи информации.
2. Объясните принцип работы одноранговой сети.
3. Выясните названия компаний, действующих в вашем регионе (городе) в качестве провайдеров.
4. В каком случае информационная модель объекта представляется в форме графа?
5. Придумайте для одной из заданных тем структуру сайта и изобразите ее в форме графа.
 - а) "Моя школа"
 - б) "Моя семья"
 - в) "Моя Родина"
6. Для чего нужен IP-адрес?
7. Принадлежность к каким группам и структурам доменных имен указывает домен ".gov"?
8. Проанализируйте следующие доменные имена:
 - а) *bakimektebleri.edu.az*
 - б) *az.wikipedia.org*

Известные алгоритмы

Алгоритм Эвклида (нахождение наибольшего общего делителя)

Алгоритм Эвклида – это алгоритм нахождения наибольшего общего делителя двух целых чисел.

Как вы знаете из математики, **наибольший общий делитель** двух чисел (**НОД**) – это наибольшее число, на которое эти числа делятся без остатка.

Описание алгоритма нахождения НОД вычитанием

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

Пример:

Найти НОД для 30 и 18.

$$30 - 18 = 12$$

$$18 - 12 = 6$$

$$12 - 6 = 6$$

$$6 - 6 = 0 \quad \text{Конец: НОД равен уменьшаемому или вычитаемому.}$$

$$\text{НОД}(30, 18) = 6$$

Программа на языке Python

```
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.

Описание алгоритма нахождения НОД делением

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

Пример:

Найти НОД для 30 и 18.

$30/18 = 1$ (остаток 12)

$18/12 = 1$ (остаток 6)

$12/6 = 2$ (остаток 0). Последний делитель и есть НОД: $\text{НОД}(30, 18) = 6$

Программа на языке Python

```
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 (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` (одно из переменных всегда будет равно 0 и не окажет влияния на сумму).

$$\begin{aligned} \text{НОД}(a; b) &= \text{НОД}(a-b; b) \\ \text{Если } a > b \\ \text{НОД}(36; 48) &= \text{НОД}(36; 48-36) = \\ &= \text{НОД}(36; 12) = \text{НОД}(12; 36-12) = \text{НОД}(12; 24) = \\ &= \text{НОД}(12; 24-12) = \text{НОД}(12; 12) = 12 \end{aligned}$$

Программа на языке Python с использованием функции

```
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))

```

Задачи для самостоятельной работы

1. Напишите программу, находящую наибольший общий делитель трех чисел.
2. Используя алгоритм Эвклида для нахождения наибольшего общего делителя двух чисел, напишите программу для вычисления НОД N заданных чисел.
3. Используя формулу $\text{НОК} = ab / \text{НОД}(a, b)$, напишите программу вычисления наименьшего общего кратного (НОК) двух натуральных чисел a и b .
4. Даны N целых чисел. Найдите среди них такие два числа, чтобы их НОД был наибольшим.
5. Даны N целых чисел. Найдите среди них такие два числа, чтобы их НОК был наименьшим.

Вычисление факториала

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. Используя эту формулу и цикл `while`, программу, вычисляющую факториал числа, можно записать так:

Программа на языке Python

```

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)

```

Допустим $n = 5$, то есть надо вычислить $5!$. При первом проходе в теле цикла `while` переменной f присваивается 1×1 . После второго прохода $- 1 \times 2$, затем $- 2 \times 3$, 6×4 и 24×5 . В шестой раз цикл `while` не будет выполняться, так как значение переменной i станет равным 5 и условие $i < n$ не будет выполнено. Заметим, что по договоренности принято: $0! = 1$ и $1! = 1$

$$0! = 1$$

Задачи для самостоятельной работы

1. Напишите программу вычисления по формуле для заданных натуральных чисел n и k ($k < n$).

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

2. Напишите программу вычисления для заданного натурального числа n произведения $1 \times 3 \times \dots \times (2n-1)$.
3. Напишите программу вычисления для заданного натурального числа n произведения $2 \times 4 \times \dots \times (2n)$.
4. Напишите программу нахождения наименьшего натурального числа n , при котором $n!$ делится на 990.
5. Напишите программу, определяющую по заданному числу n количество единиц, присутствующих в числе $n!$.

Перевод числа из десятичной системы счисления в двоичную

Вы знакомы с алгоритмом получения двоичного числа из десятичного:

1. Данное десятичное число делится на два (на основание 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 два 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.

