Федеральное агентство по рыболовству
БГАРФ ФГБОУ ВО «КГТУ»
Калининградский морской рыбопромышленный колледж

Утверждаю
Врио заместителя начальника колледжа
по учебно-методической работе
М.Ю. Никишин

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00861698655D171818D836B833A86412BD
Владелец: НИКИШИН МИХАИЛ ЮРЬЕВИЧ
Действителен: с 13.02.2026 до 09.05.2027

Учебно-методическое пособие к практическим занятиям по дисциплине

ООД.08 ИНФОРМАТИКА

(1 ЧАСТЬ)

15.02.06 Монтаж, техническая эксплуатация и ремонт холодильно-компрессорных и теплонасосных машин и установок (по отраслям)

МО–15 02 06-ООД.08.ПЗ

РАЗРАБОТЧИКИ	Сукорская А.О. Иванова Т.Ю.
ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ	Никишин М.Ю.
ГОД РАЗРАБОТКИ	2026

Содержание

Урок №1 Техника безопасности при работе в компьютерном классе. Безопасность, гигиена, эргономика, ресурсосбережение.....4

Практическое занятие №1 (Входной контроль). Информация, информационные процессы, информационные системы. Информационная деятельность человека. Образовательные информационные ресурсы КМРК. Электронная библиотека 11

Практическое занятие №2 Подходы к измерению информации. Единицы измерения информации. Определение объемов информации.....21

Урок №2 Компьютер и цифровое представление информации. Устройство компьютера.....27

Практическое занятие №3 Логические основы компьютера.....43

Практическое занятие №4 Кодирование данных различного вида. Их представление в компьютере.....57

Практическое занятие №5 Системы счисления. Переводы чисел из одной системы счисления в другую. Арифметические действия в различных системах счисления. .67

Практическое занятие №6 Элементы комбинаторики, теории множеств и математической логики. 95

Практическое занятие №7 Структура информации. Графы. Введение и понятия. Способы задания графов. Алгоритм построения дерева решений. 105

Практическое занятие № 8 Решение логических задач с помощью графов. Анализ алгоритмов в профессиональной области.....117

Практическое занятие № 9 Алгоритмы и способы их описания. Линейные и условные алгоритмы. (составление трассировочных таблиц) Описание алгоритмов с помощью блок-схем 123

Практическое занятие № 10 Циклические алгоритмы (составление трассировочных таблиц). Описание алгоритма с помощью блок-схем 132

Практическое занятие № 11 Введение в язык программирования Python. Ввод и вывод данных. Типы данных.....139

Практическое занятие № 12 Оператор присваивания. Математические операции с целыми и вещественными Стандартные функции. Математический модуль math. Составление программ профессиональной направленности. 151

Практическое занятие № 13 Проверка условий в Python. Синтаксис If,If-else,if-elif-else. Составление программ с проверкой условий из профессиональной деятельности.....167

Практическое занятие № 14 Реализация циклических алгоритмов в Python. Синтаксис цикла с предусловием и постусловием. Составление программ профессиональной направленности. 176

Практическое занятие № 15 Функция range. Синтаксис цикла с параметром. Составление программ профессиональной направленности. 183

Практическое занятие № 16 Понятие списка в Python. Создание и считывание списков. Функции и методы списков. Работа со строками. Понятие картежа и словаря. Создание словарей и кортежей. Методы словарей.....188

Практическое занятие № 17 Система координат в графике Python, управление цветом. Примитивы в Python Использование процедур в графике.....213

Практическое занятие № 18 Использование цикла в графике и закрашивание областей 222

Практическое занятие № 19. Построение графиков математических функций в Python.....226

Практическое занятие № 20 Анимация в Python 229

Урок №1 Техника безопасности при работе в компьютерном классе. Безопасность, гигиена, эргономика, ресурсосбережение

Цель занятия: Ознакомить с техникой безопасности при работе в кабинете информатики

Познакомить с понятиями гигиена, эргономика и ресурсосбережение

Исходные данные: раздаточный материал, журнал инструктажа по технике безопасности

Содержание и порядок выполнения задания:

1. ПРАВИЛА ПОВЕДЕНИЯ УЧАЩИХСЯ В УЧЕБНОМ КАБИНЕТЕ ИНФОРМАТИКИ

1.1. Находясь в кабинете информатики, учащиеся обязаны:

- соблюдать дисциплину и порядок, правила техники безопасности и чистоту;
- занимать рабочие места согласно указаниям преподавателя и не менять их самовольно;
- заниматься только тем видом деятельности, которую определил преподаватель;
- немедленно сообщать преподавателю о любых замеченных неисправностях оборудования или неверной работе программного обеспечения;
- немедленно сообщать преподавателю о любом случае травматизма в кабинете, особенно от электрического тока.

1.2. Находясь в кабинете информатики, учащийся имеет право:

- на помощь и консультацию преподавателя;
- самостоятельно экстренно отключить электрооборудование, если от этого зависит безопасность его или окружающих.

2 ПРАВИЛА ТЕХНИКИ БЕЗОПАСНОСТИ В КАБИНЕТЕ ИНФОРМАТИКИ

1. ИСТОЧНИКИ ОПАСНОСТИ:

- электроприборы с напряжением питания 220 В, мониторы и телевизоры, которые могут явиться источником электротравматизма;
- наличие электроприборов увеличивает опасность возгорания;

- мониторы компьютеров, телевизоры являются слабыми источниками ионизирующего излучения электромагнитных, электрических и магнитных статических полей.

3. ПРАВИЛА БЕЗОПАСНОСТИ

ЗАПРЕЩАЕТСЯ:

- работать с электроприборами, имеющими повреждения корпуса или изоляции соединительных проводов;
- производить самовольное переключение разъёмов оборудования;
- приносить и самовольно подключать какое-либо оборудование;
- вставлять в отверстие приборов посторонние предметы;
- выключать или включать приборы без разрешения преподавателя.

Если производится выключение/включение, то интервал времени между включением/и выключением/включением должен быть не менее 15 секунд.

В СЛУЧАЕ ПОРАЖЕНИЯ ЭЛЕКТРИЧЕСКИМ ТОКОМ, НЕОБХОДИМО:

- прекратить действие тока (лучше всего экстренным выключением приборов, т.к. попытка оттащить пострадавшего может привести к поражению током спасающего);
- немедленно сообщить о происшедшем преподавателю (даже если на первый взгляд всё обошлось лёгким испугом);
- оказать первую помощь, если необходима.

4. ПРАВИЛА ПОЖАРНОЙ БЕЗОПАСНОСТИ

ЗАПРЕЩАЕТСЯ:

- использовать источники открытого огня (спички, зажигалки, петарды и др.);
- приносить на уроки легковоспламеняющиеся вещества (лаки, краски, порох и т.п.);
- пользоваться неисправными электроприборами (в случае появления специфического запаха горящей изоляции, соответствующий прибор необходимо немедленно отключить и сообщить учителю);
- загромождать или закрывать проходы к путям эвакуации и доступ к средствам первичного пожаротушения;
- производить тушение возгорания не отключенных электроприборов водой или обычными огнетушителями;
- привлекать учащихся к тушению пожара.

В СЛУЧАЕ УГРОЗЫ ПОЖАРА (возгорания, задымленность) НЕОБХОДИМО:

- немедленно отключить все электроприборы, определить источники возгорания (задымленности) и ликвидировать его средствами первичного пожаротушения;
- если первичные действия по ликвидации возгорания в течение первых же минут не дали результата, учащиеся эвакуируются согласно плану эвакуации, по школе объявляется тревога, сообщается о пожар.

САНИТАРНО-ГИГИЕНИЧЕСКИЕ НОРМЫ ПРИ РАБОТЕ С КОМПЬЮТЕРОМ

Требования к организации занятий в кабинете информатики

- расстояние от центра экрана до глаз учащихся должно быть не менее 60 см;
- время интенсивной непрерывной работы на компьютере не должно превышать 25 минут, после чего обязателен перерыв с разминкой;
- в кабинете должна быть обеспечена вентиляция и проветривание между уроками.

Теоретический материал

Эргономика – наука о том, как люди с их различными физическими данными и особенностями жизнедеятельности взаимодействуют с оборудованием и машинами, которыми они пользуются.

Цель эргономики состоит в том, чтобы обеспечить комфорт, эффективность и безопасность при пользовании компьютерами уже на этапе разработки клавиатур, компьютерных плат, рабочей мебели и др. для устранения физического дискомфорта и проблем со здоровьем на рабочем месте.

В связи с тем, что всё больше людей проводят много времени перед компьютерными мониторами, ученые многих областей, включая анатомию, психологию и охрану окружающей среды, вовлекаются в изучение правильных, с точки зрения эргономики, условий работы.

Так называемые *эргономические заболевания* – быстрорастущий вид профессиональных болезней.

Если в организации рабочего места оператора ПК допускается несоответствие параметров мебели антропометрическим характеристикам человека, то это вызывает необходимость поддержания вынужденной рабочей позы и может

привести к нарушениям в костно-мышечной и периферической нервной системе. Длительный дискомфорт в условиях недостаточной физической активности может вызывать развитие общего утомления, снижения работоспособности, боли в области шеи, спины, поясницы. У операторов часто диагностируются заболевания опорно-двигательного аппарата и периферической нервной системы: невриты, радикулиты, остеохондроз и др.

Главной частью профилактических мероприятий в эргономике является правильная посадка.

Негативные последствия работы за монитором возникают из-за того, что:

а) наш глаз предназначен для восприятия отражённого света, а не излучаемого, как в случае с монитором (телевизором),

б) пользователю приходится вглядываться в линии и буквы на экране, что приводит к повышенному напряжению глазных мышц.

Для нормальной работы нужно поместить монитор так, чтобы глаза пользователя располагались на расстоянии, равном полутора диагоналям видимой части монитора:

- не менее 50-60 см для 15" монитора;
- не менее 60-70 см для 17" монитора;
- не менее 70-80 см для 19" монитора;
- не менее 80-100 см для 21" монитора.

Если зрение не позволяет выдерживать это расстояние, тогда уменьшите разрешение изображения и увеличьте шрифты.

Оптимальная диагональ экрана для работ с текстовыми документами - 15"-17" с разрешением 1024x768. Для графических работ необходим монитор 19"-21" при разрешении 1280x1024 и выше. Для игр рекомендуется 17"-19". Мониторы больших диагоналей приобретать не рекомендуется, т.к. от работы за слишком крупными мониторами, по словам пользователей, "глаза становятся квадратными".

От большого монитора необходимо сидеть дальше, чем от маленького. И в итоге угловая площадь монитора остается такой же. Но сфокусировать глаз на мелком изображении, находящемся в 1-1.5 метрах от глаза становится труднее, что ведет к перенапряжению зрительного аппарата. Чем крупнее объект на экране монитора, тем меньше утомляемость. Поэтому компьютерные игры с их рисованными фигурами утомляют меньше, чем цифры и буквы.

