Вопросы к экзамену по дисциплине
«Информатика и основы искусственного интеллекта»

Лекция1

**1.Информация, как продукт взаимодействия данных и адекватных им методов.**

Информатика-это научная и прикладная область знаний, изучающая законы, методы и способы накопления, обработки и передачи информации с помощью компьютерных и других технических средств

Информация возникает и существует в момент диалектического взаимодействия объективных данных и субъективных методов.

1. Данные (фиксируемые в виде определенных сигналов воспринимаемые факты окружающего мира) ОБЪЕКТИВНЫ
2. Методы (преобразуют, транспортируют, дают возможность потреблять данные) СУБЪЕКТИВНЫ

**2. Свойства информации.**

АТРИБУТИВНЫЕ(присутствуют всегда):

- неотрывность информации от физического носителя и языковая природа информации;

Хотя информация и неотрывна отфизического носителя и имеет языковую природу она не связана жестко ни с конкретным языком, ни с конкретным носителем

- дискретность;

Знания характеризуют отдельные фактические данные, закономерности и свойства изучаемых объектов, которые распространяются в виде различных сообщений, состоящих из буквы, цифры,символа, знака

- непрерывность.

Информация имеет свойство сливаться с уже зафиксированной и накопленной ранее, тем самым способствуя поступательному развитию и накоплению.

ПРАГМАТИЧЕСКИЕ(появляются в процессе использования):

- полнота:

Информация полна, если ее достаточно для понимания и принятия решений. Не полнота информации сдерживает принятие решений или может повлечь ошибки

- достоверность:

Информация достоверна, если она не искажает истинное положение дел. Недостоверная информация может привести к неправильному пониманию или принятию неправильных решений

- адекватность:

Степень соответствия реальному объективному состоянию дел зависит от адекватности информации.

- доступность:

Информация становится понятной,если она выражена языком, доступным людям, для которых она предназначена

- актуальность:

При работе в постоянно изменяющихся условиях важно иметь актуальную, т. е. соответствующую действительности,информацию

ДИНАМИЧЕСКИЕ (характеризуют изменение информации во времени):

- рост:

Определяет свойство многократногораспространения или повторяемости информации

- старение

## 3. Виды информации.

ПО СПОСОБАМ ВОСПРИЯТИЯ:

- визуальная
- аудиальная
- тактильная
- обонятельная
- вкусовая

ПО ФОРМЕ ПРЕДСТАВЛЕНИЯ:

- текстовая
- числовая
- графическая
- звуковая

ПО ОБЩЕСТВЕННОМУ ЗНАЧЕНИЮ:

- массовая
- специальная
- личная

## 4. Информационная система. Схемы управления информационных системам

**-** это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Схемы управления информационных систем:

Любая информационная система может действовать по правилам разомкнутой или замкнутой схемы управления.

В **разомкнутой информационной системе** получаемая потребителем информация используется произвольно. От потребителя в информационную систему ничего не поступает (библиотечный каталог)

В **замкнутой информационной системе** существует тесная связь между ее структурой и потребителем. Это достигается за счет введения в ее структуру канала обратной связи (система бронирования билетов)

## 5. **Информационные процессы.**

Информационный процесс - совокупность последовательных действий (операций), производимых над информацией (в виде данных, сведений, фактов, идей, гипотез, теорий и пр.) для получения какого-либо результата (достижения цели).

К основным информационным процессам, изучаемям в курсе информатики,

относятся:

- поиск
- отбор
- хранения
- передача
- кодирование
- обработка
- защита информации (сл. 18)

## 6. **Информационные революции. Их важность сточки зрения информатики**.

Этапы появления средств и методов обработки информации, вызвавшие кардинальные изменения в обществе, определяются как **информационные революции.**

Информационные революции привели к появлению многих новых областей знаний, одной из которых является информатика.

ПЕРВАЯ ИНФОРМАЦИОННАЯ РЕВОЛЮЦИЯ

Связана с изобретением письменности, обусловившей гигантский качественный скачок вразвитии цивилизации. Появилась возможность накопления знаний и их передачи последующим поколениям. С позийии информатики это можно оценить как появление средств и методов накопления информации.

ВТОРАЯИНФ РЕВОЛЮЦИЯ (середина 16 века)

Связана с изобретением книгопечатания, изменившего человеческое общество, культуру и организацию деятельности самым радикальным образом. С точки зрения информатики,значение этой революции в том, что она выдвинула качественно новый способхранения информации.

ТРЕТЬЯИНФ РЕВОЛЮЦИИЯ (конец 19 века)

СВязана с изобретением электричества, благодаря которому появились телеграф, телефон,радио, позволяющие оперативно передавать и

накапливать информацию в любом объеме. Этот этап важен для информатики, прежде всего тем, что ознаменовал появления средств информационной коммуникации.

ЧЕТВЕРТАЯИНФ РЕВОЛЮЦИЯ (70-е годы 20 века)
Связанас изобретением микропроцессорной технологии и появлением персонального компьютера. Толчком к четвертой инф революции послужило изобретение в середине 40-х годов электронно-вычислительной машины (ЭВМ).

## 7. **Кодирование текстовой и числовой информации.Системы счисления.**

Для реализации работы с данными, относящимися к различным типам, используется прием кодирование, то есть выражение данных одного типа через данные другого типа.
Естественные человеческие языки - это не что иное, как системы кодирования понятий для выражения мыслей посредством речи.
Система кодирования существующая в вычислительной технике называется двоичным кодированием и основана на представлении данных последовательностью всего двух знаков: 0 и 1.

Эти знаки называются **двоичными цифрами**, по-английски – **bi**nary digi**t** или сокращенно bit (**бит**).
Одним битом могут быть выражены два понятия: **0** или **1** (**да** или **нет**, **истина** или **ложь)**.

Если количество битовувеличить до двух, то уже можно выразить четыре различных понятия:

00 01 10 11
0 1 2 3

Тремя битами можно закодировать восемь различных значений:

000 001 010 011 100 101 110 111
0 1 2 3 4 5 6 7

Система, основанная на16разрядном кодировании символов, получила **название универсальной** – **UNICODE** . Шестнадцать разрядов позволяют обеспечить уникальные коды для 65 536 различных символов – этого поля достаточно для размещения в одной таблице символов большинства языков планеты.На текущий момент стандарт является основным в Интернете.
Символический метод записи чисел, когда каждому числу дается уникальное представление.
Системыразделяются на:

1. позиционные

- двоичные
- восьмеричные
- десятичные
- шестнадцатеричные

2. непозиционные

- римская

## 8. Правила перевода из десятичной системысчисления в любую другую позиционную систему счисления и наоборот.

ДЕСЯТИЧНАЯ СС:
Основание системы - 10.
Содержит 10 цифр: 1,2,3,4,5,6,7,8,9,0.
Любое десятичное число можно представить
как сумму степеней числа 10.
Пример:

$$3751_{10} = 3 \cdot 10^3 + 7 \cdot 10^2 + 5 \cdot 10^1 + 1 \cdot 10^0$$

ВОСЬМЕРИЧНАЯ СС:
Основание системы - 8.
Содержит 8 цифр: 0,1,2,3,4,5,6,7.
Любое восьмеричное число можно представить
как сумму степеней числа 8.

Пример:

3751, 1607, 62

ШЕСТНАДЦАТИРИЧНАЯ СС:

Основание системы -16.

Содержит 16 символов:0,1,2,3,4,5,6,7,8,9,

A, B, C, D, E, F

10 11 12 13 14 15

Любое шестнадцатеричное число можно представить как сумму степеней числа16.

Примеры шестнадцатеричных чисел :

32FD8,A6E, 156, 3751.

ДВОИЧНАЯ СС:

Основание системы -2.

Содержит 2 символа: 0,1

Любое двоичное число можно представить как сумму степеней числа 2.

Примеры двоичных чисел :

101101, 101, 1101011001.

СЛАЙД14,15,16, 17 в обратном направлении (там картинки наглядно объясняют происходящее)

- Для перехода из восьмеричной системысчисления в десятичную необходимо восьмеричное число представить в виде суммы степеней восьмерки и найти ее десятичное значение.

Пример: $35_8 = 3* 8^1+ 5*8^0 =24 + 5=29_{10}$

- Для перехода из шестнадцатеричной системы счисления в десятичную необходимо шестнадцатеричное число представить ввиде суммы степеней шестнадцати и найти ее десятичное значение.

Пример: $3A5_{16} = 3*16^2+ 10*16^1 + 5*8^0 =768 +160 +5=933_{10}$

## 9. Правила перевода из двоичной системысчисления в восьмеричную и шестнадцатеричную системы и наоборот.

Основание системы -2.

Содержит 2 символа: 0,1

Любое двоичное число можно представить как сумму степеней числа 2.

Примеры двоичных чисел :

101101, 101, 1101011001.

- из двоичной в восьмеричную: Разбить двоичное число на триады справа налево. Заменить каждую триаду восьмеричной цифрой.Недостающие позиции в крайней левой триаде дополнить нулями.

Пример: $10010101_2$= 010 . 010 . 101 =$225_8$

- из восьмеричной в двоичную: Каждую цифру восьмеричного числа представить в двоичном виде (три двоичных разряда).Отбросить нули в крайней левой позиции

Пример: $225_8$= 010. 010 .101=10 010 $101_2$

- из двоичной в шестнадцатеричную:Разбить двоичное число на четвертки справа налево. Заменить каждую четвертку двоичных цифр шестнадцатеричной цифрой. Недостающие позиции в крайней левой четвертки дополнить нулями.

Пример: $110010101_2$= 0001 . 1001 . 0101 =$195_{16}$

- из шестнадцатеричную в двоичную:Каждую цифру шестнадцатеричного числа представить в двоичном виде(4 двоичных разряда). Отбросить нули в крайних левых позициях

Пример: $195_{16}$= 0001 . 1001 . 0101= 11 001 $0101_2$

## 10. Кодирование растровых изображений.Преимущества и недостатки.

Графическаяи звуковая информация в ЭВМ представляется в **дискретной** форме.