Программа на языке Python

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

print (n)
```

$$10_2 = 2_{10}$$

Задачи для самостоятельной работы

1. Напишите программу нахождения числа единиц в двоичной записи десятичного числа 123456.
2. Напишите программу, определяющую, чего больше в двоичной записи десятичного числа 135797: единиц или нулей.
3. Напишите программу нахождения числа единиц в двоичной записи натурального числа n в интервале $[1, 100]$. Число n вводится с клавиатуры.
4. Напишите программу нахождения количества натуральных чисел в интервале $[1, 100]$, в двоичной записи которых имеется только цифра 1.
5. Напишите программу нахождения количества натуральных чисел в интервале $[a, b]$, в двоичной записи которых имеется только цифра 1.

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.

Формула:

$$F_1 = 1$$

$$F_2 = 1$$

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

Пример:

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

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

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

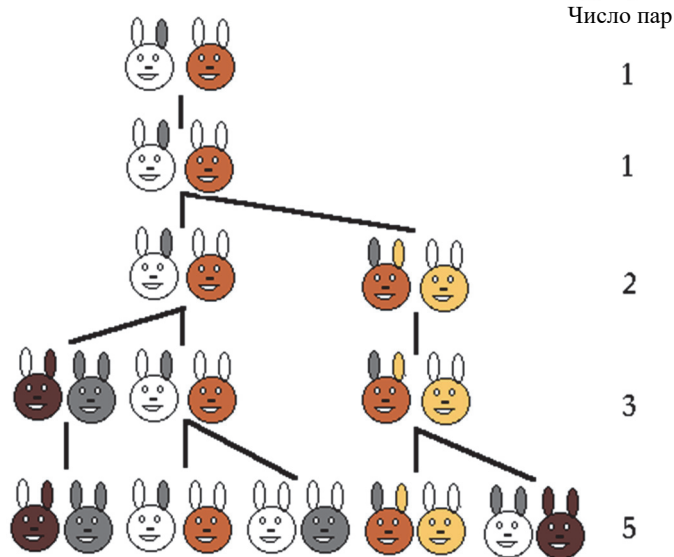
$$F_6 = F_5 + F_4 = 5 + 3 = 8$$

Алгоритм вычисления n-го числа в последовательности Фибоначчи

1. Вводится номер определяемого элемента **n**.
2. Проверяем: если **n < 3**, то выводим ответ 1.
3. Если же **n >= 3**, то
4. Первым двум членам последовательности **fib1** и **fib2** присваиваются начальные значения (1 и 1).
5. Затем, начиная с номера 3 до **n** (включительно) выполняем:
 - a. Вычисляется сумма **fib1** и **fib2** и результат присваивается третьей переменной **fib_sum**.
 - b. Производятся замены **fib1 = fib2** и **fib2 = fib_sum**.
6. Выводим ответ.

Программа на языке Python