Экран монитора должен быть абсолютно чистым. Периодически и при необходимости протирайте его специальными салфетками.

Усталость от работы с монитором тем меньше, чем ниже яркость экрана и чем крупнее объекты на экране. Установите минимальную яркость, при которой можно без напряжения различать символы на экране. Учтите, что лучше увеличить шрифт или изображение, чем пододвинуться поближе к экрану или увеличить яркость. Современные операционные системы имеют для этого специальные средства. Шрифты на экране можно масштабировать, задавать минимальные размеры элементов рисунков и прочее.

Система гигиенических требований

Длительная работа с компьютером может приводить к расстройствам состояния здоровья. Кратковременная работа с компьютером, установленным с грубыми нарушениям гигиенических норм и правил, приводит к повышенному утомлению. Вредное воздействие компьютерной системы на организм человека является комплексным. Параметры монитора оказывают влияние на органы зрения. Оборудование рабочего места влияет на органы опорно-двигательной системы. Характер расположения оборудования в компьютерном классе и режим его использования влияет как на общее психофизиологическое состояние организма, так и на органы зрения.

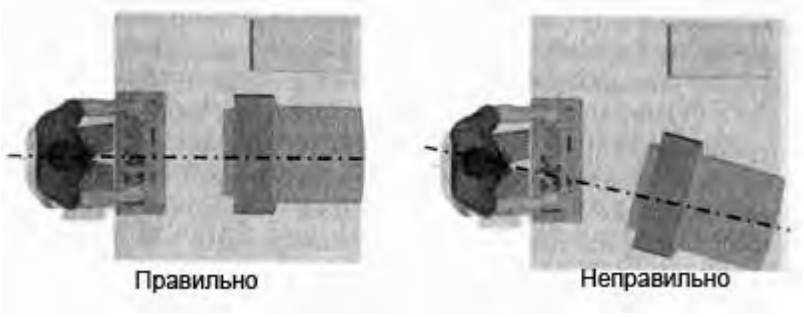
Требования к видеосистеме

В прошлом монитор рассматривали в основном как источник вредных излучений, воздействующих прежде всего на глаза. Сегодня такой подход считается недостаточным. Кроме вредных электромагнитных излучений (которые на современных мониторах понижены до сравнительно безопасного уровня) должны учитываться параметры качества изображения, а они определяются не только монитором, но и видеоадаптером, то есть всей видеосистемы в целом.

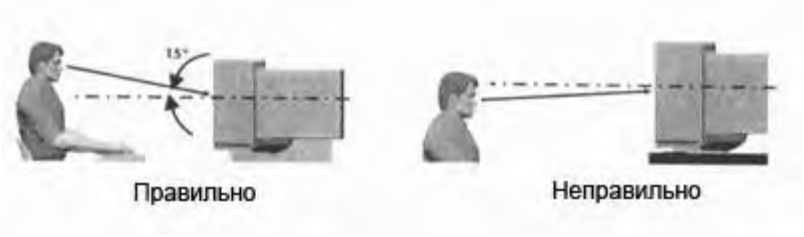
Требования к рабочему месту

В требования к рабочему месту входят требования к рабочему столу, посадочному месту (стулу, креслу), Подставкам для рук и ног. Несмотря на кажущуюся простоту, обеспечить правильное размещение элементов компьютерной системы и правильную посадку пользователя чрезвычайно трудно. Полное решение проблемы требует дополнительных затрат, сопоставимых по величине со стоимостью отдельных узлов компьютерной системы, поэтому и в быту и на производстве этими требованиями часто пренебрегают.

Монитор должен быть установлен прямо перед пользователем и не требовать поворота головы или корпуса тела.

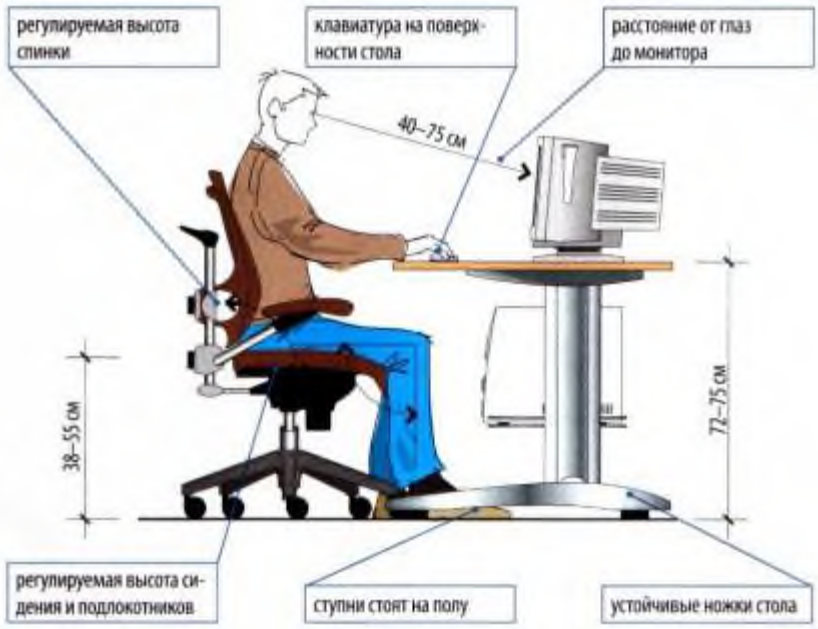


Рабочий стол и посадочное место должны иметь такую высоту, чтобы уровень глаз пользователя находился чуть выше центра монитора. На экран монитора следует смотреть сверху вниз, а не наоборот. Даже кратковременная работа с монитором, установленным слишком высоко, приводит к утомлению шейных отделов позвоночника.



Если при правильной установке монитора относительно уровня глаз выясняется, что ноги пользователя не могут свободно покоиться на полу, следует установить подставку для ног, желательно наклонную. Если ноги не имеют надежной опоры, это непременно ведет к нарушению осанки и утомлению позвоночника. Удобно, когда компьютерная мебель (стол и рабочее кресло) имеют средства для регулировки по высоте. В этом случае проще добиться оптимального положения.

Клавиатура должна быть расположена на такой высоте, чтобы пальцы рук располагались на ней свободно, без напряжения. Для работы рекомендуется использовать специальные компьютерные столы, имеющие выдвижные полочки для клавиатуры.



При длительной работе с клавиатурой возможно утомление сухожилий кистевого сустава. Известно тяжелое профессиональное заболевание — кистевой туннельный синдром, связанное с неправильным положением рук на клавиатуре.

При работе с мышью рука не должна находиться на весу. Локоть руки или хотя бы запястье должны иметь твердую опору. Если предусмотреть необходимое расположение рабочего стола и кресла затруднительно, рекомендуется применить коврик для мыши, имеющий специальный опорный валик. Нередки случаи, когда в поисках опоры для руки (обычно правой) располагают монитор сбоку от пользователя (соответственно, слева), чтобы он работал вполоборота, опирая локоть или запястье правой руки о стол. Этот прием недопустим. Монитор должен обязательно находиться прямо перед пользователем.

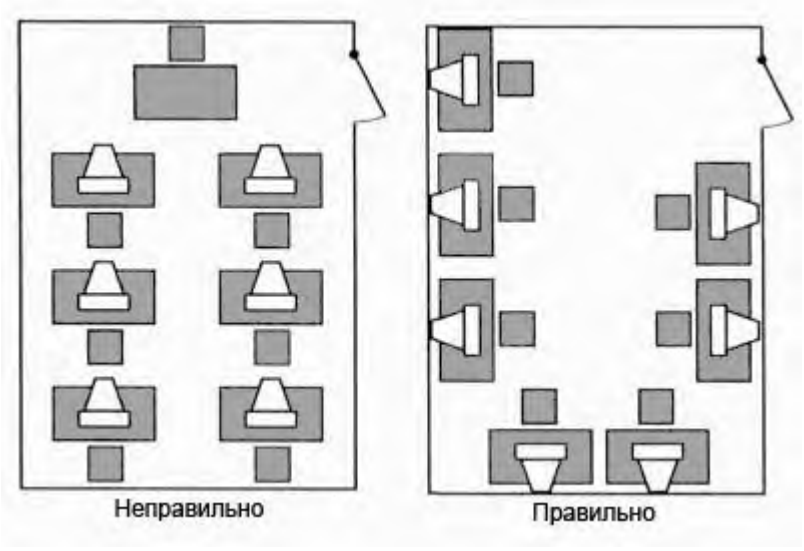
Требования к организации занятий.

Экран монитора — не единственный источник вредных электромагнитных излучений.

Монитор компьютера следует располагать так, чтобы задней стенкой он был обращен не к людям, а к стене помещения. В компьютерных классах, имеющих несколько компьютеров, рабочие места должны располагаться по периметру помещения, оставляя свободным центр. При этом дополнительно необходимо проверить каждое из рабочих мест на отсутствие прямого отражения внешних источников освещения. Как правило, добиться этого для всех рабочих мест

одновременно достаточно трудно. Возможное решение состоит в использовании штор на окнах и продуманном размещении искусственных источников общего и местного освещения.

Сильными источниками электромагнитных излучений являются устройства бесперебойного питания. Располагать их следует как можно дальше от посадочных мест пользователей.



В организации занятий важную роль играет их продолжительность, от которой зависят психофизиологические нагрузки.

Выполнение практической работы не предусматривает выставление оценок

Вопросы для самопроверки

1. Перечислите основные негативные последствия работы за монитором
2. Объясните цель эргономики
3. Что является сильными источниками электромагнитных излучений?

Практическое занятие №1 (Входной контроль). Информация, информационные процессы, информационные системы. Информационная деятельность человека. Образовательные информационные ресурсы КМРК. Электронная библиотека

Цель занятия:

1. Проверить остаточные знания по предмету Информатика.
2. Ознакомиться с основными этапами развития информационного общества, этапами развития технических средств и информационных ресурсов.

3. Научиться работать с документами в локальной сети.
4. Научиться выполнять поиск информации по теме.
5. Научиться пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью.

Исходные данные:

Папка на РС «Практическое занятие № 1»

Содержание и порядок выполнения задания:

Выполните задания входного тестирования.

Изучить теоретическую часть практического занятия и выполнить задания.