Преобразование информации из аналоговой формы в дискретную называется дискретизацией, при этом графическое изображение или звуковой сигнал разбивается на отдельные элементы икаждому элементу присваивается в виде кода.

В соответствии с методами,применяемыми для представления изображений, их можно разделить на двекатегории: растровые и векторные.

**Растровое изображение** представляет собой **набор точек (**элементов изображения), которые называются пикселями.

Основные форматы графических файлов:

.bmp – хранит информации о каждой точке изображения,

.gif , .jpg, .tiff, .png – используется сжатие файла.

**Векторное изображение** представляет собой **набор линий идуг**.

Основные форматы графических файлов:

.wmf -универсальный формат для Windows-приложений.

.eps- формат файлов, предназначенный для обмена графическими данными между различными приложениями.

.cdr -файл проекта, созданный в программе CorelDRAW.

Представление растровых изображений: Линейные координаты и яркость каждой точки можно выразить с помощью целых чисел, т.е. можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графических данных.

Для кодирования цветных графических изображений применяется принцип **декомпозиции** произвольного цвета на основные составляющие. В качестве таких составляющих используют три основных цвета: *красный (**Red , R**), зеленый (**Green , G**) и синий (**Blue , B**).*

Такая система кодирования называется системой ***RGB.***

Расчет объема графической информации сводится к вычислению произведения количества точек на изображении на количество разрядов, необходимых для кодирования цвета одной точки.

Например, для цветной картинки, составленной из 256 цветов в графическом режиме монитора 640 x 480, требуется объемвидеопамяти, равный: 8 • 640 • 480 = 2457600 бит = 307200 байт = 300 Кбайт.

Основные преимущества:

- позволяет создать максимально реалистичные изображения (к примеру, фотографии), которые можно использовать в рекламном производстве, оформлении сайтов и т.д.;

- позволяет создать рисунок любой сложности, с плавными переходами цветов, с различной глубиной цвета, большим количеством деталей;
- намного большая распространенность,по сравнению с векторной, растровую графику можно встретить на плакатах, сайтахи других часто встречаемых местах;
- открыть файл с растровым изображением намного проще, так как большинство программ для просмотра изображений поддерживают форматы, в котором хранятся такие изображения;
- возможность быстро обработать изображение.

Недостатки:

- потеря качества при значительном увеличении (картинка становится «зернистой», то есть, становится видно те самые пиксели);
- если изображение имеет большое разрешение, то имеет и большой размер файла;
- невозможно масштабировать без потери первоначального качества;
- невозможно произвольно поворачивать изображение (только от 90 градусов) без искажения картинки;
- самое простое изображение будет занимать больше места, чем такое же векторное;
- невозможно вывести на печать навекторный графопостороитель.

## 11. Кодирование векторных изображений.Преимущества и недостатки.

Векторные изображение строятся с помощью математических описаний объектов (например,прямая описывается уравнением, окружность - координатами центра и радиусом).

Всеобъекты имеют свойства (атрибуты): толщины, цвет, тип линий.

Такое описание заставляет устройство само рисовать изображение, а не воспроизводить комбинацию пикселей.

Преимущества:

- Масштабирование, растягивание,перемещение без ухудшения качества изображений.
- Небольшой размер простых иллюстраций, упрощающий хранение и

отправку файлов заказчикам.

- Универсальность применения: на сайте, для контекстной и таргетированной рекламы, в полиграфии.

Основные недостатки:

- Большой размер файла с высокой детализацией графики и, как следствие, повышенные требования к производительности компьютера.
- Ограниченность в использовании эффектов: теней, градиентов, свечения и пр.
- Сложность создания детализированных фотореалистичных изображений.
- Трудности в совместимости форматов с приложениями из-за конкуренции фирм-производителей.

## 12. **Кодированиезвуковой информации**.

Припреобразовании звуковой информации в цифровую форму ее подвергают **дискретизации** и **квантованию.**

**Дискретизация** заключается в замерах величины аналогового сигнала огромное количество раз в секунду. Полученной величине аналогового сигнала сопоставляется определенное значение из заранее выделенного диапазона: 256 (8бит) или 65536 (16бит). Приведение в соответствие уровня сигнала определенной величине диапазонаи есть **квантование.**

Чем больше разделен сигнал в секунду, тем выше **частота дискретизации**. Именно по ней мы часто и определяем качество звука: 44 кГц,96 кГц и т.д. - т.е. на сколько тысяч таких участков была разделена секундазвука.

НА СЛАЙДЕ 36 ПОКАЗАН ГРАФИК

### 13. Этапы созданияЭВМ:

1. **Первоепоколение ЭВМ (1946 – начало 50-х годов)**:

· Использованиеэлектронных ламп в качестве элементной базы.

· Большие размерыустройств и низкая производительность. Примером является ENIAC, способныйвыполнять 300 операций умножения или 5000 операций сложения в секунду.

· Проблемы:сложность программирования, высокая стоимость и

ненадежность, так как лампычасто выходили из строя.

· Основноеназначение: научно-технические расчеты.

2. **Второепоколение ЭВМ (конец 50-х – начало 60-х годов)**:

· Переход натранзисторы, что позволило уменьшить размеры и повысить производительностькомпьютеров.

· Быстродействиедостигло 105-106 операций в секунду, ёмкость оперативной памяти увеличилась до1000 Кбайт.

· Появление первыхязыков программирования высокого уровня, таких как Фортран и Кобол, чтоупростило процесс программирования.

3. **Третьепоколение ЭВМ (1965-1971 годы)**:

· Использованиемикросхем в качестве элементной базы.

· Увеличениепроизводительности до 10 млн операций в секунду и уменьшение размеровустройств.

· Совместимость ивозможность подключения различных периферийных устройств стали ключевымиособенностями, что позволило создавать сети и интегрировать новые устройства.

4. **Четвертоепоколение ЭВМ (с 1971 года по настоящее время)**:

· Использованиемикропроцессоров, больших и сверхбольших интегральных схем.

· Развитиеперсональных компьютеров и суперкомпьютеров. Современные ПК стали компактными,высокопроизводительными и доступными для массового пользователя.

· Примеромсуперкомпьютера является El Capitan, который выполняет 1,742 квинтиллионаопераций в секунду (экзафлопс).

**14. Принципыпостроения компьютера, сформулированные Дж. фон Нейманом:**

· Программа и данные хранятся в единой памяти.

· Управление компьютером осуществляется с помощью инструкций,выполняемых последовательно.

· Все операции делятся на арифметические и логические, выполняемыецентральным процессором.

· Использованиепамяти с произвольным доступом (RAM) для хранения временных данных.

**Лекция 4**

**15.Последовательность действий по созданию ЦП:**

1. **Изготовлениекристалла**:

· Выращиваниемонокристалла кремния цилиндрической формы из особо чистого кварцевого песка.

· Разрезаниекристалла на пластины толщиной около 0,9мм и их полировка до зеркальногосостояния

2. **Созданиечипов**:

· На пластинахсоздаются интегральные схемы методом фотолитографии и травления в специальныхчистых помещениях.

· Происходитповторная чистка пластин перед тестированием.

3. **Тестирование**:

· Выборочноетестирование процессоров для проверки их функциональности с использованиемспециального оборудования.

4. **Корпусировка**:

· Готовые пластиныразрезаются на отдельные процессорные ядра, которые помещаются в корпуса сконтактами для установки на материнскую плату.

**16. Основныепараметры ЦП:**

· **Тактовая частота**: определяетколичество элементарных операций, выполняемых процессором за единицу времени,измеряется в ГГц (1 Герц соответствует выполнению одной операции за однусекунду).Современные компьютеры заряжены процессорами, тактовая частота которыхколеблется от 1 до 4 ГГц.

· **Разрядность**: определяетколичество бит данных, которые процессор может обработать за один такт.

· **Внутренняя разрядность процессора**определяет, какое количество битов он может обрабатывать одновременно привыполнении арифметических операций.

· **Внешняя разрядность процессора** определяетсколько битов одновременно он может принимать или передавать во внешниеустройства.

· Современные процессоры семейства Intelявляются 32- и 64-разрядными.

· **Рабочее напряжение**: обычно непревышает 3 В и обеспечивается мат.

платой.

· **Кэш-память**: память внутрипроцессора для хранения данных, к которым часто обращается процессор. Кэшбывает трех уровней (L1, L2, L3), различающихся скоростью и объемом.

· Самой быстрой памятью является кэш-память первого уровня(L1-cache), она является частью процессора, т.к. расположена на одном с нимкристалле и входит в состав функциональных блоков, без неё процессор несможет функционировать. Память L1 работает на частоте процессора, объемэтой памяти обычно невелик – не более 64Кб.

· Второй побыстродействию является L2-cache (в отличие от L1 ее можно отключить ссохранением работоспособности процессора), память L2 обычно расположена наотдельном кристалле, но в границах процессора. Объем L2 от 128Кб до 1–4Мб.

· L3-cache наименее быстродействующий и обычновыполняется на отдельных быстродействующих микросхемах с расположением наматеринской плате и имеет объем один и больше Мбайт. L3-cache значительнобыстрее чем оперативная память.

**17. Назначениечипсета:** Чипсет представляет собой набор микросхем системной логики,которые обеспечивают взаимодействие между компонентами ПК. Он состоит из двухмикросхем:

**Микросхемасеверного моста** обеспечивает работу с наиболее скоростными подсистемами. Он содержит: контроллер системнойшины, посредством которого происходит взаимодействие с процессором; контроллерпамяти, осуществляющий работу с системной памятью; контроллер графической шины,обеспечивающий взаимодействие с графической подсистемой; контроллер шины связис южным мостом. Частота работы этой микросхемы равна тактовой частотематеринской платы. Современные North Bridge работают на высоких тактовыхчастотах и поэтому дополнительно оборудованы устройствами охлаждения.