```
fib1 = 1
fib2 = 1
n = int(input("Номер числа в последовательности
Фибоначчи:"))
if n<3:
    print(1)
else:
    fib1 = 1
    fib2 = 1
    i = 3
    while i <= n:
        fib_sum = fib2 + fib1
        fib1 = fib2
        fib2 = fib_sum
        i = i + 1
    print (fib_sum)
```



Задачи для самостоятельной работы

1. Напишите программу, выводящую на экран первые n чисел последовательности Фибоначчи. Число n вводится с клавиатуры.
2. Напишите программу, выводящую на экран n_1 -й и n_2 -й элемент последовательности Фибоначчи, а также их сумму. Числа n_1 и n_2 вводятся с клавиатуры.
3. Напишите программу, определяющую, является ли вводимое с клавиатуры число числом Фибоначчи.
4. Напишите программу, определяющую количество чисел Фибоначчи в заданном интервале $[a, b]$.
5. Напишите программу, находящую сумму всех чисел Фибоначчи, находящихся в интервале $[0, 100]$.

Словарь терминов

A

b

c

d

e

f

g

h

i

j

k

l

m

n

o

p

q

r

s

t

u

v

w

x

y

z

abacus ~ счеты, абак ~ abakus ~ çötkə, abak – счетная доска, разделенная на полосы, применявшаяся для арифметических вычислений в Древней Греции, Риме. Во время вычислений маленькие камешки или косточки передвигались по линиям или углублениям, которые представляли разряды числа.

absolute address ~ абсолютный адрес ~ salt adres ~ mütləq ünvan – в электронных таблицах: адрес ячейки, не изменяющийся при копировании и перемещении формулы.

absolute value (abs) ~ абсолютное значение ~ mutlak değer ~ mütləq qiymət – абсолютное значение числа, то есть его величина без знака.
active cell ~ активная ячейка ~ etkin hücre ~ aktiv xana – текущая (выделенная) ячейка электронной таблицы, то есть рабочая ячейка в текущий момент времени.

active window ~ активное окно ~ etkin pencere ~ aktiv pəncərə – в многооконных системах: окно, используемое в данный момент для ввода данных пользователя, создания изображений или окно для выбора пункта меню.

address ~ адрес ~ adres ~ ünvan – 1. Указывает ячейку в памяти компьютера. Каждая ячейка имеет уникальный адрес.
2. В электронной таблице буква и число, указывающие столбец и строку, на пересечении которых находится ячейка.
3. Набор чисел для указания компьютера в сети Интернет.
4. Адрес электронной почты.
5. Место веб-страницы в Интернете.

administrator ~ администратор ~ yönetici ~ inzibatçı, administrator – специалист, ответственный за контроль и обслуживание сети, системы управления базами данных, их проектирование, создание, конфигурацию.

algorithm ~ алгоритм ~ algoritma ~ alqoritm – конечная последовательность действий, выполняемая для достижения определенной цели.

algorithmic language ~ алгоритмический язык ~ algoritmik dil ~ alqoritmik dil – произвольный язык программирования, основанный на решении проблем (задач) с помощью алгоритмов.

alignment ~ выравнивание ~ **hizalama** ~ **düzləndirmə** – размещение объектов по вертикали или горизонтали на одном уровне.

analog ~ аналоговый ~ **örneksel** ~ **analoq** – термин, относящийся к объектам (например, речь), описываемым с помощью непрерывно меняющихся физических величин (например, амплитуда, напряжение, частота).

analog-to-digital converter (ADC) ~ аналого-цифровой преобразователь (АЦП) ~ **örneksel-sayısal dönüştürücü (ÖSD)** ~ **analoq-rəqəm çeviricisi** – устройство, преобразующее аналоговые сигналы в цифровые.

animation ~ анимация ~ **canlandırma** ~ **canlandırma, animasiya** – моделирование движения, осуществляемое путем отображения серии последовательно сменяющих друг друга образов на экране.

antivirus software ~ антивирусная программа ~ **virüs-önler yazılım** ~ **antivirus programı** – программа, защищающая компьютер от вирусов и других вредоносных программ.

application ~ приложение ~ **uygulama** ~ **tətbiqi program** – специализированная программа, предназначенная для выполнения однотипных работ, например: текстовый процессор, электронная таблица, система учета.

assembly language ~ язык ассемблера ~ **assembler dili, çevirici dili** ~ **assembler dili** – категория языков программирования низкого уровня, в которых каждому оператору соответствует ровно одна машинная команда.

assignment operator ~ оператор присваивания ~ **atama işleci** ~ **mənimlətmə operatoru** – оператор языка программирования, который назначает переменной конкретное значение.

at sign ~ коммерческое **at**, «собака» ~ **kuyruklu a** ~ **quyruqlu a** – символ **@**. Используется в адресе электронной почты для разделения имени пользователя от доменного имени (*informatika@edu.az*).