Пример входного тестирования

1. **Под информацией понимают:**

Сигналы от органов чувств человека
Характеристику объекта, выраженную в числовых величинах
Разнообразие окружающей действительности

2. **В позиционной системе счисления значение каждой цифры зависит:**

От значения числа
От значений соседних знаков
От позиции, которую занимает знак в записи числа

3. **За единицу измерения информации в теории кодирования принят:**

1 байт
1 бод
1 бит

4. Электронная таблица предназначена для:

Обработки преимущественно числовых данных, структурированных с помощью таблиц,

Визуализации структурных связей между данными, представленными в таблицах,

Хранения и редактирования больших объемов текстовой информации,

5. Производительность работы компьютера (быстрота выполнения операций) зависит от...

размера экрана дисплея
частоты процессора
быстроты, нажатия на клавиши

6. Драйвер – это:

1) устройство компьютера;
2) программа для работы с устройствами компьютера,
3) язык программирования.

7. В целях сохранения информации CD-ROM необходимо оберегать от...
1) холода
2) магнитных полей
3) загрязнения

8. Укажите устройство, не являющееся устройством вывода информации:
1) монитор
2) принтер
3) клавиатура

Часть В.

9. Ниже в табличной форме представлен фрагмент базы данных о результатах тестирования учащихся (используется сто балльная шкала).

Фамилия	Пол	математика	химия	информатика	Биология
Аганян	Ж	52	43	82	74
Воронин	М	92	75	93	55
Григорчук	М	66	69	51	68
Роднина	Ж	73	51	40	92
Сергеевко	Ж	81	83	83	41
Черепанова	ж	94	64	71	20

Сколько записей в данном фрагменте удовлетворяет условию
«Математика > 60 И Информатика > 55»?
В ответе укажите одно число — искомое количество записей.
Ответ: _____

10. Пользователь работал с каталогом С:\Архив\Рисунки\Натюрморты.
Сначала он поднялся на один уровень вверх, затем еще раз поднялся на один уровень вверх и после этого спустился в каталог Фотографии. Запишите полный путь каталога, в котором оказался пользователь.
1) С:\Архив\Рисунки\Фотографии
2) С:\Архив\Фотографии
3) С:\Фотографии\Архив
4) С:\Фотографии

Часть С.

11. Сколько Кбайт информации содержит сообщение объемом 2^{16}бит? В ответе укажите одно число.
Ответ: _____

Теоретическая часть

Этапы развития информационного общества.

В развитии человечества существуют четыре этапа, названные информационными революциями, которые внесли изменения в его развитие. В настоящее время можно выделить **шесть информационных революций.**

1. **Второй этап – связан с изобретением письменности.** Это обусловило качественный гигантский и количественный скачек в развитии общества (5 с половиной тысяч лет назад). Знания стало возможно накапливать и передавать последующим поколениям, т.е. появились средства и методы накопления информации. В некоторых источниках считается, что содержание первой информационной революции составляет распространение и внедрение в деятельность и сознание человека языка.

2. **Третий этап – изобретение книгопечатания.** Это дало в руки человечеству новый способ хранения информации, а так же сделало более доступным культурные ценности.(1445-1500г)

3. **Четвертый этап – изобретение электричества.** Появились телеграф, телефон и радио, позволяющие быстро передавать и накапливать информацию в любом объеме. Появились средства информационных коммуникаций(1600-1700).

4. **Пятый этап – изобретение микропроцессорной технологии и персональных компьютеров.** Толчком к этой революции послужило создание в середине 40-х годов ЭВМ (1960 г.) Эта последняя революция дала толчок человеческой цивилизации для перехода от индустриального к информационному обществу- обществу, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей ее формой – знанием. Началом этого послужило внедрение в различные сферы деятельности человека современных средств обработки и передачи информации – этот процесс называется информатизацией.

Первую информационную революцию можно связать с появлением языка и членораздельной человеческой речи. Развитие языка оказало колоссальное влияние на развитие сознания людей, а его использование в их практической деятельности и стало информационной основой создания первых технологий. Язык сделал возможным развитие процессов абстрактного мышления, зарождения интеллектуальной деятельности людей, а также новой технологии накопления и передачи информации от поколения к поколению в виде легенд, мифов и сказаний. Это были «живые знания», носителями которых были люди (старейшины, жрицы, шаманы). Такое сохранение информации был недостаточно надежным. Со смертью их носителей многие знания утрачивались и должны были формироваться заново. Именно поэтому **вторую информационную революцию связывают с изобретением письменности**.

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

Шестую информационную революцию связывают с формированием на нашей планете новой цивилизации- **информационного общества**, в котором главную роль будет играть интеллект и знания. (1990г- по сегодняшние дни)

Стадии развития ИТ

Для того, чтобы оставаться конкурентоспособными, большинство отраслей с течением времени используют всё новые решения в области информационных технологий.

Если попытаться оценить различные этапы развития ИТ, то они могут быть классифицированы следующим образом:

1. **Создание компьютеров.** Некоторые образцы компьютеров в 1940-1950гг., аналоговые вычислительные машины, разработка радарных систем.

2. **Расширение области применения компьютеров, появление первых вычислительных центров.** Характерно для 1960-1975гг., наряду с научно-исследовательскими институтами, было начато использование компьютеров в сфере производства, всеобщая эйфория, ожидания в области использования информационных технологий были очень высоки.

3. **Локальное использование компьютеров.** Характерно для 1975-1980гг., когда при предприятиях стали появляться вычислительные центры, при автоматизации производства стали использоваться компьютеры для конкретного направления и оптимизации процессов (энергетические системы, транспортные задачи), или для управления станками, конвейерами и многим другим.

4. **"Автоматизированные" рабочие места.** В 1980-х годах начали появляться персональные компьютеры. Правда, вычислительные центры и ИТ-отделы продолжали свою деятельность и крупно объемные вычислительные работы проводились всё ещё там, возникли новые возможности для обработки текстов, а также в области дизайна и проектирования.

5. **Интеграция внутри предприятия.** В 1990-х стали широко использоваться персональные компьютеры и благодаря наличию операционных систем хорошего

уровня наряду с достаточно надежными каналами связи стало возможным использование многоуровневой модели компьютерной системы.

6**. Глобализация.** Быстрое развитие интернета и цифровых коммуникационных технологий были основой для быстрого появления новых логических сетей связи. Вычислительная техника превратилась из локального инструмента в незаменимый элемент при работе в глобальной системе. В связи с появившимися новыми возможностями можно говорить о так называемой «новой экономике», где прочно утвердились новые модели бизнеса (ИТ-компании, работа на дому, электронный бизнес, «облачные технологии» и др.).

Основные черты информационного общества.

Информационное общество — общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей её формы — знаний.

Некоторые характерные черты информационного общества:

1. Объёмы информации возрастут и человек будет привлекать для её обработки и хранения специальные технические средства.
2. Неизбежно использование компьютеров.
3. Движущей силой общества станет производство информационного продукта.
4. Увеличится доля умственного труда, т.к. продуктом производства в информационном обществе станут знания и интеллект.
5. Произойдёт переоценка ценностей, уклада жизни и изменится культурный досуг.
6. Развиваются компьютерная техника, компьютерные сети, информационные технологии.
7. У людей дома появляются всевозможные электронные приборы и компьютеризированные устройства.
8. Производством энергии и материальных продуктов будут заниматься машины, а человек главным образом обработкой информации.
9. В сфере образования буде создана система непрерывного образования.
10. Дети и взрослые смогут обучаться на дому с помощью компьютерных программ и телекоммуникаций.

11. Появляется и развивается рынок информационных услуг.

Задание 1

Заполните таблицу «Информационные революции»

Информационная революция	Период времени	Радикальные изменения в истории человечества	Основные изобретения (место, изобретатели)
Первая			
Вторая			
Третья			
Четвертая			
Пятая			
Шестая			

Задание 2

Заполните таблицу «Развитие информационных технологий»

Этап	Время	Информационная технология	Коммуникация, связь
1			
2			
3			
4			
5			

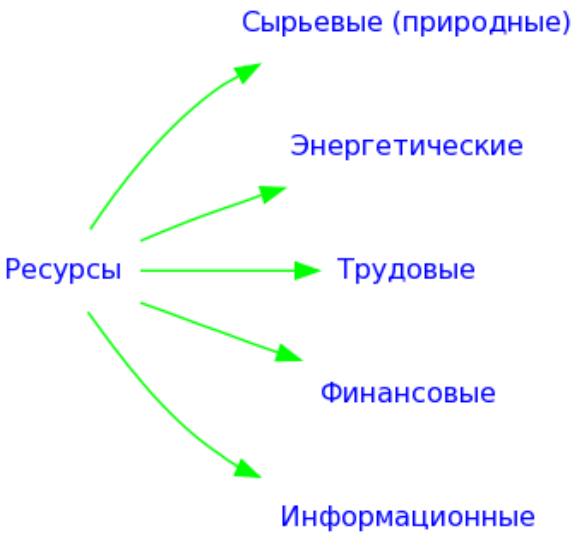
Задание 3

Изучите понятие информационных ресурсов, классификацию информационных ресурсов. И создайте из полученной информации ***«Облако тегов».***

Облако тегов или облако ключевых слов- это визуальное представление списка категорий. Ключевые слова чаще всего представляют собой отдельные слова и важность каждого ключевого слова обозначается размером шрифта или цветом.

Понятие информационных ресурсов. Классификация информационных ресурсов.

В информационном обществе особое внимание уделяется информационным ресурсам. Они существовали всегда, но в условиях индустриального общества никогда не рассматривались как отдельная категория.



Информационный ресурс может принадлежать одному человеку, организации, городу, региону. Например, если информация собирается со всего мира, тогда говорят и мировых информационных ресурсах. Понятие информационный ресурс опубликовано в Федеральном законе от 27 июля 2006 года № 149-ФЗ “Об информации, информационных технологиях и о защите информации”.

Информационные ресурсы – знания, подготовленные людьми для специального использования и зафиксированные на материальном носителе. Информационные ресурсы - отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах).

Информационная система – совокупность информационных потоков, а также методов, средств и технологий по обработке информации в соответствии с требованиями отдельных лиц.

Мировые информационные ресурсы – называют информационные ресурсы, рассматривающие совокупность информационных ресурсов различных государств.

Например: каждый раз, покупая газету или журнал, Вы приобретаете информационный ресурс (если в нем опубликована международная информация, то такой ресурс можно считать мировым).

Разделить мировые информационные ресурсы на определенные виды (категории) сложно, но можно выделить несколько критериев, по которым принято разбивать их на виды:

1 критерий – по форме представления – текстовый, графический и др. Самый обобщенный критерий, по нему можно легко производить разбиение, не требуется

тщательный анализ информации для определения принадлежности к тому или иному виду.

2 критерий – по ограниченности доступа – сведения, составляющие государственную тайну, персональные данные, сведения, составляющие коммерческую тайну. Чтобы отнести документ в какую – либо категорию нужно опираться на законодательство.