**Южный мост** обеспечиваетработу с более медленными компонентами системы и периферийными устройствами. Он содержит: контроллер, обеспечивающийработу с внутренними накопителями, вчастности с винчестерами и оптическими дисководами; USB-контроллеры;контроллер,

который поддерживает работу внешних портов.

**Лекция 5**

**18. Основные шины:**

· **Шина данных**: передает данныемежду оперативной памятью и процессором.

· **Адресная шина**: передает адресаячеек памяти или устройств ввода-вывода.

· **Командная шина**:передает команды процессору для выполнения операций. Команды представлены ввиде байтов. Простые команды вкладываются в один байт, но есть и такие команды,для которых нужно два, три и больше байта.

**19. Устройствапамяти. ПЗУ и CMOS:**

· **ПЗУ (постоянное запоминающее устройство)**: содержит базовуюсистему ввода-вывода (BIOS), обеспечивающую начальную загрузку и тестированиесистемы при включении. Основное назначение программ этого пакета состоит в том,чтобы проверить состав и работоспособность компьютерной системы и обеспечить взаимодействиес клавиатурой, монитором, жестким диском.

· **CMOS**: разновидностьПЗУ. Это память с невысоким быстродействием и минимальным электропотреблениемот батарейки 3,6 V.

· Используется для хранения информации оконфигурации и составе оборудования компьютера, а также о режимах егоработы(объем оперативной памяти, количество и тип гибких дисков, характеристикижестких дисков, порядок загрузки, энергосбережения, использование системных ивстроенных контроллеров и т.д.).

· *В CMOS-памяти хранится информация о текущих показаниях часов иконфигурации компьютера*

· Содержимое CMOS изменяется специальной программойsetup, находящейся в BIOS.

**20. Устройствопамяти. Динамическая и статическая память:**

· **Динамическая память (DRAM)**: представляютсобой микроконденсаторы. Это наиболее распространенный и экономичный

видпамяти. Из-за переходных процессов запись данных происходит сравнительномедленно, к тому же заряды достаточно быстро (сотые доли секунды) рассеиваютсяв пространстве, что вызывает необходимость постоянной подзарядки  ячеек памяти но дешевле.

· **Статическаяпамять (SRAM)**: можно представить как электронные микроэлементы – триггеры,состоящее из нескольких транзисторов. В триггере хранится состояние  включен-выключен, что обеспечивает болеевысокое быстродействие.

· быстрее, но дорожеи сложнее в производстве, используется в качестве кэш-памяти.

**21. Память спроизвольным доступом (RAM):** Информация в НЖМД записывается на жёсткиеалюминиевые
или стеклянные пластины, покрытые слоемферромагнитного материала, чаще всего диоксида хрома.

С целью адресации пространства  поверхности пластин диска делятся на дорожки (track)–концентрические кольцевые области. Каждая дорожка делится наравные отрезки – секторы (sector).

Цилиндр (cylinder)– совокупность дорожек, равноотстоящих от центра, на всех рабочих поверхностях пластин (platter) жёсткого диска.
Номер головки (head) задает используемую рабочую поверхность (то есть конкретную дорожку из цилиндра).

**22. SSD.Преимущества и недостатки:**

· **Преимущества**: высокая скоростьчтения и записи, низкое энергопотребление и как результат меньшее нагреваниесистемы, отсутствие движущихся частей, что делает SSD более устойчивыми кмеханическим повреждениям, не требует дефрагмантации, полное отсутсвие шума.

· **Недостатки**: высокаястоимость, ограниченный ресурс записи, риск потери данных при внезапномотключении питания.

**23. Контроллеры:**это электронное устройство, предназначенное для подключения к магистраликомпьютера разных по принципу действия, интерфейсу и конструктивному исполнениюпериферийных устройств. **Контроллеры** служат дляуменьшения нагрузки на центральный процессор и повышают общуюпроизводительность системы. Значение контроллеров состоит в том, что ониосвобождают процессор от наиболее медленных функций ввода/вывода информации.Операционная система практически всегда имеет дело с контроллером, а не с самимустройством.

**24. Разновидностивидеокарты. Их преимущества и недостатки:**

· **Интегрированные видеокарты**: блок обработкиграфики не имеет своего процессора и использует общую оперативную память ипроцессор ПК, что снижает производительность, но они дешевле и потребляютменьше энергии, что повышает общий срок работы батареи.

· **Дискретныевидеокарты**: имеют собственный графический процессор и память, разгружаетсяцентральный процессор, обеспечивая высокую производительность всей системы, нотребуют мощные блоки питания, больше энергии и дополнительного охлаждения.

25. Мониторыжидкокристаллические и плазменные.

1. Жидкокристаллический монитор (LCD):

• Состоит из двух стеклянных или пластиковыхпластин.

• Между пластинами находятся тонкопленочныйтранзистор, цветной фильтр и суспензия с кристаллами.

• Кристаллы расположены параллельно,позволяя свету проходить.

• Тонкопленочный транзистор создаетэлектрическое поле, изменяющее положение кристаллов и блокирующее свет.

2. Плазменные мониторы:

• Основаны на плазменной панели с инертнымигазами (ксенон и неон) в микрокамерах между стеклами.

• Содержат два типа электродов: управляющиеи прозрачные сканирующие.

• При зарядке электродов происходитионизация газа, что вызывает электрический разряд и испускание ультрафиолетовыхфотонов.

•Ультрафиолетовые фотоны заставляют светиться фосфорное покрытие микрокамер,создавая видимый свет.

## 26. Уровнипрограммного обеспечения.

Базовый уровень является низшим уровнем программного обеспечения.Отвечает за взаимодействие с базовыми аппаратными средствами.
Содержит базовую системуввода-вывода BIOS, записанную в ПЗУ на этапе производства.
Системный уровень - является переходным.
Обеспечиваетвзаимодействие программ с базовым уровнем и аппаратным обеспечением. Влияет наэксплуатационные показатели вычислительной системы.
Другой класспрограмм системного уровня отвечает за взаимодействие с пользователем.
Служебный уровень - программы этого уровнявзаимодействуют как с программами базового уровня, так и с программамисистемного уровня. Назначение служебных программ (утилит) состоит вавтоматизации работ по проверке и настройки компьютерной системы, а также дляулучшения функций системных программ.
Прикладнойуровень - состоитиз прикладных программ для выполнения конкретных задач. Тесная взаимосвязь ссистемным программным обеспечением.

## 27. Операционнаясистема: основные принципы, эксплуатационные требования и функции современных ОС.

Операционнаясистема - комплекс программ, предназначенных для управления ресурсамикомпьютера и организации взаимодействия с пользователем.
Типы ОС:
- Десктопные
- Мобильные
- Серверные

Основные принципы

- управлениересурсами
- поддержкапрограммного обеспечения
- управлениефайлами и папками

Эксплуатационные требования

- расширяемость
- переносимость
- совместимость
- надежностьи отказоустойчивость.
- безопасность(аутентификация, авторизация, аудит)
- производительность.

Функции современных ОС

- Управлениеконфигурацией ПК
- Управлениепроцессами, потоками и заданиями.
- Управлениепамятью.
- Обеспечение информационнойбезопасностью.
- Управлениеподсистемой ввода-вывода.
- Управлениевнешней памятью.
- Управлениефайловой системой.
- Поддержкасетей

## 28. Основныекомпоненты операционных систем

- ядрооперационной системы –основа ОС и связующее звено между аппаратным ипрограммным обеспечением компьютера, выполняет множество сложных задач,обеспечивая работоспособность ОС и управляя ресурсами компьютера.
- модули,выполняющие вспомогательные функции ОС

Состав ядра

- Менеджерпамяти.
- Планировщикзадач.
- Драйверыустройств. Драйверы устройств – это специальные программы, которыепозволяют ОС взаимодействовать с устройствами компьютера.

- Системныевызовы.
- Модульбезопасности.
- Управлениепроцессами и потоками.
- Управлениефайловой системой.

Вспомогательные модули

- утилиты -программы, решающие отдельные задачи управления и сопровождения вычислительнойсистемы
- системныеобрабатывающие программы
- программыпредоставления пользователю дополнительных услуг
- библиотекипроцедур и функций различного назначения, облегчающие разработкупользовательских приложений

## 29. Файловаясистема

Файловая система - это средстводля организации хранения файлов на носителях данных, таких как жесткие диски,SSD-накопители, флеш-диски и др.

Она управляет созданием,удалением, чтением и записью файлов, а также предоставляет структуру дляорганизации файлов.

Связывает носительинформации с API для доступа к файлам.

Файловая система организуетсяна уровне **разделов** или **томов** – областей устройства для хранениянайловой системы.

Процесс создания файловойсистемы называется **форматированием**, в ходе которого определяются правилахранения информации и структура каталогов.

**Каталоги**:используются для организации файлов в иерархическую структуру, позволяягруппировать файлы по критериям.

Путьсостоит из имени тома, имен каталогов, имени файла, разделенных специальнымисимволами.

Пример: C:\Учебныематериалы\БД\cars.txt

Имя тома Имя каталога Имякаталога Имя файла

## 30. Прикладнойуровень ПО.

Прикладное программное обеспечение:комплекс программ для выполнения конкретных задач, доступность которых зависитот операционной системы и взаимодействия человек-программа-оборудование.

1. Текстовые редакторы.

Основные функции - это ввод иредактирование текстовых данных.
Пример: Блокнот (NotePad)

2. Текстовые процессоры.

Функции: форматированиетекста, взаимодействие с графикой и таблицами, автоматизация редактирования.
Примеры: Microsoft Office Word, LibreOfficeWriter, Google Docs

3. Электронные таблицы.

Предоставляют средства дляхранения и обработки данных; автоматическое обновление ячеек при измененииформул.
Примеры: Microsoft Office Excel, LibreOfficeCalc, Google Sheets.