attachment ~ вложение ~ **ek** ~ **qoşma** – файл, который передается как часть электронного письма. В качестве вложения может быть текстовый, графический файл, программа и другие типы файлов.

backslash ~ обратная косая черта, наклонная черта влево ~ **ters eğik çizgi, sola eğik çizgi** ~ **tərs əyik cizgi, sola əyik cizgi** – символ **"\"**. Используется в некоторых операционных системах в качестве разделителя имен подкаталогов.

A

B

c

d

e

f

g

h

i

j

k

l

m

n

o

p

q

r

s

t

u

v

w

x

y

z

base ~ **основание** ~ **taban** ~ **əsas** – количество цифр, используемых в конкретной системе счисления. В компьютерах применяют четыре системы счисления: десятичную, двоичную, восьмеричную и шестнадцатеричную. В известной всем десятичной системе счисления (с основанием 10) используется десять цифр (от 0 до 9), в двоичной системе (с основанием 2) две цифры – 0 и 1; в шестнадцатеричной системе (с основанием 16) шестнадцать символов (от 0 до F), причем цифрам от 10 до 5 соответствуют буквы английского (латинского) алфавита. При записи числа в некоторой системе счисления ее основание указывается после числа в нижнем индексе в скобках, например: 24AE₍₁₆₎.

bit (binary digit) ~ **бит (двоичная цифра)** ~ **ikil** ~ **bit (ikilik rəqəm)** – один из двух знаков (0 и 1), используемых в двоичной системе счисления.

board ~ **плата** ~ **kart** ~ **lövhə, kart** – общий термин, обычно обозначающий плоскую электронную плату с электронными компонентами, которые соединяются печатными токопроводящими дорожками. Основная плата компьютера, на которой расположен микропроцессор, называется материнской.

buffer ~ **буфер** ~ **arabellek** ~ **bufer** – место промежуточного хранения данных: зарезервированная область памяти, в которой данные хранятся до их перемещения в запоминающее устройство. Часто используется для передачи данных или ускорения операций ввода-вывода. Некоторые устройства, такие как принтеры или поддерживающие их адаптеры, имеют собственный буфер.

bullet ~ **маркер** ~ **madde imi** ~ **nişan** – в узком смысле: точка (кружок), выделяющая небольшой фрагмент текста или каждый элемент списка. С той же целью часто используются и другие виды меток, например, квадратики и другие декоративные значки.

bus ~ **шина** ~ **veri yolu** ~ **şin** – группа электрических соединений для передачи данных между различными компонентами компьютера.

byte (bunary term) ~ **байт (двоичный элемент)** ~ **bayt, sekiz ikil** ~ **bayt (ikilik element)** – единица измерения информации, состоящая из восьми бит; в технике и системах хранения данных: эквивалент одиночного символа, например, буквы, цифры или знака пунктуации.

cache ~ **кэш** ~ **önbellek** ~ **keş** – специальная подсистема памяти, в которой запоминаются наиболее часто используемые значения данных, благодаря чему доступ к ним производится очень быстро.

calculator ~ калькулятор ~ **hesap makinesi** ~ **kalkulator** – устройство, выполняющее операции над числами. На латинском языке слово "calculus" означает "маленький камешек" (в древних вычислительных устройствах абак использовались маленькие камешки). Первый электронный ручной калькулятор был выпущен фирмой Texas Instruments в 1967 году.

call ~ вызывать ~ **çağırma** ~ **çağırmaq** – передавать в программе управление некоторому разделу кода (как правило, какой-либо подпрограмме), сохраняя всю информацию, необходимую для возобновления выполнения основной программы с точки вызова.

caption ~ заголовок ~ **başlık** ~ **başlıq** – 1. Заголовок документа. 2. Заголовок окна.

card ~ плата ~ **kart** ~ **kart** – термин, часто употребляемый по отношению к печатной плате или адаптеру.

cell ~ ячейка ~ **hücre, göze** ~ **xana** – 1. В электронных таблицах место пересечения столбца и строки. 2. Адресуемый (по имени или по номеру) элемент памяти.

central processing unit (CPU) ~ центральный процессор (ЦП) ~ **merkezi işlem birimi** ~ **mərkəzi prosessor** – блок выполнения вычислений и управления компьютером; устройство, интерпретирующее и выполняющее команды.

channel ~ канал ~ **kanal** ~ **kanal** – 1. Путь или соединение, по которому передается информация между двумя устройствами. По отношению к компьютеру канал может быть внутренним или внешним. Во внутренней среде компьютер использует канал, чаще называемый шиной, по которому обмениваются данными компоненты системы, например, клавиатура и память компьютера. 2. В системах связи канал: носитель для передачи информации, употребляется также термин "линия".