3 критерий – разбивка по тематике – законодательные ресурсы, статистические ресурсы, обучающие ресурсы. Самый большой и сложный критерий, так как одни и те же ресурсы можно отнести к нескольким категориям.

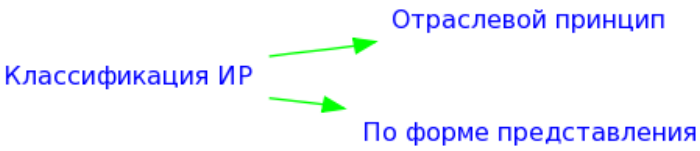
Это три основных способа разбивки мировых информационных ресурсов на виды, притом можно выделить много критериев: по форме предоставления, происхождению и т. д.)

Провести четкую грань между коммерческими или некоммерческими информационными ресурсами не просто. К полностью некоммерческим ресурсам можно отнести:

- центральное телевидение;
- радио;
- бесплатные рассылки и брошюры.

Наряду с центральным телевидением есть и коммерческие каналы, за подключение к которым нужно платить. В настоящее время очень развит Internet, где практически все организации и компании печатают информацию, которая распространяется бесплатно. Кроме этого существуют и коммерческие проекты, которые взимают плату за пользование их ресурсами.

Классификация информационных ресурсов



Классификация национальных информационных ресурсов.

Образовательными информационными ресурсами

Под образовательными информационными ресурсами понимают текстовую, графическую и мультимедийную информацию, а также исполняемые программы (дистрибутивы), то есть электронные ресурсы, созданные специально для использования в процессе обучения на определенной ступени образования и для определённой предметной области.

К образовательным электронным ресурсам можно отнести: – учебные материалы (электронные учебники, учебные пособия, рефераты, дипломы), – учебно-методические материалы (электронные методики, учебные программы), – научно-методические (диссертации, кандидатские работы), – дополнительные текстовые и иллюстративные материалы (лабораторные работы, лекции), – системы тестирования (тесты – электронная проверка знаний), – электронные полнотекстовые библиотеки; – электронные периодические издания сферы образования; – электронные оглавления и аннотации статей периодических изданий сферы образования, – электронные архивы выпусков.

При работе с образовательными ресурсами появляются такие понятия, как субъект и объект этих ресурсов. Субъекты информационной деятельности классифицируются следующим образом:

– субъект, создающий объекты (все пользователи образовательной системы - преподаватель, студент);

– субъект, использующий объекты (все пользователи образовательной системы);

– субъект, администрирующий объекты, то есть обеспечивающий среду работы с объектами других субъектов (администраторы сети);

– субъект, контролирующий использование объектов субъектами (инженеры).

Образовательные ресурсы КМРК – сайт do.kmrk.ru – сервер дистанционного обучения.

Вход на сервер дистанционного обучения КМРК

Введите адрес сервера ДО КМРК http://do.kmrk.ru/

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список используемых источников
5. Выводы и предложения

Вопросы для самопроверки:

1 Что такое информационные ресурсы?

2 Что такое образовательные информационные ресурсы?

3 Что относится к образовательным информационным ресурсам?

4 Каковы субъекты и объекты образовательных информационных ресурсов?

Практическое занятие №2 Подходы к измерению информации. Единицы измерения информации. Определение объемов информации.

Цель занятия:

1. Изучить понятие количества информации,
2. Познакомиться с содержательным и алфавитным подходом при определении количества информации,
3. Познакомиться с единицами измерения информации, получить практические навыки по определению количества информации.

Исходные данные:

презентация

Содержание и порядок выполнения задания:

Изучите теоретическую часть.

Выполните задания.

Теоретическая часть

Существует 2 подхода при определении количества информации – **содержательный** (вероятностный) и **объемный** (алфавитный).

Содержательный применяется для измерения информации, используемой человеком, а алфавитный – компьютером.

Компьютер не понимает смысла информации, поэтому для её измерения нужен другой подход. Информация передаётся с помощью сигналов. Горит зелёный свет – можно переходить улицу, горит красный – стой. Поднял руку на уроке – учитель понял, что ты можешь ответить на его вопрос. В этих примерах сигнал имеет два состояния, из двух вариантов мы выбираем один.

Сообщение содержит информацию, если оно приводит к уменьшению неопределенности наших знаний.

Количество информации можно рассматривать как меру уменьшения неопределенности знания при получении информационных сообщений Информативным для человека является то сообщение, которое содержит новые и понятные сведения.

Главная формула информатики связывает между собой количество возможных информационных сообщений N и количество информации I, которое несет полученное сообщение:

$$N = 2^I$$

За единицу количества информации в теории информации принимается такое количество информации, которое содержится в информационном сообщении, уменьшающем неопределенность знания в два раза.

Такая единица названа **бит**.

Игра «Угадай число».

- Маша загадай число от 1 до 20. На мои вопросы отвечай больше/меньше.
- это число больше 10.
- это число больше 15.
- это число больше 18.

Вывод: я уменьшила неопределенность знаний в 2 разас и угадала число.

Бит – наименьшая единица измерения информации.

С помощью набора битов можно представить любой знак и любое число. Знаки представляются восьмиразрядными комбинациями битов – байтами.

1байт = 8 битов=2^3битов

Байт – это 8 битов, рассматриваемые как единое целое, основная единица компьютерных данных.

- Если у нас две двоичные цифры (бита), то число возможных комбинаций из них:

$2^2=4$: 00, 01, 10, 11

- Если четыре двоичные цифры (бита), то число возможных комбинаций:

$2^4=16$: 0000, 0001, 0010, 0011,
0100, 0101, 0110, 0111,
1000, 1001, 1010, 1011,
1100, 1101, 1110, 1111

Так как в байте- 8 бит (двоичных цифр), то число возможных комбинаций битов в байте:

$2^8=256$

Т.о., байт может принимать одно из 256 значений или комбинаций битов.

Для измерения информации используются более крупные единицы:

килобайты, мегабайты, гигабайты, терабайты и т.д.

1 Кбайт =1 024 байт

1 Мбайт = 1 024 Кбайт

1 Гбайт = 1 024 Мбайт

1 Тбайт = 1 024 Гбайт

Алфавитный подход к измерению информации

Суть алфавитного подхода к измерению информации определяется по количеству использованных для ее представления знаков некоторого алфавита. Например, если

при представлении числа XVIII использовано 5 знаков римского алфавита, то это и есть количество информации. То же самое число, т. е. ту же самую информацию, можно записать в десятичной системе (18). Как видим, получается 2 знака, т. е. другое значение количества информации. Алфавитный подход позволяет измерить количество информации в тексте, составленном из символов некоторого алфавита. Это объективный, количественный метод для измерения информации, циркулирующей в информационной технике.

Алфавит – конечный набор символов, используемых для представления информации.

Мощность алфавита – число символов в алфавите.

Например, алфавит десятичной системы счисления – множество цифр- 0,1,2,3,4,5,6,7,8,9.

Мощность этого алфавита – 10.

Компьютерный алфавит, используемый для представления текстов в компьютере, использует 256 символов.

Алфавит двоичной системы кодирования информации имеет всего два символа- 0 и 1.

Алфавиты русского и английского языков имеют различное число букв, их мощности – различны.

Информационный вес символа (количество информации в одном символе), выраженный в битах (i), и мощность алфавита (N) связаны между собой формулой:

$$\mathbf{N = 2^i}$$

где N – это количество знаков в алфавите знаковой системы или мощность

Информационная емкость знаков зависит от их количества в алфавите. Так, информационная емкость буквы в русском алфавите, если не использовать букву «ё», составляет:

$32 = 2^I$,

I = 5 битов

В латинском алфавите 26 букв. Информационная емкость буквы латинского алфавита также 5 битов.

Чем большее количество знаков в алфавите, тем большее количество информации несет 1 знак алфавита.

Количество информации в сообщении или информационный объём текста- Iс, равен количеству информации, которое несет один символ-I, умноженное на количество символов K в сообщении:

Iс = K * i бит

Например, в слове «информатика» 11 знаков (K=11), каждый знак в русском алфавите несет информацию 5 битов (I=5), тогда количество информации в слове «информатика» **Iс=5x11=55 (битов).**

С помощью формулы $N = 2^I$ можно определить количество информации, которое несет знак в двоичной знаковой системе: $N=2 \Rightarrow 2=2^I \Rightarrow 2^1=2^I \Rightarrow I=1$ бит

Таким образом, в двоичной знаковой системе 1 знак несет 1 бит информации. При двоичном кодировании объем информации равен длине двоичного кода.

Чем большее количество знаков содержит алфавит знаковой системы, тем большее количество информации несет один знак

Задания

1. Переведи 40 бит в байты.
2. У вас есть хранилище данных, объем которого составляет 5 петабайт. Сколько байтов содержится в этом хранилище?
3. Решите задачу: Считая, что один символ кодируется одним байтом, подсчитать в байтах количество информации, содержащееся в фразе: «Век живи, век учись.»
4. На экзамене вы берете экзаменационный билет, и преподаватель сообщает, что сообщение о его номере несет 4 бита информации. Определите количество экзаменационных билетов (запишите полное решение)
5. Песня содержит 32 ноты. Какое количество информации несет одна нота этой песни? (запишите полное решение)
6. Считая, что один символ кодируется одним байтом, подсчитать в байтах количество информации, содержащееся в фразе: «Не делай из мухи слона.»
7. Сообщение о том, что ваш друг живет на десятом этаже несет в себе 4 бита информации. Сколько этажей в доме?
8. Какое количество информации о цвете вынутого шарика будет получено, если в непрозрачном пакете хранятся: 10 белых, 20 красных, 30 синих и 40 зеленых шариков?

9. Найти объем информации, содержащейся в тексте из 3000 символов, написанном русскими буквами.

10. Для записи письма был использован алфавит мощностью в 16 символов. Письмо состояло из 25 строк. В каждой строке вместе с пробелами было 64 символа. Сколько байт информации содержало письмо?

11. Черно-белое изображение имеет 8 градаций яркости. Размер изображения 10*15 см. Разрешение 300 точек на дюйм (1 дюйм = 2,5 см). Сколько Кбайт памяти требуется для хранения изображения в несжатом виде?

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Список используемых источников
4. Выводы и предложения
5. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Определите количество информации с позиции «ИНФОРМАТИВНО» или «НЕИНФОРМАТИВНО»

- Столица России – Москва
- Сумма квадратов катетов равна квадрату гипотенузы
- Эйфелева башня весит 9000 тонн
- Дифракцией света называется совокупность явлений, которые обусловлены волновой природой света и наблюдаются при его распространении в среде с резко выраженной оптической неоднородностью

2. Есть ли связь между алфавитным подходом к измерению информации и содержанием информации?
3. Оцените объем одной страницы данного практического занятия в байтах.
4. Что такое бит с позиции алфавитного подхода к измерению информации?
5. Какие единицы используются для измерения объема информации на компьютерных носителях?
6. Какие существуют подходы к определению количества информации?
7. В чем состоит отличие одного подхода от другого?