4. Графические редакторы

- растровыередакторы (Adobe Photoshop, GIMP);
- векторныередакторы (CorelDraw, Inkscape);
- 3-D редакторы(трехмерная графика) (FreeCad, Autodesk AutoCad)

5. Системы управления базами данных(СУБД).

Организация большихмассивов данных в табличные структуры.
Основныефункции СУБД:

- созданиебазы;
- заполнениеданными;
- доступ, поиск и фильтрация.
- Пример: PostgreSQL; Microsoft Office Access

6. Системы автоматизированногопроектирования (CAD-системы).

- Предназначеныдля автоматизации проектно-конструкторских работ. Применяются в машиностроении,приборостроении,

архитектуре.
Пример:Компас 3D;SolidWorks; nanoCAD
7. Браузеры (средства просмотраWeb-документов)
Программныесредства предназначены для просмотра электронных документов, созданных вформате HTML.
Пример: Google Chrome;Mozilla Firefox; Microsoft Edge.

## 31. Технологиивиртуализации и контейнеризации.

Технологиивиртуализации
Виртуализация возникла в60-х годах как способ расширения оперативной памяти. Разработана специалистамикомпаний, таких как GE, Bell Labs и IBM.
Виды виртуализации:серверная, сетевая, десктоп-виртуализация, виртуализация памяти и приложений.Наиболее активно развивается серверная виртуализация.
Виртуализация – этотехнология, с помощью которой можно создавать виртуальные версии физическихресурсов.
Главной составляющей технологиивиртуализации является виртуальная машина (ВМ).
ВМ – это изолированнаяпрограммная среда, эмулирующая аппаратное обеспечение. Она позволяет превращатьодин физический сервер в несколько виртуальных.
**Гипервизор**:Это программное решение, которое управляет несколькими ВМ, позволяя им работатьпараллельно и эффективно делить ресурсы сервера.
**Преимущества виртуализации**:

- **Изоляция**: Сбой в одной ВМ не влияетна другие.
- **Гибкость**: Быстрый перенос ВМ междусерверами
- **Управляемость**: Оптимальноераспределение ресурсов.
- **Оптимизация**: Эффективноеиспользование ресурсов физического сервера.
- **Обслуживание**: Централизованноеобновление ВМ.

**Недостатки виртуализации**:

- **Высокие издержки**: Необходимость вмощном и дорогом оборудовании.
- **Сложность управления**
- **Задержки при миграции**
- **Сниженная производительность**
- **Зависимостьот гипервизора**

Технологииконтейнеризации
Контейнеризация – этотехнология изоляции приложений и их зависимостей путём упаковки в единоеисполняемое окружение – контейнер.
Первые контейнеры появились в 2004 году в Solaris 10.
В 2006 году Google разработала систему ProcessContainers, позже переименованную в control groups и включенную в ядро Linux.
В 2013 году был запущенпроект Docker, который стал основоположником экосистемы для управленияконтейнерами.
Docker позволяет упаковать код и зависимости приложения длябыстрого запуска.
Kubernetes управляетбольшим количеством контейнеров и поддерживает сложные приложения.
Контейнеры служат связующим звеном между системнымядром и приложениями, разделяя ресурсы, но обеспечивая изоляцию безнеобходимости эмуляции ОС.

**Преимуществаконтейнеризации**

- **Легковесность**: меньше ресурсов,быстрое развертывание.
- **Изоляция**: повышенная безопасность истабильность
- **Переносимость**: легкость вперемещении между средами.
- **Масштабируемость**
- **Управляемость**: легкость в созданиии удалении контейнеров.

**Недостатки**:

- **Ограниченность**: необходимостьиспользования одной ОС
- **Меньшая изоляция**: контейнеры менееизолированы, чем виртуальные машины.

- **Уязвимость** общей ОС.
- **Сложность** настройки и управления.

## 32. Компьютерные сети. Классификация по размеру охваченной территории.

Компьютернаясеть – совокупность компьютеров, соединенных с помощью каналов связи исредств коммутации в единую систему для обмена сообщениями и доступапользователей к программным, техническим, информационным и организационнымресурсам сети.

1. **BAN (Body AreaNetwork)**:
• Набор устройств,встроенных или закрепленных на теле человека.
• Малые размеры и низкое энергопотребление.
• Датчики передаютинформацию на смартфоны для мониторинга здоровья пациента.
• Радиус действия 1-2 метра.

2. **PAN (PersonalArea Network)**:
• Сети для взаимодействияустройств одного владельца (например, ПК с мышкой).
• Используют технологииWi-Fi или Bluetooth.

3. **LAN (Local AreaNetwork)**:
• Локальные сети срадиусом действия до 1 километра.
• Включают ПК, смартфоны,принтеры и промежуточные узлы (коммутаторы, роутеры).
• Основные функции:оптимизация работы, возможность общения, удаленное администрирование, экономияи безопасность обмена информацией.

4. **CAN (Campus AreaNetwork)**:
• Объединяет нескольколокальных сетей (например, в учебном заведении).
• Радиус действия от 1 до5 километров.

5. **MAN (MetropolitanArea Network)**:
• Сети масштаба города,управляемые провайдерами.
• Радиус действия 10-15километров.

6. **WAN (Wide AreaNetwork)**:
• Глобальные сети, обычноне принадлежащие отдельным лицам или компаниям.
• Пример – Интернет,включающий сети провайдеров и крупных компаний.

## 33. Компьютерные сети. Классификация по типу функционального назначения.

1. **Точка-точка**:
• Простейший вид сети, соединяющий двакомпьютера напрямую через коммуникативное оборудование.
• Достоинства: простота и дешевизна.
• Недостаток: ограничение на соединениетолько двух компьютеров.
• Используется для быстрой передачиинформации между двумя компьютерами
2. **Технология"Клиент-сервер"**:
• Архитектура, где устройства выступают какклиенты (запрашивающие) или серверы (отвечающие на запросы).
• Клиентом обычно является ПК, сервером –машина, предоставляющая ресурсы.
• Оба термина могут относиться как кфизическим устройствам, так и к программному обеспечению.
3. **Сеть с выделенным сервером**:
• Локальная сеть с централизованнымуправлением через один или несколько серверов.
• Рабочие станции (клиенты) обращаются кресурсам сети через серверы.

4. **Одноранговая сеть**:

- Сети, основанные на равноправииучастников, без выделенных серверов.
- Каждый узел является и клиентом, исервером.
- Позволяет сохранять работоспособность сетипри любом количестве узлов.

## 34. Компьютерные сети. Классификация по типу сетевой технологии.

Топология "Шина":

**Описание**:Общий кабель, к которому подключены все рабочие станции.

**Достоинства**:

- Быстрое время установки.
- Низкая стоимость (меньшая длина кабеля именьше устройств).
- Простота настройки.
- Неисправность одной станции не влияет навсю сеть.

**Недостатки**:

- Неполадки (обрыв кабеля) блокируют всюсеть.
- Сложности в выявлении неисправностей.
- Падение производительности при добавлениииновых станций.

Топология "Звезда":

**Описание**:Все компьютеры подключены к центральному узлу.

**Достоинства**:

- Неисправность одной станции не влияет насеть.
- Высокая производительность.
- Низкая стоимость.

**Недостатки**:

- Выход из строя центрального узла блокируетсеть.
- Требуется больше кабеля для прокладки.
- Ограниченное число рабочих станций взависимости от портов узла.

Топология "Кольцо":

**Описание**:Каждый компьютер соединен с двумя другими, информация передается по кругу.

**Достоинства**:

- Простота установки.
- Минимальное количество дополнительногооборудования.
- Устойчивость к нагрузкам без значительногопадения скорости.

**Недостатки**:

- Неисправность одной станции влияет на всюсеть.
- Сложности в конфигурировании и настройке.
- Добавление/удаление станции требуетостановки сети.

Топология "Ячеистая":

**Описание**:Рабочие станции соединены друг с другом и могут выполнять роль коммутатора.

**Достоинства**:

- Высокая отказоустойчивость благодарямножеству связей.

**Недостатки**:

- Сложная настройка.

Топология "Дерево":

**Описание**:Узлы более высокого уровня связаны с узлами более низкого уровня, образуякомбинацию звезд.

**Достоинства**:

- Легкость в увеличении и контроле сети(поиск неисправностей)

**Недостатки**:

- Выход из строя родительского узла блокируетдочерние узлы и может затруднить доступ к сети.

## 35. Устройство сети Интернет. Протоколы TCP и IP. Адресация в сети.

**Интернет**-комплекслокальных сетей и автономных компьютеров, соединенных средствами связи ипрограммным обеспечением, работающим на основе протоколов TCP и IP.
В Интернетесуществует единая система адресации, которая помогает компьютерам найти другдруга в процессе обмена информацией. Интернет работаетпо протоколу TCP/IP, который отвечает за физическую пересылкусообщений между компьютерами в сети Интернет.
Протокол- это правило передачи информации в Сети.

- **TCP/IP**: Протоколы, отвечающие запередачу данных между компьютерами в сети.

**IP-протокол**: Набор правил для доставкиданных с использованием уникальных IP-адресов отправителя и получателя.
Программные модули протокола IP

- **IPv4**:Позволяет более четырехмиллиардов уникальных IP-адресов (формат: четыре десятичных числа от 1 до 255)
- **IPv6**: Расширяет длину IP-адреса до128 бит, увеличивая число доступных идентификаторов практически добесконечности.

**TCP**:Обеспечивает пересылку больших объемов информации, разбивая данные на части,нумеруя их и оборачивая в TCP-конверт, который помещается в IP-конверт дляпередачи по сети.

## 36. Устройство сети Интернет. Маршрутизаторы,шлюзы.

- **Опорная сетьИнтернета**: Состоит из узловых компьютеров (серверов или хостов) и каналовсвязи между ними.

Маршрутизаторы: Установлены на каждомузле, определяют направление передачи TCP-пакетов по IP-адресу.