character ~ символ ~ **karakter** ~ **simvol** – буква, цифра, знак пунктуации или какой-либо другой спецзнак или управляющий код, например: A, 3 или &. Символ не обязательно должен иметь визуальное изображение на экране или на бумаге; например, пробел – это такой же символ, как буква или цифра.

click ~ щелчок, щелкнуть ~ **tık, tıklamak** ~ **çıqqlı, çıqqlatmaq** – нажать один раз кнопку мыши и тут же ее отпустить, не двигая мышь. Обычно это действие производится с целью выбора какого-либо элемента (при этом указатель устройства управления, например, мыши, помещают "поверх" выбираемого объекта и

a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z

нажимают активную кнопку мыши) или отмены этого действия, а также активизации программы или некоторой функции.

copy ~ копировать ~ **kopyalamak** ~ **kopiyalamaq, köçürmək** – дублировать информацию или воспроизводить ее в другой части документа, в другом файле, в другой области памяти или на другом носителе.

cursor ~ курсор ~ **imleç** ~ **kursor** – специальный экранный индикатор, например мерцающий символ подчеркивания или прямоугольник, который отмечает место появления избираемых символов.

desktop ~ рабочий стол ~ **masaüstü** ~ **iş masası** – рабочая область на экране, которая с помощью значков и меню моделирует поверхность стола. Пользователь может перемещать объекты на рабочем столе с помощью мыши точно так же, как он перемещает бумаги и другие предметы, работая за обычным столом. В операционной системе Windows рабочий стол является специальным каталогом для каждого пользователя. Обычно в нем бывает множество ярлыков программных файлов.

dialog box ~ диалоговое окно ~ **iletişim kutusu** ~ **dialog boksı** – в графическом пользовательском интерфейсе: специальное окно, отображаемое системой или приложением, когда необходим ответ пользователя.

digit ~ цифра ~ **rakam** ~ **rəqəm** – один из символов, используемых для выражения числа в системе счисления. В любой системе счисления количество возможных цифр равно основанию системы.

digital-to-analog converter (DAC) ~ цифро-аналоговый преобразователь (ЦАП) ~ **sayısal-örneksel dönüştürücü** ~ **rəqəm-analoq çeviricisi** – устройство, которое преобразует цифровые данные в аналоговый сигнал.

directory ~ каталог ~ **dizin** ~ **kataloq** – перечень имен подчиненных файлов и каталогов, хранящихся на диске; способ организации и группирования файлов на диске, избавляющий пользователя от необходимости иметь дело с большим количеством файлов. Каталог самого верхнего уровня называется *корневым*; каталоги, содержащиеся внутри других каталогов, называются подкаталогами. В операционных системах Windows и Mac OS каталоги называются папками.

download ~ загрузка ~ **indirmek** ~ **endirmək** – в системах связи: процесс передачи с помощью модема или сетевых линий копии файла из удаленного компьютера в компьютер, выдавший запрос.

drag ~ перетаскивать ~ **sürüklemek** ~ **sürükləmək** – в компьютерной графике: переносить изображение или объект с одного места на экране в другое, "зацепив" его и "перетаскив" его в другое место.

extension ~ расширение ~ **uzantı** ~ **uzantı** – в названии файла набор символов, добавляемых к имени файла в целях дополнения имени или описания файла.

file ~ файл ~ **dosya** ~ **fayl** – законченная именованная совокупность информации (например, программа), набор данных, используемый программой, или документ, созданный пользователем.

folder ~ папка ~ **dizin** ~ **qovluq** – в графическом пользовательском интерфейсе: место хранения программ и файлов; на экране символически изображается графическим значком в виде папки. Папка в определенном смысле эквивалентна каталогу.

font ~ шрифт ~ **yazı tipi** ~ **şrift** – набор символов одного стиля (например, Courier), начертания (например, курсив), толщины (например, полужирные) и размера.

gigabyte (GB) ~ гигабайт ~ **gigabayt** ~ **gigabayt** – в строгом смысле: один миллиард байт. Точное значение часто меняется с контекстом: в компьютерах количество байт часто выражается числом, кратным степени двойки, поэтому гигабайт может означать как 1000 Мбайт, так и 1024 Мбайт, то есть 2^{20} (1 048 576) байт.

graph ~ граф ~ **çizge** ~ **qraf** – в программировании: структура данных, состоящая из определенного (возможно, нулевого) числа вершин, соединенных между собой произвольным образом (с помощью ребер).

hexadecimal (hex) ~ шестнадцатеричная система счисления ~ **onaltılı sayı sistemi** ~ **onaltılıq say sistemi** – система счисления с основанием 16, в состав которой входят цифры от 0 до 9 и прописные или строчные буквы от A (эквивалент числа 10) до F (эквивалент числа 15).