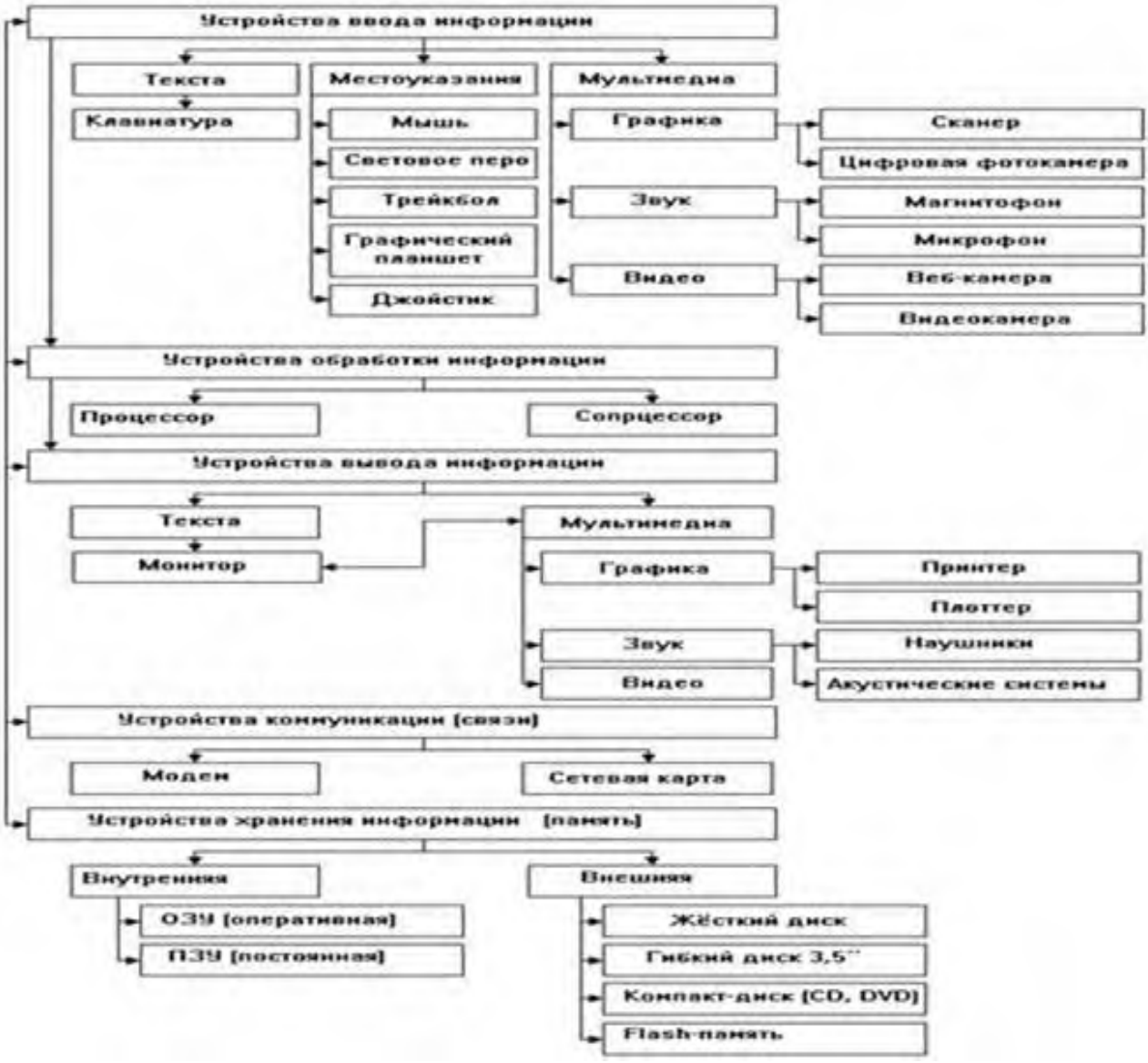
8. На какую величину отличается байт от Кб, Кб от Мб, Мб от Гб?
9. Что такое бит при смысловом и алфавитном подходе к определению количества информации?

Урок №2 Компьютер и цифровое представление информации. Устройство компьютера.

Цель занятия:

1.Познакомить обучающихся с аппаратным устройством компьютера и назначением основных частей компьютера.

Рис. 1. Структура аппаратного обеспечения ПК.



Иногда говорят «персональный компьютер». Уточнение «персональный» здесь не случайно – это значит свой, личный, доступный большинству людей, ведь существует большое количество других видов компьютеров, которые персональными никак не назвать – рабочие станции для предприятий, серверы для связи множества компьютеров в сеть и др. в дальнейшем, говоря «компьютер» мы будем иметь в виду именно персональный компьютер.

Персональный компьютер – это компьютер, предназначенный для обслуживания одного рабочего. По своим характеристикам он может отличаться от больших ЭВМ, но функционально способен выполнять аналогичные операции. По способу эксплуатации различают настольные, портативные и карманные модели ПК. В дальнейшем мы будем рассматривать настольные модели и приемы работы с ними. На современном рынке вычислительной техники разнообразие модификаций и вариантов компьютеров огромно, но любой, даже самый необычный комплект неизменно включает одни и те же виды устройств.

Базовая конфигурация ПК - минимальный комплект аппаратный средств, достаточный для начала работы с компьютером. В настоящее время для настольных ПК базовой считается конфигурация, в которую входит четыре устройства:

- Системный блок;
- Монитор;
- Клавиатура;
- Мышь.

Системный блок – основной блок компьютерной системы. В нем располагаются устройства, считающиеся внутренними. Устройства, подключающиеся к системному блоку снаружи, считаются внешними.

В системный блок входит процессор, оперативная память, накопители на жестких и гибких магнитных дисках, на оптический дисках и некоторые другие устройства. На лицевой панели вы видите несколько кнопок – уже известная вам кнопка Power – включения и кнопка Reset – перезагрузка компьютера, пользоваться которой можно лишь с разрешения учителя. Несколько световых индикаторов – включения и обращения к жесткому диску. Два дисковода – для компакт-дисков и дискет.

Монитор – устройство для визуального воспроизведения символьной и графической информации. Служит в качестве устройства вывода. Они отдаленно напоминают

бытовые телевизоры. В настольных компьютерах обычно используются мониторы на электронно-лучевой трубке (ЭЛТ). Изображение на экране монитора создается пучком электронов, испускаемых электронной пушкой. Этот пучок электронов разгоняется высоким электрическим напряжением (десятки киловольт) и падает на внутреннюю поверхность экрана, покрытую люминофором (веществом, светящимся под воздействием пучка электронов).

Система управления пучком заставляет пробегать его построчно весь экран (создает растр), а также регулирует его интенсивность (соответственно яркость свечения точки люминофора). Пользователь видит изображение на экране монитора, так как люминофор излучает световые лучи в видимой части спектра. Качество изображения тем выше, чем меньше размер точки изображения (точки люминофора), в высокачественных мониторах размер точки составляет 0,22 мм. Однако монитор является также источником высокого статического электрического потенциала, электромагнитного и рентгеновского излучений, которые могут оказывать неблагоприятное воздействие на здоровье человека. Современные мониторы практически безопасны, так как соответствуют жестким санитарно-гигиеническим требованиям, зафиксированным в международном стандарте безопасности TCO'99.

В портативных и карманных компьютерах применяют плоские мониторы на жидких кристаллах (ЖК). В последнее время такие мониторы стали широко использоваться и в настольных компьютерах.

LCD (Liquid Crystal Display, жидкокристаллические мониторы) сделаны из вещества, которое находится в жидком состоянии, но при этом обладает некоторыми свойствами, присущими кристаллическим телам. Фактически это жидкости, обладающие анизотропией свойств (в частности, оптических), связанных с упорядоченностью в ориентации молекул. Молекулы жидких кристаллов под воздействием электрического напряжения могут изменять свою ориентацию и вследствие этого изменять свойства светового луча, проходящего сквозь них. Преимущество ЖК-мониторов перед мониторами на ЭЛТ состоит в отсутствии вредных для человека электромагнитных излучений и компактности. Но ЖК-мониторы обладают и недостатками. Наиболее важные из них – это плохая цветопередача и смазывание быстро движущейся картинки. Иначе говоря, если взять достаточно качественный ЭЛТ-монитор, то он будет пригоден для любых задач без оговорок – для работы с текстом, для обработки фотографий, для игр и

так далее; в то же время среди ЖК-мониторов можно выделить модели, подходящие для игр – но они непригодны для работы с фотографиями, можно выделить модели, имеющие прекрасную цветопередачу – но они плохо подходят для динамичных игр, и так далее.

Мониторы могут иметь различный размер экрана. Размер диагонали экрана измеряется в дюймах (1 дюйм =2,54 см) и обычно составляет 15, 17, 19 и более дюймов.

Клавиатура – клавишное устройство, предназначенное для управления работой компьютера и ввода в него информации. Информация вводиться в виде алфавитно-цифровых символьных данных. Стандартная клавиатура имеет 104 клавиши и 3 информирующих о режимах работы световых индикатора в правом верхнем углу.

Мышь – устройство «графического» управления. При перемещении мыши по коврику на экране перемещается указатель мыши, при помощи которого можно указывать на объекты и/или выбирать их. Используя клавиши мыши (их может быть две или три) можно задать тот или другой тип операции с объектом. А с помощью колесика можно прокручивать вверх или вниз не умещающиеся целиком на экране изображения, текст или web-страницы. В оптико-механических мышах основным рабочим органом является массивный шар (металлический, покрытый резиной). При перемещении мыши по поверхности он вращается, вращение передается двум валам, положение которых считывается инфракрасными оптопарами (т.е. парами «светоизлучатель-фотоприемник») и затем преобразующийся в электрический сигнал, управляющий движением указателя мыши на экране монитора. Главным «врагом» такой мыши является загрязнение.

В настоящее время широкое распространение получили оптические мыши, в которых нет механических частей. Источник света размещенный внутри мыши, освещает поверхность, а отраженны свет фиксируется фотоприемником и преобразуется в перемещение курсора на экране. Современные модели мышей могут быть беспроводными, т.е. подключающимися к компьютеру без помощи кабеля.

Периферийными называют устройства, подключаемые к компьютеру извне. Обычно эти устройства предназначены для ввода и вывода информации.

Вот некоторые из них:

• Принтер;

• Сканер;

• Модем;

• Web-камера.