- **Функциимаршрутизаторов**:
  - Сканируют соседние серверы и общаются с ихмаршрутизаторами.
  - Учитывают состояние соседей для оптимальнойпереправки пакетов.

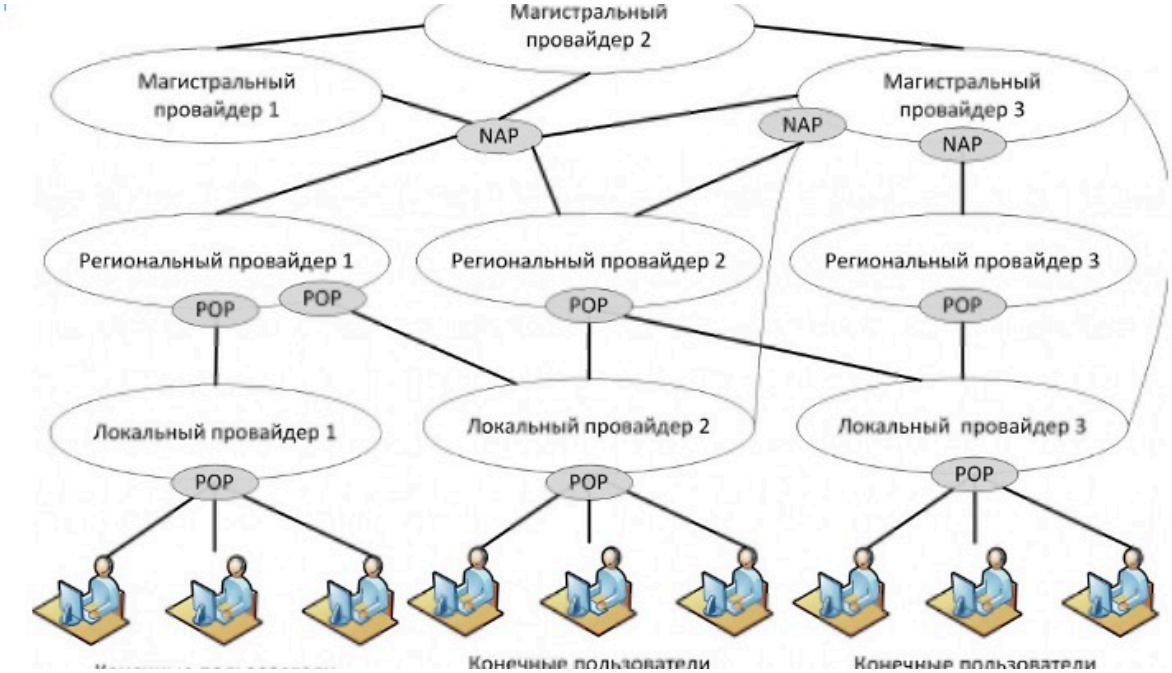
Шлюзы: Подключают локальные сети,работающие на других протоколах, к Интернету, преобразуя данные между форматамилокальной сети и Интернетом.

**37. Устройство сети Интернет. Провайдеры. Системадоменных имен.**

ПРОВАЙДЕРЫ

Пользователи подключаются к сети благодаря провайдерам – организациям, оказывающим услугидоступа в Интернет и другие услуги, связанные с Интернетом, например выделение дискового пространства для хранения и обеспечения работы сайтов(хостинг); поддержка работы почтовых ящиков или виртуального почтового сервера;содержание линий связи, то есть поддержание их в рабочем состоянии, и другие.

Магистральные провайдеры имеют магистральные каналы связи в собственности,а региональные арендуют у них каналы связи. Взаимоотношения междупровайдерами осуществляются на основе пиринговых соглашений.




СИСТЕМА ДОМЕННЫХ ИМЕН

Чтобы не запоминать числовые адреса была введена система доменныхимен Domain Name System (DNS). Она обеспечивает соответствие числовому IP-адресу каждого компьютера уникального доменногоимени, которое обычно

состоит из двух-четырех слов, разделенных точками(доменов).
Доменноеимя читается слева направо. Самое правое слово в доменном имени является доменом верхнего, или первого, уровня. Существует два типа доменов верхнегоуровня: географические (двухбуквенные – указывают на страну, в которойнаходится узел) и административные (трехбуквенные) – указывают на тип илипрофиль организации. Каждой стране мира выделен свой географический домен.
Таблицысоответствия DNS-адресов IP-адресам размещают на специальных DNS-серверах,подключаемых к Интернету. Если устройство не знает IP-адреса компьютера,с которым собирается установить связь, а имеет только символьный, то онозапрашивает DNS-сервер, предоставляя ему текстовый вариант, и получает в ответIP-адрес нужного адресата.

**5.255.255.60 -** IP-адрес

**yandex.ru** - символьный DNS-адрес

**38. Устройство сети Интернет. Сервисы сети.**

ВСЕМИРНАЯ ПАУТИНА:
Всемирная Паутина – это распределенная система, предоставляющая доступ к связаннымежду собой документам, расположенным на различных компьютерах, подключенных кИнтернету.
Всемирная паутина использует технологию гипертекста, в которой документы связаны междусобой гиперссылками. Документы, содержащие гиперссылки, называютсявеб-страницами, а серверы Интернета, их хранящие, – веб-серверами.
Передача веб-страниц по сети Интернет осуществляется с помощью протокола пересылки гипертекста HypertextTransferProtocol (HTTP).Посредством HTTP можно передавать любую информацию, в том числеизображения, звук, видео.
Всемирная паутина работает по принципу клиент-сервер. Веб-сервер принимает HTTP-запросы от клиентов, которыми обычно являются веб-браузеры, и выдает HTTP-ответы.

ЭЛЕКТРОННАЯ ПОЧТА:
Электронное письмо может содержать не только текстовое сообщение, но

и вложенные файлы(программы, графику, звук и т. д.). Кроме того, электронная почта позволяет посылать сообщение сразу нескольким абонентам, пересылать письма на другиеадреса и т. д.
Перваячасть электронного почтового адреса имеет произвольный характер и задаетсясамим пользователем при регистрации почтового ящика. Вторая часть является именем почтового сервера Интернета, на котором пользователь зарегистрировал свой почтовый ящик. Части адреса разделяются символом @.
Процесс передачи сообщения начинается с доставки сообщения в пользовательский почтовый ящик на удаленном почтовом сервере. Почтовый сервер сразу же отправит это сообщение через систему почтовых серверов Интернета на почтовый сервер получателя в его почтовый ящик.

ПОТОКОВОЕ МЕДИА:
Потоковоемедиа представляет собой Интернет-контент (аудио-, видео- или аудиовидеофайлы),который пользователь может смотреть или слушать как непрерывный поток в режимереального времени, не дожидаясь окончания загрузки всего файла на персональныйкомпьютер.
Потоковое медиа пересылается непрерывнымпотоком в виде последовательности сжатых пакетов и проигрывается по мере того,как передается на компьютер получателя.
Еслимедиа не потоковое, то посмотреть файл можно только после его полной загрузкина жесткий диск.

СЕРВИС ПЕРЕДАЧИ ФАЙЛОВ(FTP):
FTP-сайт- это компьютер в сети Интернет, на котором ведется файловый архив. FTP-серверэто программа, работающая на таком компьютере, и которая обеспечивает ведениеархива и обработку запросов к архиву. FTP-клиенты – это программы, используемыедля доступа к FTP-сайтам и открывающие папки на них (FTP-папки) какпростые папки на компьютере пользователя.
Благодарясервису FTP пользователи могут пересылать (копировать, передавать) файлы вИнтернете с удаленного компьютера на локальный и с

локальногона удаленный. В отличие от веб-серверов, которые предоставляют информациютолько для чтения, FTP-серверы позволяют пользователям не только скачиватьинформацию, но и добавлять ее на сервер.

ПОИСКОВЫЕ СИСТЕМЫ:

Поисковые системы решают задачу поиска по большим объемам неструктурированной информации.

В Интернете находится несколько тысяч поисковых систем. Это программно-аппаратные комплексы,предназначенные для осуществления поиска в сети и реагирующие на запрос пользователя выдачей списка ссылок на источники информации в порядке релевантности, сервис, который помогает пользователям быстро найти необходимые сведения.

Поисковая система обычно имеет поле для ввода ключевых слов, по которым она находит документы, содержащие эти ключевые слова.

Существует 2 основных типа поисковых систем : индексные и классификационные (каталоговые).

Индексные поисковые системы работая в автоматическом режиме обновления своей информации, просматривают в Интернет содержимое серверов, индексируя информацию, содержащуюся в них и внося информацию о расположении слов на страницах сайтов в свои базы данных.

Каталоговые системы поиска содержат тематически структурированный каталог серверов,и чаще всего пополняются вручную.

ВЭБ - ФОРУМЫ:

Веб-форум – сервисдля общения между пользователями Интернета (более двух участников) на одну темуили на несколько тем (зависит от специализации форума).

Сутьработы форума заключается в создании пользователями (посетителями форума) своихТем с их последующим обсуждением, путём размещения сообщений внутри этих тем. Форумы могут существовать как автономно, без привязки к какому-либо сайту, так и быть частью веб-порталов. Форум отличается от чата разделением обсуждаемых тем и возможностью

общения не в реальном времени

СЕРВИС IMS (служба мгновенных сообщений):
Спомощью IMS кроме текстовых сообщений можно передавать звуковые сигналы, картинки, видео, файлы, а также, например,производить такие действия как совместное рисование или игры. Для этого необходима клиентская программа, так называемый мессенджер. Ключевой особенностью многих клиентов обмена мгновенными сообщениями является возможность видеть, находится ли друг или коллега в сети и подключается ли он через выбранную службу – эта возможность известна как присутствие. Чат-боты –это вариант обмена мгновенными сообщениями. Они имитируют взаимодействие с пользователем при обмене мгновенными сообщениями, но вместо того, чтобы разговаривать с другим человеком, пользователь разговаривает с автоматизированной программой. Чат-боты используются для автоматизации простых запросов обслуживания клиентов.
TELNET:
Это сервис, который позволяет осуществлять удаленный доступ в другую вычислительную систему. Информация вводится на одном компьютере, передается на обработку другому, а результаты возвращаются на первый.
Telnet позволяет работать так, будто клавиатура одного компьютера подключена непосредственнок другому, то есть дает возможность пользоваться всеми средствами, которые удаленный компьютер предоставляет локальным терминалам, входить в систему,выполнять команды или получать доступ к множеству специальных сервисныхсредств.