Internet Service Provider (ISP) ~ поставщик услуг Internet, Интернет-провайдер ~ **İnternet servis sağlayıcısı** ~ **İnternet xidmət provayderi** – компания, предоставляющая возможность пользования службами Интернета (доступ в Интернет) как юридическим, так и физическим лицам.

italic ~ курсив ~ **italik** ~ **kursiv** – начертание шрифта при печати или отображении с равномерным наклоном вправо. Курсивом обычно выделяют иностранные слова, заглавия, технические термины и цитаты.

a
b
c
D
E
F
G
H
I
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z

label ~ 1. Этикетка ~ etiket ~ yarlıq – наклейка на дискете или на каком-нибудь устройстве; слово, символ, группа символов для идентификации файла, раздела устройства хранения данных, элемента программы или документа.

2. Метка ~ etiket ~ nişan – в электронных таблицах: описательное имя типа "Доход" или "Расход", идентифицирующее группу ячеек, поименованную величину или формулу.

3. Метка ~ etiket ~ nişan – в диаграммах: слово, имя или число, идентифицирующее отметку на оси графика.

LAN (Local Area Network) ~ локальная вычислительная сеть, ЛВС ~ yerel alan ağı, YAA ~ yerli hesablama şəbəkəsi, lokal hesablama şəbəkəsi – группа компьютеров или других устройств, расположенных в относительно ограниченной зоне и соединенных линиями связи.

machine code ~ машинный код ~ makina kodu ~ maşın kodu – окончательный результат трансляции с любого языка высокого уровня типа C или Pascal; содержит последовательность 0 и 1, загружаемых и выполняемых микропроцессором.

mainframe computer (= mainframe) ~ мейнфрейм ~ ana bilgisayar ~ meynfreym kompüteri – компьютер высокой мощности, разработанный для наиболее интенсивных вычислительных задач. Мейнфрейм обычно используется одновременно многими пользователями, работающими на подключенных к нему терминалах.

media ~ носитель ~ ortam ~ vasitə, daşıyıcı – собирательное значение, описывающее различные материалы, например бумага, магнитный диск или лента, используемые для хранения или передачи информации.

megabit (Mbit) ~ мегабит ~ megabit ~ meqabit – обычно 1048576 бит, но иногда интерпретируется как 1 миллион бит.

megabyte (MB) ~ мегабайт ~ megabayt ~ meqabayt – 1 миллион байт, то есть 1048576 байт (2^{20}).

megapixel ~ мегапиксель ~ megapiksel ~ meqapiksel – один миллион пикселей; этот термин используется при указании разрешения цифровых камер, сканеров и других устройств. Например, изображение размера 1024×1024 пикселей часто именуют мегапикселем.

modem ~ модем ~ **modem** ~ **modem** – устройство связи, позволяющее передавать информацию из компьютера по обычной телефонной линии.

motherboard ~ материнская плата ~ **anakart** ~ **ana lövhə** – основная плата, содержащая процессор, оперативную память, контроллер шины, соединители и другие компоненты компьютерной системы.

network ~ (вычислительная) сеть ~ **ağ** ~ **şəbəkə** – группа компьютеров и связанных с ними устройств, соединенных средствами связи. Сеть может включать постоянные соединения, например, с помощью кабелей, и временные – по телефону или другим временно коммутируемым линиям связи; иметь небольшие размеры (масштаб), например, локальная вычислительная сеть, состоящая из небольшого количества компьютеров, принтеров и других устройств. Существуют довольно большие сети, например, глобальные, состоящие из большого количества компьютеров, расположенных в обширной географической зоне.

password ~ пароль ~ **parola** ~ **parol** – один из способов ограничения доступа к компьютерной системе и ее файлам.

peer-to-peer network ~ одноранговая сеть ~ **eşler arası ağ** ~ “**tay-tuşlar**” **şəbəkəsi** – сетевая архитектура, основанная на равноправии в ней всех компьютеров.

peripheral ~ периферийный ~ **çevresel, yanbirim** ~ **çevrəsəl, periferiya** – устройство, например, диск, принтер, модем или джойстик, подключаемое к компьютеру и управляемое его микропроцессором.

personal computer (PC) ~ персональный компьютер ~ **kişisel bilgisayar** ~ **fərdi kompüter** – компьютер, разработанный для использования одним человеком.

pixel ~ пиксель ~ **piksel** ~ **piksel** – одна из точек, формирующих изображение на экране или при печати.

pointer ~ указатель ~ **göstergeç** ~ **göstərici** – 1. В графических средах: значок или символ, например, стрелка, управляемая мышью или другим устройством для указания, выделения, выбора элементов изображения, экрана или меню.