• DVB-карта и спутниковая антенна;

Принтер служит для вывода информации на бумажный носитель (бумагу). Существуют три типа принтеров: • матричный • струйный • лазерный

Матричные принтеры — это принтеры ударного действия. Печатающая головка матричного принтера состоит из вертикального столбца маленьких стержней (обычно 9 или 24), которые под воздействием магнитного поля «выталкиваются» из головки и ударяют по бумаге (через красящую ленту). Перемещаясь, печатающая головка оставляет на бумаге строку символов. Недостатки матричных принтеров состоят в том, что они печатают медленно, производят много шума и качество печати оставляет желать лучшего (соответствует примерно качеству пишущей машинки).

В последние годы широкое распространение получили черно-белые и цветные струйные принтеры. В них используется чернильная печатающая головка, которая под давлением выбрасывает чернила из ряда мельчайших отверстий на бумагу. Перемещаясь вдоль бумаги, печатающая головка оставляет строку символов или полоску изображения.

Струйные принтеры могут печатать достаточно быстро (до нескольких страниц в минуту) и производят мало шума. Качество печати (в том числе и цветной) определяется разрешающей способностью струйных принтеров, которая может достигать фотографического качества 2400 dpi. Это означает, что полоска изображения по горизонтали длиной в 1 дюйм формируется из 2400 точек (чернильных капель).

Лазерные принтеры обеспечивают практически бесшумную печать. Высокую скорость печати (до 30 страниц в минуту) лазерные принтеры достигают за счет постраничной печати, при которой страница печатается сразу целиком.

Высокое типографское качество печати лазерных принтеров обеспечивается за счет высокой разрешающей способности, которая может достигать 1200 dpi и более.

Плоттер. Для вывода сложных и широкоформатных графических объектов (плакатов, чертежей, электрических и электронных схем и пр.) используются специальные устройства вывода — плоттеры. Принцип действия плоттера такой же, как и струйного принтера.

Сканеры служат для автоматического ввода текстов и графики в компьютер.

Сканеры бывают двух типов: • ручные • планшетные.

Ручной сканер для компьютера похож на сканер, используемый в супермаркетах для считывания штрих-кода. Такой сканер перемещается по листу с информацией построчно вручную, и информация заносится в компьютер для дальнейшего редактирования. Планшетный сканер выглядит и работает примерно так же, как и ксерокс - приподнимается крышка, текст или рисунок помещается на рабочее поле, и информация считывается. Планшетные сканеры в наше время обычно все цветные. Системы распознавания текстовой информации позволяют преобразовать отсканированный текст из графического формата в текстовый.

Разрешающая способность сканеров составляет 600 dpi и выше.

Модем или модемная плата служит для связи удалённых компьютеров по телефонной сети. Модем бывает внутренний (установлен внутри системного блока) и внешний (располагается рядом с системным блоком и соединяется с ним при помощи кабеля.

DVB-карта и спутниковая антенна служат для так называемого «асинхронного» подключения компьютера к сети Интернет. При наличии DVB-карты и спутниковой антенны для соединения с Интернетом используется два канала связи: для передачи данных от пользователя используется модем, а для приема – спутниковый канал, скорость потока данных в котором в несколько раз превышает модемную.

Для организации на бескрайних Интернета видеоконференций (или просто болтовни) пригодится **Веб-камера**. С помощью этих устройств (и, естественно, быстрых локальных сетей), можно в любой момент устроить совещание со своими сотрудниками, не отрывая оных от насиженных рабочих мест. А это, как показывает практика, дает весьма ощутимую практическую пользу.

Оговоримся сразу — о настоящих видеокамерах здесь речи не идет. То есть можете даже и не мечтать о хорошей оптике, о качественной цветопередаче и тому подобной роскоши. Да и сохранять видеоизображение с веб-камеры вам и в голову не придет. Ведь нужен-то этот агрегат совсем для другого — обеспечивать поступление на ваш компьютер видеопотока с качеством и объемом, достаточным для передачи в Интернете.

Практически все веб-камеры рассчитаны на работу отнюдь не в медленном режиме модемного подключения. Подавай им цифровые каналы связи — и вот тогда-то эти устройства покажут себя во всей красе.

Что же касается России, то возможности обеспечить передачу такого потока данных в режиме реального времени, увы, пока нет. Ни у передающих устройств, ни у каналов связи. Поэтому максимум, на что сможет рассчитывать ваш собеседник — это появление вашей личности в крохотном окошке размером чуть поменьше сигаретной пачки (размер изображения — до 320x200 точек). Если этого вам достаточно, что ж, приобретение веб-камеры сможет чуть скрасить ваши серые компьютерные будни. Поскольку изображение веб-камера выдает не статичное, нужно учесть и другую важную величину — частоту обновления кадров. Так вот, на обычном, модемном соединении даже при крохотной картинке 150x200 точек искомых 24 кадров вы, скорее всего, не получите (реально — от 10 до 20). А значит, рывки и задержки неизбежны... Обратите внимание на другие показатели веб-камеры — реакция на различные условия освещения, наличие встроенного или дополнительного микрофона, длину соединительного USB-шнура, способность камеры работать «в связке» с популярными программами для голосового и видеообщения (например, Microsoft NetMeeting). И, конечно же, на максимальное разрешение: хотя качество картинки 640x480 точек уже давно стало стандартом, на рынке встречаются модели с куда более низким порогом разрешения (многие камеры стоимостью до 50 долл. обеспечивают разрешение лишь до 352x288 точек).
Кстати, а знаете ли вы, что хорошая веб-камера с успехом может заменить цифровой фотоаппарат? Большинство камер умеет не только передавать на компьютер поток видеоинформации, но и выдергивать из этого потока отдельные кадры-картинки. А вот их будущая судьба зависит от качества камеры: дорогие модели могут сохранять изображения во встроенной памяти, не требуя постоянного подключения к компьютеру, более же дешевые вынуждены сразу сбрасывать весь свой «груз» на жесткий диск.
Хотя, конечно, настоящий цифровой фотоаппарат работает гораздо лучше, и качество дает другое... Тем более что многие цифровые фотоаппараты средней ценовой категории также могут, в случае необходимости, поработать и веб-камерами.
И последнее. Почти все модели камер, выпущенные после 1999 г., подключаются к компьютер через разъем USB и не требуют дополнительного источника питания.

Внутренние устройства компьютеров.

Внутренними считаются устройства, располагающиеся в системном блоке. Доступ к некоторым из них имеется на лицевой панели, что удобно для быстрой смены

информационных носителей. Разъемы некоторых устройств выведены на заднюю стенку – они служат для подключения периферийного оборудования. К некоторым устройствам системного блока доступ не предусмотрен – для обычной работы он не требуется.

Материнская плата – самая большая плата ПК. На ней располагаются магистрали, связывающие процессор с оперативной памятью, - так называемые шины. К шинам материнской платы подключаются также все прочие внутренние устройства компьютера. Управляет работой материнской платы микропроцессорный набор микросхем – так называемый чипсет.

На компьютерном жаргоне материнская плата называется «мама», «мать».

Процессор. Микропроцессор – основная микросхема ПК. Все вычисления выполняются в ней. Процессор аппаратно реализуется на большой интегральной схеме (БИС). Большая интегральная схема на самом деле не является большой по размеру и представляет собой, наоборот, маленькую плоскую полупроводниковую пластину размером примерно 20х20 мм, заключенную в плоский корпус с рядами металлических штырьков (контактов). БИС является большой по количеству элементов. Использование современных высоких технологий позволяет разместить на БИС процессора огромное количество функциональных элементов, размеры которых составляют всего около 0.13 микрон (1 микрон = 10-6 м). Например, в процессоре Pentium 4 их около 42 миллионов.

Характеристика	Pentium II	Pentium III	Pentium 4
Число транзисторов	7.5 млн.	9.5 млн.	42 млн.
Тактовая частота	0.45 ГГц	1 ГГц	1.4 ГГц и выше

В настоящее время для производства процессоров семейства AMD64 корпорация AMD переходит на использование новейшего техпроцесса - HiPerMOS 8 (HiP8). Этот технологический процесс, по которому в дальнейшем будут выпускаться все процессоры AMD, включает в себя ряд наиболее прогрессивных технологий для производства интегральных схем: 90 нм технологическое разрешение, медные проводники, low-k диэлектрик и технологию SOI.

Переход на технологическое разрешение 90 нм позволяет выпускать процессоры с высокой тактовой частотой, потребляющие меньше энергии и имеющие небольшое

тепловыделение. Кроме того, использование транзисторов с меньшей длиной затвора позволяет на подложке равной площади разместить большее число транзисторов, например процессор AMD Opteron с кэшем L2 1 Мб, состоит более чем из 106 млн. транзисторов. Освоение 90 нм техпроцесса - важный шаг к производству двуядерных процессоров. Снижение уровня тепловыделения и уменьшения площади ядра при переходе на 90 нм техпроцесс позволит сохранить мощность двуядерных процессоров на прежнем уровне, при этом общая площадь кристалла увеличится незначительно.

Основная характеристика процессора – тактовая частота (измеряется в мегагерцах (МГц) и гигагерцах (ГГц)). Чем выше тактовая частота, тем выше производительность компьютера. Так, например, при тактовой частоте 2000 МГц процессор может за одну секунду изменить свое состояние 2000 миллионов раз. Для большинства операций одного такта недостаточно, поэтому количество операций, которые процессор может выполнить в секунду, зависит не только от тактовой частоты, но и от сложности операций.

Есть еще насколько важных характеристик процессора – тип ядра и технология производства, частота системной шины, но мы их рассматриваем пока не будем.

Единственное устройство, о существовании которого процессор «знает от рождения» – оперативная память – с нею он работает совместно. Данные копируются в ячейки процессора (регистры), а затем преобразуются в соответствии с командами (программой).

Оперативная память (ОЗУ), предназначена для хранения информации, изготавливается в виде модулей памяти. Оперативную память можно представить как обширный массив ячеек, в которых хранятся данные и команды в то время, когда компьютер включен. Процессор может обратится к любой ячейки памяти.

Важнейшей характеристикой модулей памяти является быстродействие. Модули памяти могут различаться между собой по размеру и количеству контактов, быстродействию, информационной емкостью и т.д.

Может возникнуть вопрос - почему бы не использовать для хранения промежуточных данных жесткий диск, ведь его объем во много раз больше? Это делать нельзя, так как скорость доступа к оперативной памяти у процессора в сотни тысяч раз больше, чем к дисковой.

Для длительного хранения данных и программ широко применяются жесткие диски (винчестеры). Выключение питания компьютера не приводит к очистке внешней

памяти. **Жесткий диск** – это чаще не один диск, а пакет (набор) дисков с магнитным покрытием, вращающихся на общей оси. Основным параметром является емкость, измеряемая в гигабайтах. Средний размер современного жесткого диска составляет 80 — 160 Гбайт, причем этот параметр неуклонно растет.