**39. Основные функции СУБД. Функции современных БД**

ОСНОВНЫЕ ФУЕКЦИИ СУБД.

- Определение данных - определить, какая именно информация будет храниться в базе данных,задать свойства данных, их тип (например, число цифр или символов), а также указать, как эти данные связаны между собой.
- Обработка данных с обязательным условием обеспечения целостности БД (база данных содержитполную и непротиворечивую информацию)
- Управление данными - определение правил индивидуального и

коллективного доступа.

ФУНКЦИИ СОВРЕМЕННЫХ БД

•описаниеданных, их структуры;

•первичныйввод информации;

•удалениеустаревшей информации;

•корректировкаданных;

•упорядочение(сортировка) данных;

•поискинформации по некоторым признакам;

•подготовкаи генерация отчетов;

•защитаинформации;

•резервноесохранение и восстановление БД;

•дружественныйпользовательский интерфейс.

## 40. Иерархические и сетевые БД.

ИЕРАРХИЧЕСКИЕ БД

В иерархических базах данных каждая запись имеет одного «родителя». Это создаёт древовидную структуру, в которой записи классифицируются по их отношениям с цепочкой родительских записей.


СЕТЕВЫЕ БД

Записимогутиметь более одного родителя. А значит, можно моделировать сложные отношения.

## 41. NoSQL базы данных

NOSQL

Базаданных **NoSQL**,или нереляционнаябаза данных, дает возможность хранить и обрабатывать неструктурированные или слабо структурированные данные

Документно-ориентированные базыданных (NoSQL)

Единицей хранения является документ (который может быть в формате json,или xml,или в каком-нибудь еще формате). Удобство таких баз в том, что в них быстро илегко записывать любые типы данных, при этом эти данные не обязаны обладать четкой структурой. Минус таких баз в том, что данные в них неудобно анализировать.

Пример:MongoDB

Key-value базыданных (NoSQL)

В базах данных «ключ-значение» для хранения информации вы предоставляете ключ иобъект данных, который нужно сохранить.

Пример:**Redis - The Real-time Data Platform**Пример:**Redis - The Real-time Data Platform**

**42. Логическая модель БД, ее основные компоненты.Первичные и внешние ключи.**

Логическая модель базы данных – это абстрактное представление структуры данных, которое используется для планирования и проектирования баз данных.

Она описывает,как данные будут организованы и как они будут взаимодействовать друг с другом,не привязываясь к конкретной системе управления базами данных (СУБД).

Логическая модель помогает разработчикам и аналитикам понять требования к данным и спроектировать эффективную и масштабируемую базу данных.

ОСНОВНЫЕ КОМПОНЕНТЫ ЛОГИЧЕСКОЙ МОДЕЛИ:

Сущности– это объекты, которые мы хотим хранить в базе данных.

Каждаясущностьимеет атрибуты, которые описывают еёхарактеристики.

Например,

сущность"Студент"

атрибуты "Имя","Фамилия", "Дата рождения",  "Номерстуденческого билета».

Связи между сущностями описывают как эти сущности будут взаимодействовать друг с другом. Они могут быть трех типов:

-один к одному (1:1)

-один ко многим (1:N)

-многие ко многим (M:N)

Например,связь между сущностями "Группа" и"Студент" можетбыть "Один ко многим",так как один студент может принадлежать только одной группе, но одна группа может включать многих студентов.

ПЕРВИЧНЫЙ И ВНЕШНИЙ КЛЮЧИ.

Первичный ключ –это уникальный идентификатор сущности.

Внешний ключ –это атрибут, который ссылается на первичный ключ другой сущности, создаваясвязь между ними.

Например,в сущности "Студент" первичным ключом можетбыть "Номер студенческого билета", а внешний ключ "Номер группы"будет ссылаться на соответствующий первичныйключ всущности"Группа".

Первичныеи внешние ключи обеспечивают уникальность и целостность данных в базе данных.

Онипомогаютизбежать дублирования данных и обеспечивают правильность связей междусущностями.

**43.  Базовые понятия реляционных баз данных.**

ОСНОВНЫЕ ПОНЯТИЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

- тип данных
- домен
- атрибут
- кортеж
- первичный ключ
- отношение

Реляционная модель предусматривает единственный способ представления

данных – в виде набора двумерных таблиц, которые называют отношениями.
Например,таблица хранит информацию об элементах отношения «студенты».Каждая строка отвечает одной сущности «студент», а каждыйстолбец отвечает одному из атрибутов множеству сущностей «студенты».
Отношения могут хранить не только множества сущностей, но и экземпляры связей.

ПРИМЕР ОТНОШЕНИЯ «СТУДЕНТ»

| Фамилия | Имя | Отчество | Год | № зачетки |
|---|---|---|---|---|
| Иванов | Петр | Сергеевич | 2006 | 12345689 |
| Орлова | Елена | Викторовна | 2006 | 12345735 |
| Седов | Олег | Петрович | 2005 | 12345863 |

Строки – экземпляры сущности
Столбы - атрибуты

Вверхней части таблицы отношения задается перечень наименований атрибутов. Атрибуты выполняют функцию заголовков столбцов и содержательно описывают смысл и назначение элементов данных в соответствующих ячейках.
Наименования отношения и атрибутов этого отношения называют схемой отношения.Схема отношения представляется в виде имени отношения, за которым идут список атрибутов заключенных в круглые скобки.
Пример:Студенты (Фамилия, Имя, Отчество,  Год, №зачетки)
Атрибуты  схемы отношения образуют множество.
Полагается,что с каждым атрибутом ассоциирован определенный домен.т.е.некоторый базовый тип. Значения атрибутов должны

принадлежать соответствующим доменам, определяемым каждым из атрибутов отношения.

Строки отношения,отличные от первой,которая представляет наименования атрибутов, называют кортежами(tuples).Кортеж содержит по одному компоненту для каждого атрибута отношения.

## 44. Язык SQL. Создание и заполнение таблиц. Просмотрведенной информации.

СОЗДАНИЕ ТАБЛИЦЫ

В команде **CREATE** указываем имя таблицы, имена заголовков столбцов,тип данных в столбце, **ширину столбцов** и значения по умолчанию.

```
CREATETABLE Products
(
   Id SERIAL PRIMARY KEY,
   ProductNameVARCHAR(30)NOT NULL,
   ManufacturerVARCHAR(20)NOT NULL,
   ProductCountINTEGER DEFAULT 0,
   PriceNUMERIC
);
```

ЗАПОЛНЕНИЕ ТАБЛИЦЫ

Для занесения в таблицу данных в команде INSERT указываем имя таблицы,названия заголовкови **данные**

```
INSERT INTO Products (ProductName,Manufacturer, ProductCount, Price)
VALUES
(
('iPhone 16', 'Apple', 3, 170000),
('iPhone 15', 'Apple', 2, 90000),
('Galaxy 22', 'Samsung', 4, 46000),
('Galaxy S8 Plus', 'Samsung', 2, 56000),
('Xiaomi 15', 'Xiaomi', 3, 63000);
);
```

**45. Язык SQL. Фильтрация данных. Операторы IN, BEETWIN, LIKE**

**Фильтрация данных:**

```
SELECT * FROM Products
WHERE Manufacturer ='Apple';

SELECT * FROM Products
WHERE Price <39000;

SELECT * FROM Products
WHERE Price *ProductCount > 90000;
```

**Операторы IN:**

WHERE выражение (NOT) IN ( выражение )

Выражение в скобках после IN  определяет наборзначений.

SELECT * FROM Products

WHERE Manufactures IN

Оператор **BETWEEN:**

WHERE выражение [NOT] BETWEENначальное_значение AND конечное_значение

SELECT * FROM Products

WHERE Price BETWEEN 20000 AND50000;

**Оператор LIKE**

WHERE выражение [NOT] LIKE шаблон_строки

WHERE ProductName LIKE 'Galaxy**%**'

WHERE ProductName LIKE 'Galaxy _'

SELECT * FROM Products

WHERE ProductName LIKE 'iPhone**%**';

**Вопрос 46.**

**ORDERBY. Сортировка**

SELECT * FROM Products

ORDER BY ProductCount;

По умолчанию данные сортируются по возрастанию, однако с помощьюоператора **DESC** можнозадать сортировку по убыванию

```
SELECT ProductName, Manufacturer
FROM Products
ORDER BY Manufacturer DESC;
```

***DISTINCT. Выборка уникальных значений***

```
SELECT DISTINCT Manufacturer FROM Products
```

**Вопрос 47.**

**Получение диапазона строк.LIMIT**

***Оператор LIMIT*** позволяет извлечь определенное количествострок:

```
SELECT * FROM Products
ORDER BY ProductName
LIMIT 4;
```

**Получение диапазона строк.OFFSET**

Оператор **OFFSET** позволяетуказать, с какой строки надо начинать выборку.

```
SELECT * FROM Products
ORDER BY ProductName
LIMIT 3 OFFSET 2;
```

**Вопрос 48.**

***Запросы. Агрегатные функции***

- **AVG**: находит среднее значение.
- **COUNT(*)**: находит количество строк в запросе
- **SUM**: находит сумму значений
- **MIN**: находит наименьшее значение
- **MAX**: находит наибольшее значение

```
SELECTAVG(Price)FROMProducts;
SELECTMIN(Price)FROMProducts;
```