2. В программировании и обработке информации: переменная, содержащая адрес области памяти, а не сами данные.

a
b
c
d
e
f
g
h
i
j
k
l
M
N
o
P
q
r
s
t
u
v
w
x
y
z

port ~ **порт** ~ **port** ~ **port** – соединение (разъем) для подключения периферийных устройств компьютера (в некоторых случаях расположено на задней панели системного блока).

printer ~ **принтер** ~ **yazıcı** ~ **printer** – устройство, предназначенное для отображения (печати) информации на бумажном носителе, то есть устройство вывода для получения твердой (бумажной) копии документа.

RAM (random access memory) ~ **оперативная память, оперативное запоминающее устройство (ОЗУ), память прямого доступа, память с произвольной выборкой** ~ **rastgele erişimli bellek** ~ **ixtiyari erişimli yaddaş, operativ yaddaş** – устройство памяти для хранения данных, связанное с микропроцессором и участвующее в его операциях над данными.

resolution ~ **разрешающая способность** ~ **çözünürlük** ~ **çözüm-lülük** – характеристика, определяющая четкость изображения. При выводе изображения на экран его разрешение определяется количеством пикселей на 1 дюйм; при выводе изображения на принтер – количеством точек на 1 дюйм.

screen saver ~ **хранитель экрана** ~ **ekran koruyucusu** ~ **ekran qoruyucusu** – программа, отключающая экран монитора или вывод-
ящая на него анимационную картинку в случае простоя компьютера.

screenshot ~ **снимок экрана** ~ **ekran görüntüsü** ~ **ekran görüntüsü** – рисунок, полученный копированием изображения экрана или отдельной его части.

scroll bar ~ **полоса прокрутки** ~ **kaydırma çubuğu** ~ **fırlatma zolağı** – в графических пользовательских интерфейсах: вертикальная или горизонтальная полоса, расположенная справа или в нижней части документа, служащая для прокрутки документа, не уместящегося в окне.

search ~ **поиск** ~ **arama** ~ **arama, axtarış** – нахождение местоположения файла или определенных данных.

search engine ~ **поисковая система** ~ **arama motoru** ~ **axtarış sistemi** – программа, дающая возможность поиска информации по ключевым словам, темам и другим признакам (например, AltaVista, Excite, Google, HotBot, Infoseek, Lycos, Magellan, Rambler, Yahoo!).

sector ~ сектор ~ kesim ~ sektor – часть области хранения данных на диске. Диск имеет нижнюю и верхнюю стороны, кольцевые дорожки на каждой из них и сектора. Во время форматирования каждая дорожка делится на сектора определенного объема (обычно – 512 байт).

setup ~ установка ~ kurmak ~ quraşdırma – в программном обеспечении: процедура подготовки программы для ее работы на компьютере и подсоединенных к нему устройств.

shortcut ~ ярлык ~ kısayol ~ qısayol – файл, служащий указателем на объект, программу или команду и содержащий дополнительную информацию. Чаще всего ярлыки создаются на рабочем столе для быстрого запуска программ, находящихся в «неудобных» местах.

software ~ программное обеспечение ~ yazılım ~ proqram təminatı – программы вычислительной машины; команды, которые заставляют аппаратные средства машины работать.

source code (SRC) ~ исходный код ~ kaynak kodu ~ ilkin kod – читаемые человеком операторы программы, написанной на языке высокого уровня или ассемблере. Объектный код программы вырабатывается из исходного кода в виде, понятном компьютеру.

statement ~ оператор ~ deyim ~ deyim – самый малый исполняемый элемент в языке программирования. Вообще каждая строка программы – это индивидуальный оператор и считается отдельной командой.

status bar ~ строка состояния ~ durum çubuğu ~ status zolağı – горизонтальная полоса, расположенная в нижней части окна программы и отображающая важную информацию для пользователя.

storage ~ запоминающее устройство (ЗУ) ~ depolama, bellek ~ yaddasaxlama qurğusu – в вычислительной технике: внешнее устройство, на котором хранится информация компьютера.

taskbar ~ панель задач ~ görev çubuğu ~ tapşırıqlar zolağı – в операционной системе Windows: горизонтальная панель в нижней части экрана, включающая список задач. Панель задач можно разместить также в верхней части экрана, в его левой и правой части.

a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
T
u
v
w
x
z

title bar ~ строка заголовка ~ **başlık çubuğu** ~ **başlıq zolağı** – в графическом пользовательском интерфейсе: горизонтальная полоса в верхней части окна; на ней указывается имя окна (программы) и размещаются другие кнопки.

topology ~ топология ~ **topoloji** ~ **topologiya** – конфигурация, образованная связями между устройствами в локальной сети. Большинство сетей основывается на трех топологиях: шина, кольцо и звезда.