Винчестером он сначала в шутку был назван в 1973 году, так как некоторые его технические характеристики по названию походили на марку знаменитой винтовки «винчестер». С тех пор название прижилось.

Не смотря на свое название жесткий диск – весьма хрупкий прибор, требующий бережного отношения. Он боится ударов, толчков, резких перепадов температуры и влажности, а так же сильных магнитных полей. Перемещение жестких дисков необходимо выполнять с повышенной внимательностью и аккуратностью.

Видеоадаптер – внутренне устройство, устанавливается в один из разъемов материнской платы, и служит для обработки информации, поступающей от процессора или из ОЗУ на монитор, а также для выработки управляющих сигналов.

В первых персональных компьютерах видеоадаптеров не было. Вместо них в оперативной памяти отводилась небольшая область для хранения видеоданных. Специальная микросхема (видеоконтроллер) считывала данные из ячеек видеопамяти и в соответствии с ними управляла монитором. По мере улучшения графических возможностей компьютеров область видеопамяти отделили от основной оперативной памяти и вместе с видеоконтроллером выделили в отдельный прибор, который назвали видеоадаптером. Современные видеоадаптеры имеют собственный вычислительный процессор (видеопроцессор), который снизил нагрузку на основной процессор при построении сложных изображений. Особенно большую роль видеопроцессор играет при построении на плоском экране трехмерных изображений. В ходе таких операций ему приходится выполнять особенно много математических расчетов.

В некоторых моделях материнских плат функции видеоадаптера выполняют микросхемы чипсета — в этом случае говорят, что видеоадаптер интегрирован с материнской платой. Если же видеоадаптер выполнен в виде отдельного устройства, его называют видеокартой. Разъем идеокарты выведен на заднюю стенку. К нему подключается монитор.

Звуковой адаптер. Для компьютеров IBM PC работа со звуком изначально не была предусмотрена. Первые десять лет существования компьютеры этой платформы считались офисной техникой и обходились без звуковых устройств. В настоящее

время средства для работы со звуком считаются стандартными. Для этого на материнской плате устанавливается звуковой адаптер. Он может быть интегрирован в чипсете материнской платы или выполнен как отдельная подключаемая плата, которая называется звуковой картой.

Разъемы звуковой карты выведены на заднюю стенку компьютера. Для воспроизведения звука к ним подключают звуковые колонки или наушники. Отдельный разъем предназначен для подключения микрофона. При наличии специальной программы это позволяет записывать звук. Имеется также разъем (линейный выход) для подключения к внешней звукозаписывающей или звуковоспроизводящей аппаратуре (магнитофонам, усилителям и т.п.).

Сетевая карта (или карта связи по локальной сети) служит для связи компьютеров в пределах одного предприятия, отдела или помещения находящихся на расстоянии не более 150 метров друг от друга.

При наличии специальных дополнительных устройств можно организовать связь компьютеров и на большие расстояния.

Основным параметром сетевой карты является скорость передачи информации и измеряется она в мегабайтах в секунду. Типовая норма от 10 до 100 мегабайт в секунду.

Коммуникационные порты. Для связи с другими устройствами, например принтером, сканером, клавиатурой, мышью и т. п., компьютер оснащается так называемыми портами. Порт — это не просто разъем для подключения внешнего оборудования, хотя порт и заканчивается разъемом. Порт — более сложное устройство, чем просто разъем, имеющее свои микросхемы и управляемое программно.

Примеры портов:

• COM (последовательный порт)

• LTP (параллельный порт)

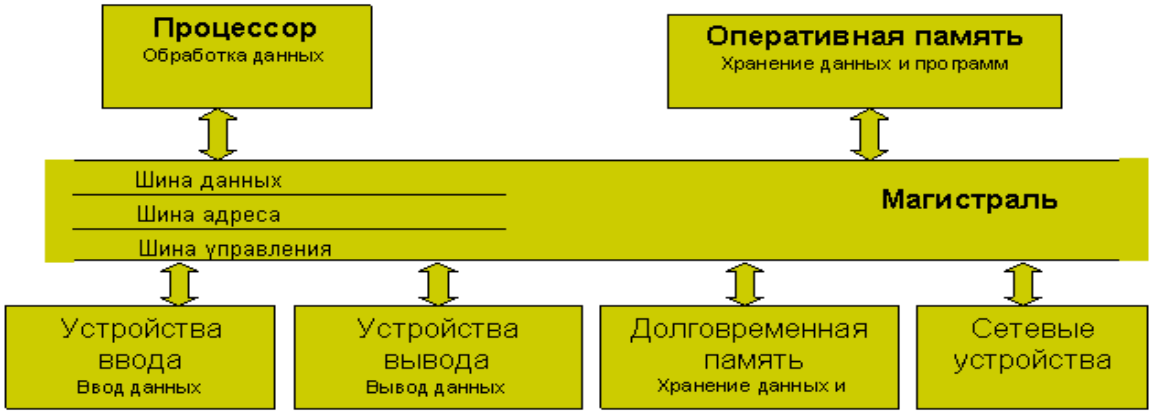
• USB (последовательный с высокой производительностью)

• PS/2 (универсальный для подключения мыши и клавиатуры)

Через последовательные порты данные передаются последовательно байт за байтом. Предельное значение производительности последовательного порта – 112 Кбит/с. Этого недостаточно для передачи больших объемов данных, поэтому к последовательным портам подключают устройства, не требующие высокой производительности: модемы, мыши, устаревшие модели принтеров.

Через параллельный порт передаются одновременно все восемь битов, составляющих один байт. Предельное значение производительности параллельного порта – 5 Мбайт/с. К этому порту, как правило, подключается принтер.
Преимущество параллельного порта от последовательного заключается еще в возможности использования более длинных кабелей для соединения (до 10 м против 1.5 м).
Все современные компьютеры комплектуются портами нового поколения – USB. Это порты последовательного типа, но с высокой производительностью (до 12 Мбайт/с). Кроме высокой производительности к достоинствам USB портов относится удобство работы с ними: не требуется выключать оборудование перед стыковкой, возможно подключение нескольких устройств в одному порту. Многие модели современной периферийного оборудования могут подключаться к портам этого типа.
Кроме универсальных коммуникационных портов, предназначенных для любого оборудования, компьютер имеет два специализированных порта для подключения мыши и клавиатуры – это порты PS/2. Другие устройства к этим портам не подключаются.

Информация, представленная в цифровой форме и обрабатываемая на компьютере, называется данными. Последовательность команд, которую выполняет компьютер в процессе обработки данных, называется программой.
Обработка данных на компьютере:
1. Пользователь запускает программу, хранящуюся в долговременной памяти, она загружается в оперативную и начинает выполняться.
2. Выполнение: процессор считывает команды и выполняет их. Необходимые данные загружаются в оперативную память из долговременной памяти или вводятся с помощью устройств ввода.
3. Выходные (полученные) данные записываются процессором в оперативную или долговременную память, а также предоставляются пользователю с помощью устройств вывода информации.
Для обеспечения информационного обмена между различными устройствами должна быть предусмотрена какая-то магистраль для перемещения потоков информации.



Магистраль (системная шина) включает в себя три многоразрядные шины: шину данных, шину адреса и шину управления, которые представляют собой многопроводные линии. К магистрали подключаются процессор и оперативная память, а также периферийные устройства ввода, вывода и хранения информации, которые обмениваются информацией на машинном языке (последовательностями нулей и единиц в форме электрических импульсов).

Шина данных. По этой шине данные передаются между различными устройствами. Например, считанные из оперативной памяти данные могут быть переданы процессору для обработки, а затем полученные данные могут быть отправлены обратно в оперативную память для хранения. Таким образом, данные по шине данных могут передаваться от устройства к устройству в любом направлении. Разрядность шины данных определяется разрядностью процессора, то есть количеством двоичных разрядов, которые могут обрабатываться или передаваться процессором одновременно. Разрядность процессоров постоянно увеличивается по мере развития компьютерной техники.

Шина адреса. Выбор устройства или ячейки памяти, куда пересылаются или откуда считываются данные по шине данных, производит процессор. Каждое устройство или ячейка оперативной памяти имеет свой адрес. Адрес передается по адресной шине, причем сигналы по ней передаются в одном направлении — от процессора к оперативной памяти и устройствам (однонаправленная шина).

Разрядность шины адреса определяет объем адресуемой памяти (адресное пространство), то есть количество однобайтовых ячеек оперативной памяти, которые могут иметь уникальные адреса.

Шина управления. По шине управления передаются сигналы, определяющие характер обмена информацией по магистрали. Сигналы управления показывают,

какую операцию — считывание или запись информации из памяти — нужно производить, синхронизируют обмен информацией между устройствами и так далее. Модульный принцип позволяет потребителю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости ее модернизацию. Каждая отдельная функция компьютера реализуется одним или несколькими модулями – конструктивно и функционально законченных электронных блоков в стандартном исполнении. Организация структуры компьютера на модульной основе аналогична строительству блочного дома. Основными модулями компьютера являются память и процессор. Процессор – это устройство управляющее работой всех блоков компьютера. Действия процессора определяются командами программы, хранящейся в памяти.

Модульная организация опирается на магистральный (шинный) принцип обмена информацией между устройствами.

Магистрально-модульный принцип имеет ряд достоинств:

1. для работы с внешними устройствами используются те же команды процессора, что и дл работы с памятью.
2. подключение к магистрали дополнительных устройств не требует изменений в уже существующих устройствах, процессоре, памяти.
3. меняя состав модулей можно изменять мощность и назначение компьютера в процессе его эксплуатации.

Принцип открытой архитектуры – правила построения компьютера, в соответствии с которыми каждый новый блок должен быть совместим со старым и легко устанавливаться в том же месте в компьютере.

В компьютере столь же легко можно заменить старые блоки на новые, где бы они ни располагались, в результате чего работа компьютера не только не нарушается, но и становится более производительной. Этот принцип позволяет не выбрасывать, а модернизировать ранее купленный компьютер, легко заменяя в нем устаревшие блоки на более совершенные и удобные, а так же приобретать и устанавливать новые блоки. Причем во всех разъемы для их подключения являются стандартными и не требуют никаких изменений в самой конструкции компьютера.

Вопросы для самопроверки:

1. Что главное в этой конфигурации?

2. Кто из вас разбирал системный блок и знает его строение?
3. Что самое главное в системном блоке?
4. Что значит магистрально-модульный принцип?

Практическое занятие №3 Логические основы компьютера

Цель занятия:

1.изучение логических элементов компьютера и их таблиц истинности.
2.построение таблиц истинности логических высказываний.