**Изменения названиявыходного столбца**

```
SELECT ProductCount AS Title,
Manufacturer,
Price * ProductCount AS TotalSum
FROM Products;
```

**Вопрос 49.**

***Запросы. Группировка данных***

Оператор**group by** определяет, как строкибудут группироваться.

```
SELECT Company, COUNT(*) AS ModelsCount
```

(Company представляет название группы, Count вычисляет количество строк в группе)

```
FROM Products
GROUP BY Company;
```

Оператор **GROUP BY** может выполнять группировку помножеству столбцов

Группировка по количеству товара

```
SELECT Company,ProductCount, COUNT(*) AS ModelsCount
FROM Products
GROUP BY Company, ProductCount;
```

Выражение GROUP BY должно идти послевыражения WHERE, но до выражения ORDER BY:

```
SELECT Company, COUNT(*) AS ModelsCount
FROM Products
WHERE Price > 30000
GROUP BY Company
ORDER BY ModelsCount DESC;
```

**Вопрос 50.**

Синтаксис.

```
SELECT столбцы
FROM таблица1
   [INNER] JOIN таблица2
   ON условие1
   [ [INNER] JOIN таблица3
   ON условие2]
```

```
CREATE TABLE Products
(
   Id SERIAL PRIMARY KEY,
   ProductName VARCHAR(30) NOT NULL,
   Company VARCHAR(20) NOT NULL,
   ProductCount INTEGER DEFAULT 0,
   Price NUMERIC NOT NULL
);
```

```
CREATE TABLE Orders
(
   Id SERIAL PRIMARY KEY,
   ProductId INTEGER NOT NULL REFERENCES Products(Id) ON
DELETE CASCADE,
   CreatedAt DATE NOT NULL,
   ProductCount INTEGER DEFAULT 1,
   Price NUMERIC NOT NULL
);
SELECT Orders.CreatedAt, Orders.ProductCount, ProductName
FROM Products
JOIN Orders  ON Products.Id = Orders.ProductId
```

**Лекция 12.**

**Вопрос 51.**

Базисные управляющиеконструкции

Блок-схемы используются на этапе проектирования и записываются стакой степенью детализации, чтобы на следующем этапе - этапе кодирования на выбранном языкепрограммирования - можно было бы каждому блоку однозначно поставить всоответствие нужную последовательность инструкций выбранного языкапрограммирования.

С использованиемосновных элементов строится набор базисных управляющих конструкций:

- последовательность;
- ветвление и выбор
- циклы с предусловием и постусловием

**Лекция 13.**

**Вопрос 52.**

**Пузырьковая сортировкамассива**

Впузырьковой сортировке каждый элемент сравнивается со следующим. Если два такихэлемента не стоят в нужном порядке, то они меняются между собой местами. Вконце каждого прохода наибольший или наименьший элемент ставится в конецсписка.

**Вопрос 53.**

**Сортировка выбором.**

Ищем наименьшее значение в массиве и ставим его на позицию, откуданачали проход. Потом двигаемся на следующую позицию Зеленымотмечается наименьший элемент в подмассиве – он ставится в начало списка.

**Вопрос 54.**

**Сортировка вставками**

Начинаем со второй позиции

Число 12 больше 5 – элементы меняются местами

**Вопрос 55.**

**Сортировка Шелла**

Алгоритм включает в себясортировку вставками. Исходный массив размером N разбивается наподмассивы с шагом N/2. Подмассивы сортируются вставками. Затем вновьразбиваются, но уже с шагом равным N/4. Цикл повторяется. Производимцелочисленное деление шага на два каждую итерацию. Когда шаг становится равен1, массив просто сортируется вставками.

**Вопрос 56.**

**Быстрая сортировка**

Массивразделяется на подмассивы, которые сортируются и затем сливаются в один.

В первуюочередь выбирается опорный элемент. Опорным может быть любой элемент. Всезначения больше опорного элемента ставятся после него, меньше – перед ним.

В полученныхмассивах также выбираем опорный элемент и разделяем по нему.

Чтобырасположить элементы большие – справа от опорного элемента, а меньшие – слева,двигаемся от начала списка. Если число будет больше опорного, то оно ставитсяна его место, а сам опорный на место перед ним.

Сложность влучшем случае: **O(n*logn).**

Сложность вхудшем случае: **O($n^2$)**.

**Вопрос 57.**

**Сортировка кучей**

Алгоритм сортировки кучей:

- Формируем бинарное дерево из массива.
- Расставляем узлы в дереве так, чтобы получилась куча.
- Верхний элемент помещаем в конец массива.

Возвращаемся на шаг 2, пока куча не опустеет.

Обращаться к дочерним узлам можно, зная, чтодочерние элементы i-го элемента находятся на позициях **2*i + 1** (левый узел) и **2*i + 2** (правый узел).

Сложность алгоритма в любом случае: **O(n*logn)**.

**Вопрос 58.**

**Сортировка слиянием**

Разделяем исходный массив на два равныхподмассива. Повторяем сортировку слиянием для этих двух подмассивови объединяем обратно.

Цикл деленияповторяется, пока не останется по одному элементу в массиве. Затем объединяем,пока не образуем полный список.

Алгоритмсортировки состоит из четырех этапов:

- Найти середину массива.
- Сортировать массив от начала до середины.
- Сортировать массив от середины до конца.
- Объединить массив.

Алгоритмобъединения массивов:

- Циклично проходим по двум массивам..
- В объединяемый ставим тот элемент, что меньше.
- Двигаемся дальше, пока не дойдем до конца обоих массивов.

**Вопрос 59.**

**Бинарный поиск**

Количествошагов поиска определится как $\log_2 n\uparrow$, где n-количествоэлементов, ↑ – округление в большую сторону до ближайшего целогочисла.

На каждом шагеосуществляется поиск середины отрезка по формуле *mid=(left+right)/2*

Если искомыйэлемент равен элементу с индексом *mid*,поиск

завершается.
В случае еслиискомый элемент меньше элемента с индексом *mid*, на место *mid*−1 перемещаетсяправая граница рассматриваемого отрезка, в противном случае – на место *mid*+1 перемещается левая граница.

## **ВОПРОС 60** ***Программирование на Python. Функции ввода-вывода.***

Для ввода нужной информации используется функция input(), которая по умолчанию возвращает в программу введённую пользователем *строку*.
a,b,c-имена переменных
= - оператор присваивания
a = input() (5)
b = input() (6)
c = a + b
print(c)  (56)
Чтобы преобразовать *строку из цифр* в целое число, воспользуемся функцией int()
a = int (input()) (5)
b = int (input()) (6)
d=2
c = (a + b ) *d  (11*2)
print(c)  (22)
Для ввода вещественного числа используем функцию float(input())
a = float (input()) (1.5)
b = float (input()) (3.05)
c = a + b  (4.55)
print(c)  (4.55)

Завывод данных в Python отвечают встроенная функция print(). Спомощью функции вывода print() можно вывести на экран любую информацию, например:

```
print('y=',y)
```

Информация в кавычках y= выводится на экран в виде текста, y показывает, что будет выведено значение переменной y(информация любого вида в зависимости от типа переменной)

```
print(5 + 10)
```

На экран будет выведена сумма чисел 5 и 10 (15)

```
print(37 // 3)
```

На экран будет выведена частное от деления нацело 37 на 3 (12)

**ВОПРОС 61** ***Программирование на Python. Переменные, имена переменных***

Переменная – это именованная область памяти, в которой во время выполнения программы хранятся данные определенного типа

Названия не должны начинаться с цифры, но могут заканчиваться цифрой. Например, назвать переменную 7up – неправильно, а так – seven11 – можно.

Названия могут состоять из комбинации строчных, заглавных букв, цифр и символов подчеркивания:lower_case, mixedCase, CapitalizedCase, UPPER_CASE, lower123.

Не следует давать переменным названия, совпадающие со служебными словами, названиями встроенных функций и методов, к примеру – print, list, dict, set, pass, break, raise.

Следует избегать использования отдельных букв, которые могут быть ошибочно приняты друг за друга– l (L в нижнем регистре), I (i в верхнем регистре) или зануль – O.

В названиях не должно быть пробелов, дефисов и специальных символов, например, ‘ или $.

**ВОПРОС 62** ***Программирование на Python. Типы данных.***

***Преобразование типов.***

**Типы данных:**

- Числовые – целые,вещественные, комплексные числа.

Примечание: для максимально точных расчетов с десятичными числами в Python используют модуль decimal (типданных Decimal), а для операций с рациональными числами (дробями) – модуль fractions (типданных Fraction).

- Булевы – логическиезначения True (истина)и False (ложь).
- Строковые – последовательностисимволов в кодировке Unicode.
- NoneType – нейтральноепустое значение, аналогичное null вдругих языках программирования.
- Последовательности – списки,кортежи, диапазоны.
- Словари – структура данныхтипа «ключ: значение».
- Множества – контейнеры,содержащие уникальные значения.

Подразделяются на изменяемые set инеизменяемые frozenset множества.

- Байтовые типы – bytes (байты), bytearray(изменяемаябайтовая строка), memoryview(предоставление доступа к внутренним даннымобъекта).

**Преобразование типов:**

Округление вещественного числа:

```
a = float(input())   (5.123)
print(int(a))        (5)
```

Преобразование целого числа в вещественное:

```
a = 5                (5)
print(float(a))      (5.0)
```

Преобразование строки в число и вывод числа безведущих нулей:

```
a = '00032567'
print(int(a))        (32567)
```

Сложение строки и числа:

```
a = 'Apollo'
b = 13
print(a + str(b))      (Apollo 13)
```

**ВОПРОС 63** Программирование на Python.Операторы языка (арифметические, логические, операции сравнения, операции состроками) Операции состроками?????