tree ~ дерево ~ **ağaç** ~ **ağac** – иерархическая структура данных, имеющая корень и несколько ветвей. В отличие от обычного дерева, изображается как правило в перевернутом виде.

turtle ~ черепашка ~ **kaplumbağa** ~ **bağa** – небольшая фигурка на экране, обычно треугольной формы или в виде изображения черепашки, которая действует как указатель на экране компьютера.

utility ~ утилита ~ **yardımcı** ~ **utilit** – небольшая программа, например, редактор или отладчик, предназначенная для выполнения определенной функции.

wallpaper ~ обои ~ **duvar kağıdı** ~ **divar kağızı** – в компьютерной технике: изображение для оформления рабочего стола в графических операционных системах. В качестве обоев рабочего стола часто используют растровые изображения, размеры которых соответствуют разрешению экрана (например, 1024×768 или 1280×1024 пиксель).

web browser (= **browser**) ~ браузер, веб-обозреватель ~ **web tarayıcısı**, **ağ gözaticısı** ~ **vab brauzer** – программное обеспечение для просмотра, поиска, обработки, отображения на экране веб-сайтов, перехода с одной веб-страницы на другую в сети Интернет.

web page ~ веб-страница ~ **ağsayfası** ~ **vab-səhifə** – документ HTML, размещенный во Всемирной паутине.

web server ~ веб-сервер ~ **web sunucusu** ~ **vab-server** – компьютер, предоставляющий доступ пользователям к службам и веб-страницам сетей Интернет и интранет.

web site (site) ~ веб-сайт ~ **web sitesi** ~ **vab-sayt** – совокупность веб-страниц с общим именем, темой и навигационной системой.

window ~ **окно** ~ **pencere** ~ **pəncərə** – прямоугольная область на экране монитора, служащая для взаимосвязи с прикладной программой или ее частью. В зависимости от назначения различают программные окна, окна документов и диалоговые окна.

wizard ~ **мастер** ~ **sihirbaz** ~ **sehrbaz** – вспомогательная функция, предоставляющая пользователю подсказку или другую нужную информацию во время работы с программой. Как правило, помощь "мастера" бывает нужна начинающим пользователям.

word processor ~ **текстовый процессор** ~ **kelime işlemci** ~ **mətn prosessoru** – разновидность прикладной компьютерной программы для работы (набор, редактирование, форматирование, печать) со всевозможными печатными документами.

worksheet ~ **рабочий лист** ~ **işlem tablosu** ~ **iş vərəqi** – файл данных или его часть, созданный и используемый электронной таблицей. Файл электронной таблицы называют книгой. Книга состоит из нескольких рабочих листов, количество которых можно увеличивать.

workstation ~ **рабочая станция** ~ **iş istasyonu** ~ **iş stansiyası** – общий термин для обозначения персональных компьютеров в локальной сети или в среде клиент-сервер.

World-Wide Web (WWW) ~ **Всемирная паутина, Сеть** ~ **dünya çapında ağ** ~ **Dünya hörümçək toru** – распределенная система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету. Для обозначения Всемирной паутины часто используют слово "веб" и аббревиатуру WWW.

a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z

BURAXILIŞ MƏLUMATI

İNFORMATİKA

*Ümumtəhsil məktəblərinin 9-cu sinfi üçün
İnformatika fənni üzrə*

DƏRSLİK

Rus dilində

Tərtibçi heyət:

Müəlliflər

**Ramin Əli Nazim oğlu Mahmudzadə
İsmayıl Calal oğlu Sadıqov
Naidə Rizvan qızı İsayeva**

Elmi redaktor

Rasim Əliquliyev

Tərcüməçi

Naidə İsayeva

Redaktor

Natella Rüstəмова

Bədii redaktor

Taleh Məlikov

Texniki redaktor

Zeynal İsayev

Dizayner

Taleh Məlikov

Rəssamlar

Məzahir Hüseynov, Elmir Məmmədov

Korrektor

Olqa Kotova

© Azərbaycan Respublikası Təhsil Nazirliyi (qrif nömrəsi:)

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

Hesab-nəşriyyat həcmi: 6,75. Fiziki həcmi: 8,5 çap vərəqi.

Formatı: 70×100¹/₁₆. Səhifə sayı: 136.

Şriftin adı və ölçüsü: Times New Roman qarnituru 10-12 pt. Ofset kağızı. Ofset çapı.

Sifariş ___. Tiraj ___. Pulsuz. Bakı – 2020.

Əlyazmanın yığma verildiyi və çapa imzalandığı tarix: 03.06.2016

Nəşriyyat:

“Bakınəşr” (Bakı, H.Seyidbəyli küç., 30)

Çap məhsulunu istehsal edən:

“Təhsil NP” MMC (Bakı, F.Xoyski küç., 121a)

LAYİHƏ

Pulsuz



Əziz məktəbli !

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

İnanırıq ki, sən də bu dərsliyə məhəbbətlə yanaşacaq, onu zədələnmələrdən qoruyacaq, təmiz və səliqəli saxlayacaqsan ki, növbəti dərs ilində digər məktəbli yoldaşın ondan sənin kimi rahat istifadə edə bilsin.

Sənə təhsildə uğurlar arzulayırıq!

LAYİHƏ