**Арифметическиеоператоры**

```
>>> 1 + 2    (сумма)
3
>>> 1 - 2     (разность)
-1
>>> 1 * 2      (умножение)
2
>>> 1 / 2       (деление)
0.5
>>> 10 // 3    (деление нацело)
3
>>> 10 % 3    (остаток от деления)
1
>>> 3 ** 2     (возведение в степень)
9
```

**Логическиеоператоры**

or – логическое "ИЛИ";

and – логическое "И";

not – логическое отрицание.

Пример:

```
>>> (1 + 1== 2) or (2 * 2 == 5)
True
>>> (1 + 1 == 2) and (2 * 2 == 5)
False
>>> (1 + 1 == 2) and not (2 * 2 == 5)
True
```

**Операторысравнения**

**==** –равно;

**!=** – не равно;

**>** – больше;

**<** – меньше;

**>=** –больше или равно;

**<=** –меньше или равно.

>>> 1 + 2 == 3

True

**ВОПРОС 64** Программирование на Python. Операторы присваивания.

**=** *значение правого операнда присвоится левому операнду;*

**+=** *сумма левого и правого операнда присвоится левомуоперанду;*

**-=** *разность левого и правого операнда присвоится левомуоперанду;*

**=* *произведение левого иправого операнда присвоится левому операнду;*

*/=* *разделит левый операнд направый и результат присвоится левому операнду;*

*//=* *результат целочисленногоделения левого операнда на правый операнд присвоится левому операнду*;

*%= разделитлевый операнд на правый по модулю и результат присвоится левому операнду;*

***=* *возведетлевый операнд в степень правого и результат присвоится левому операнду.*

Конкретные примеры работыоператоров (они скорее всего не нужны) в презентации 15 слайды 8-10

**ВОПРОС 65** Программирование на Python. Оператор выбора (if, else)

Синтаксически конструкция выглядит следующим образом:

- сначала записывается часть **if** сусловным выражением, которое возвращает истину или ложь;
- затем может следовать одна или несколько *необязательных*частей **elif** ;

- завершается запись этого составногооператора также *необязательной* частью **else**.

**ВОПРОС 66** Программирование на Python. Цикл while.

Цикл `while` ("пока") позволяет выполнить одну и ту же последовательностьдействий, пока проверяемое условие истинно. Условие записывается до тела циклаи проверяется до выполнения тела цикла. Как правило, цикл `while` используется, когда невозможно определить точное значениеколичества проходов исполнения цикла.

```
i=1
while i<10:
    print(i**2)
    i+=1
```

**ВОПРОС 67** Программирование на Python. Модуль math. Функцииэтого модуля.

Встроенныймодуль **math** в Python предоставляет наборфункций для выполнения математических, тригонометрических и логарифмическихопераций.

Для использования этихфункций в начале программы необходимо подключить модуль, что делаетсякомандой import: import math

Некоторые из основных функций модуля:

| | |
|---|---|
| **sqrt(X)** | **Квадратный корень из X** |
| **exp(X)** | Экспонента числа X |
| **log(X), log2(X), log10(X)** | Натуральный, двоичный и десятичный логарифм |
| **log(X, n)** | Логарифм X по основанию n |
| **sin(X), cos(X), tan(X)** | Синус, косинус и тангенс X, X указывается в радианах |
| **asin(X), acos(X), atan(X)** | Арксинус, арккосинус и арктангенс X |
| **atan2(X, Y)** | арктангенс отношения X/y с учётом квадранта |
| **sinh(X), cosh(X), tanh(X)** | Гиперболические синус, косинус и тангенс X |
| **asinh(X), acosh(X), atanh(X)** | Обратный гиперболический синус, косинус и тангенс X |

| | |
|---|---|
| **pi** | Выдаётся число π |
| **e** | Выдаётся число e |

**ВОПРОС 68** ***Программирование на Python. Применение цикла for когда вкачестве множества значений используется список.***

Цикл for позволяет проводить итерации –реализовывать набор инструкций нужное количество раз.

Его используют, когда количество итераций(повторов)известно заранее.

В цикле for указывается переменная цикла имножество значений, по которому будет пробегать переменная. Множество значенийможет быть задано *списком*, *кортежем*, *строкой* или *диапазоном*.

Список (list) – это упорядоченный наборэлементов, каждый из которых имеет свой номер, или индекс, позволяющий быстро получитьк нему доступ. Нумерация элементов в списке начинается с 0.В одном списке одновременно могут лежать данные разных типов –например, и строки, и числа. В один список можно положить другойсписок.

Списки называют динамическими структурами данных,потому что их можно менять на ходу: удалить один или несколькоэлементов, заменить или добавить новые.

Списки записываются с использованием квадратных скобок[].

**ВОПРОС 69** ***Программирование на Python. Применение цикла for когда вкачестве множества значений используется кортеж***

Кортежи (тип tuple) – это неизменяемый тип данныхв Python, который используется для хранения упорядоченной последовательностиэлементов.

У этих коллекций есть три свойства:

*Неизменяемость*.После того как кортеж создан, в него нельзя добавлять элементы, а такжеизменять их или удалять.

*Упорядоченность*. Элементыкортежа располагаются в определённом порядке, который тоже неизменяем. К

любомуэлементу можно обратиться по его индексу (порядковому номеру).

*Элементами кортежа могут быть объектыразных типов данных*: числа, строки, списки, другие кортежи идругие. Элементы-коллекции могут иметь неограниченную глубину вложенности.Например, кортеж может включать в себя список, который будет содержать другойсписок, который вновь будет содержать список и так далее.

Неизменяемость кортежей не абсолютна. У неё естьисключение – если внутри кортежа находятся изменяемые элементы, напримерсписки, словари или множества, то их значения можно изменить.

Кортежи записываются с использованием круглыхскобок ()

Кортежи занимают в памяти меньше места, чем списки

**ВОПРОС 70** ***Программирование на Python. Применение цикла for когда вкачестве множества значений используется диапазон.***

Диапазон (тип range) – это специальный тип данных вPython, который позволяет создавать последовательности чисел, не сохраняя их впамяти. Он часто используется в циклах for для выполнения повторяющихсяопераций.

У диапазона есть три ключевых свойства:

1. Ленивое вычисление. Диапазон не создаёт все числасразу, а генерирует их по мере необходимости, что делает его эффективным сточки зрения памяти.

2. Упорядоченность. Числа в диапазоне всегда идут взаданном порядке.

3. Гибкость. Диапазон можно настроить с учётомначального значения, конечного значения (не включается в последовательность) ишага.

Диапазон создаётся с использованием функции range() иможет принимать от одного до трёх аргументов:

- range(stop) – генерирует числа от 0 до stop-1.
- range(start, stop) – генерирует числа от start достop-1.
- range(start, stop, step) – генерирует числа от startдо stop-1 с указанным шагом step.

Пример применения цикла for с диапазоном:

1. Простая итерация по числам:

```
for i in range(5): # Диапазон от 0 до 4
    print(i)
# Вывод: 0, 1, 2, 3, 4
```

2. Указание начального и конечного значения:

```
for i in range(2,7):  # Диапазон от 2 до 6
    print(i)
# Вывод: 2, 3, 4, 5, 6
```

3. Использование шага:

```
for i in range(1, 10, 2):  # Диапазон от 1 до 9 с шагом 2
    print(i)
# Вывод: 1, 3, 5, 7, 9
```

4. Обратный порядок:

```
for i in range(10, 0, -2):  # Диапазон от 10 до 2 с шагом -2
    print(i)
# Вывод: 10, 8, 6, 4, 2
```

Преимущества использования диапазона в цикле for:

- Экономия памяти за счёт ленивого вычислениязначений.
- Удобство задания параметров последовательности(начало, конец, шаг).
- Возможность работы с большими последовательностямичисел.

Диапазоны применяются во множестве задач, например,для выполнения повторяющихся действий, обработки индексов в списках, построенияпоследовательностей чисел или организации вложенных циклов.

## ***ВОПРОС 71 Программированиена Python. Функция range. Ее аргументы. Возможныеоперации с функцией.***

Дляполучения последовательность чисел можно использовать функцию range,которая генерирует последовательность чисел.

```
# итерация по числам с нуля до 10 не включительно
for i in range(0, 10):
    print(i)
```

```
0
1
2
```

3
4
5
6
7
8
9
i – переменная цикла
range(0, 10) –последовательность чисел от 0 до 9, невключая 10
Уфункции 3 параметра:

- start – начало последовательности [включительно](не обязательный параметр, по умолчанию равен 0).
- stop – задает точку остановки последовательности[значение не включено в последовательность] (обязательный параметр).
- step – шаг последовательности (не обязательныйпараметр, по умолчанию равен 1).

range(stop)  # с одним параметров
range(start,stop, step)  # с несколькими параметрами
Приработе с функцией range() важно помнить следующее:

- Значение stop не входит впоследовательность;
- Все аргументы функции должны быть целыми числами(положительными или отрицательными);
- При отрицательном шаге step нужно помнить,что значение start должно быть больше значения stop;
- Значение step не должно быть равно 0, иначеPython вызовет исключение "ValueError".

**ВОПРОС 72** ***Программирование на Python. Модуль random. Функции random(), randint()randrange().***

**Модульrandom управляет генерацией случайных чисел. Его основные функции:**

- random():генерирует случайное число от 0.0 до 1.0
- randint():возвращает случайное число из определенного

диапазона

- randrange():возвращает случайное число из определенного набора чисел

**Функция randint(min, max)** возвращаетслучайное целое число в промежутке между двумя значениями min и max.

**Функция randrange()** возвращаетслучайное целое число из определенного набора чисел. Она имеет три формы:

- randrange(stop): в качестве набора чисел,из которых происходит извлечение случайного значения, будет использоватьсядиапазон от 0 до числа stop
- randrange(start, stop): набор чиселпредставляет диапазон от числа start до числа stop
- randrange(start, stop, step): набор чиселпредставляет диапазон от числа start до числа stop, при этом каждое число вдиапазоне отличается от предыдущего на шаг step