3.применять основные формулы и правила алгебры логики

4. научиться строить логические схемы

Исходные данные: теоретический материал, таблицы истинности

Содержание и порядок выполнения задания:

Изучить теоретическую часть

Выполнить задания

Теоретическая часть

Алгебра логики появилась в середине XIX в. в трудах английского математика Джорджа Буля.

Логическое высказывание – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинного оно или ложно. При этом не всякое предложение является логическим высказыванием. Например, «Хороший студент» не является логическим высказыванием, так как невозможно судить об его истинности или ложности, а высказывание «Иванов – хороший студент» является логическим высказыванием.

Употребляемые слова и словосочетания «*не*», «*и*», «*или*», «*если, то*», «*тогда и только тогда*» позволяют из уже заданных высказываний строить новые. Такие слова и словосочетания называю*тся логическими связками*. При этом высказывания, образованные из других высказываний с помощью логических связок, называются *составными*. Высказывание, не являющееся составным, *называется элементарным*.

Для того чтобы обращаться к логическим высказываниям им назначаются имена. Например, через А обозначим высказывание «Петя был во Франции», а через В высказывание «Петя был в Италии». Тогда составное высказывание примет вид - «Петя был и во Франции, и в Италии», далее можно записать кратко: «А И В». Здесь И – логическая связка, А,В – логические переменные, которые могут принимать только два значения: истинна и ложь, обозначаемые 1 и 0.

В алгебре логики высказывания могут принимать лишь два значения: истинна – 1 и ложь – 0.

Работу логических элементов описывают с помощью таблиц истинности.

Таблица истинности – это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

1. Операции с логическими высказываниями

С логическими высказываниями можно производить следующие операции:

1. Операция отрицания.

Операция, выражаемая словом НЕ, называется отрицанием, или инверсией, и обозначается чертой над высказыванием или знаком ¬.

Таблица 1. Таблица истинности логического отрицания

A	$\overline{A}$
0	1
1	0

2. Операция конъюнкции.

Операция, выражаемая связкой И, называется соединением, или конъюнкцией, или логическим умножением, и обозначается знаком «&» (может обозначаться знаком «∧» или «·». Высказывание F=A&B истинно только тогда, когда оба высказывания A и B истинны.

Таблица 2. - Таблица истинности логического умножения

A	B	F=A&B
0	0	0
0	1	0
1	0	0
1	1	1

3. Операция дизъюнкции.

Операция, выражаемая связкой ИЛИ, называется разделением, или дизъюнкцией, или логическим сложением, и обозначается знаком «∨» (или «+»). Высказывание F=A∨B ложно тогда и только тогда, когда оба высказывания A и B ложны.

Таблица3. -Таблица истинности логического сложения

A	B	F=A∨B
0	0	0
0	1	1
1	0	1
1	1	1

4. Операция импликации.

Операция следования, выражаемая связками «если…, то», «из … следует», «… влечет … », называется импликацией и обозначается знаком «→». Высказывание A→B ложно тогда и только тогда, когда A истинно, а B ложно.

Таблица 4 - Таблица истинности логической функции импликации

A	B	F=A→B
0	0	1
0	1	1
1	0	0
1	1	1

5. Операция эквиваленции.

Операция равенства, выражаемая связками «тогда и только тогда», «необходимо и достаточно», «… равносильно …», называется эквиваленцией, или двойной импликацией, и обозначается знаками «↔» или «~». Высказывание A↔B истинно тогда и только тогда, когда значения A и B совпадают.

Таблица 5 - Таблица истинности эквиваленции

A	B	F=A↔B
0	0	1
0	1	0
1	0	0
1	1	1

Порядок выполнения логических операций задается круглыми скобками, но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицание (НЕ), затем конъюнкции (И), затем дизъюнкции (ИЛИ) и в последнюю очередь - импликации. Это называется приоритетом операций.

Порядок выполнения логических операций в сложном логическом выражении:

1. Инверсия;
2. Конъюнкция;
3. Дизъюнкция;

4. Импликация;

5. Эквивалентность.

Для изменения указанного порядка выполнения логических операций используются скобки.

2. Основные законы алгебры логики

В алгебре логики выполняются следующие основные законы, позволяющие проводить тождественные преобразования (упрощения) логических выражений, показанные в таблице 2.1. Законы алгебры логики

Закон	для «ИЛИ»	для «И»
1. Двойного отрицания	$A=\overline{\overline{A}}$	
2. Переместительный	$A \vee B=B \vee A$	$A\&B=B\&A$
3. Сочетательный	$(A \vee B) \vee C=A \vee (B \vee C)$	$(A\&B)\&C=A\&(B\&C)$
4. Распределительный	$(A \vee B)\&C=(A\&C) \vee (B\&C)$	$(A\&B) \vee C=(A \vee C)\&(B \vee C)$
5. Законы де Моргана	$\overline{A \vee B}=\overline{A}\&\overline{B}$	$\overline{A\&B}=\overline{A} \vee \overline{B}$
6. Идемпотентности	$A \vee A=A$	$A\&A=A$
7. Исключения констант	$A \vee 1=1;\ A \vee 0=A$	$A\&1=A;\ A\&0=0$
8. Противоречия	—	$A\&\overline{A}=0$
9. Исключение третьего	$A \vee \overline{A}=1$	—
10. Поглощения	$A \vee (A\&B)=A$	$A\&(A \vee B)=A$
11. Исключения	$(A\&B) \vee (\overline{A}\&B)=B$	$(A \vee B)\&(\overline{A} \vee B)=B$
12. Контрапозиции	$(A \leftrightarrow B)=(B \leftrightarrow A)$	

Задания

№ 1. Какие из предложений являются высказываниями? Определите их истинность. Определите тип высказывания: общее, частное или единичное.

1. Все солдаты храбрые
2. Некоторые ученики двоечники
3. Все ананасы приятны на вкус
4. Некоторые мои друзья собирают марки
5. Все лекарства неприятны на вкус
6. А — первая буква в алфавите
7. Некоторые медведи — бурые
8. Тигр — хищное животное
9. У некоторых змей нет ядовитых зубов
10. Все металлы проводят тепло

A	B	¬A инверсия	AVB дизъюнкция	A&B конъюнкция	A→B импликация	A↔B эквиваленция
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Пример. Какое из приведенных имен удовлетворяет логическому условию:

¬ (последняя буква гласная → первая буква согласная) & вторая буква согласная

1) ИРИНА 2) АРТЕМ 3) СТЕПАН 4) МАРИЯ

Имя	X1: последняя буква гласная	X2: первая буква согласная	X3: вторая буква согласная	X1→X2	¬(X1→X2)	¬(X1→X2)&X3
Ирина	1	0	1	0	1	1
Артём	0	0	1	1	0	0
Степан	0	1	1	1	0	0
Мария	1	1	0	1	0	0

№ 2. а) Какое из приведенных названий животных удовлетворяет логическому условию

¬(есть мягкий знак & (вторая буква гласная → пятая буква согласная))

1) МЕДВЕДЬ 2) ВЫХУХОЛЬ 3) МУРАВЬЕД 4) ОБЕЗЬЯНА

б) Какое из приведенных имен удовлетворяет логическому условию

¬(первая буква гласная → последняя буква гласная) & вторая буква согласная

1) ИРИНА 2) ОЛЕГ 3) СТЕПАН 4) ИЛОНА

в) Какое из приведенных имен удовлетворяет логическому условию:

¬ (первая буква согласная → вторая буква согласная) & (предпоследняя буква гласная → последняя буква гласная)

1) КРИСТИНА 2) МАКСИМ 3) СТЕПАН 4) МАРИЯ

Решение логических выражений принято записывать в виде **таблиц истинности** – таблиц, в которых по действиям показано, какие значения принимает логическое выражение при всех возможных наборах его переменных.

При составлении таблицы истинности для логического выражения необходимо учитывать **порядок выполнения логических операций**, а именно:

1. действия в скобках,

2. инверсия (**отрицание**),
3. & (**конъюнкция**),
4. v (**дизъюнкция**),
5. => (**импликация**),
6. <=> (**эквивалентность**).

Алгоритм составления таблицы истинности:

1. Выяснить количество строк в таблице (вычисляется как 2^n, где n – количество переменных + строка заголовков столбцов).

2. Выяснить количество столбцов (вычисляется как количество переменных + количество логических операций).

3. Установить последовательность выполнения логических операций.

4. Построить таблицу, указывая названия столбцов и возможные наборы значений исходных логических переменных.

5. Заполнить таблицу истинности по столбцам.

6. Записать ответ.

Примеры.

1. **Составим таблицу истинности** для формулы **«А или В или не В»,** которая содержит две переменные А и В. В первых двух столбцах таблицы запишем четыре возможных пары значений этих переменных, в последующих столбцах — значения промежуточных формул и в последнем столбце — значение формулы.

Формулу перепишем в терминах алгебры логики: A∨B∨$\overline{B}$

В результате получим таблицу:

Переменные		Промежуточные логические формулы		Формула
A	B	$\overline{B}$	B∨A	A∨B∨$\overline{B}$
0	0	1	0	1
1	0	1	1	1
0	1	0	1	1
1	1	0	1	1

Т.е. формула $A \vee B \vee \overline{B}$ является **тождественно истинной**.

Пример 2

Построим таблицу истинности для выражения **F=(AvB)&(¬Av¬B)**.

1. Количество строк=2^2 (2 переменных+строка заголовков столбцов)=5.

2. Количество столбцов=2 логические переменные (A, B)+ 5 логических операций (v,&,¬,v,¬) = 7.

3. Расставим порядок выполнения операций: 1 5 2 4 3

(A v B)&(¬A v ¬B)

4-5. Построим таблицу и заполним ее по столбцам:

A	B	AvB	¬A	¬B	¬Av¬B	(AvB)&(¬Av¬B)
0	**0**	0	1	1	1	**0**
0	1	1	1	0	1	1
1	0	1	0	1	1	1
1	**1**	1	0	0	0	**0**

6. Ответ: F=0, при A=B=0 и A=B=1

Пример 3

Построим таблицу истинности для логического выражения **F=XvY&¬Z**.

1. Количество строк=2^3+1=(3 переменных+строка заголовков столбцов)=9.

2. Количество столбцов=3 логические переменные+3 логических операций = 6.

3. Укажем порядок действий: 3 2 1

X v Y & ¬Z

4-5. Построим таблицу и заполним ее по столбцам:

X	Y	Z	¬Z	Y&¬Z	XvY&¬Z
0	**0**	**0**	1	0	**0**
0	**0**	**1**	0	0	**0**
0	1	0	1	1	1
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

6. **Ответ:** F=0, при X=Y=Z=0; при X=Y=0 и Z=1.

Задание:

Построить таблицу истинности функции F

1	$F = A \vee \overline{B} \vee (\overline{A} \vee C)$	16	$F = A \leftrightarrow C \vee B \rightarrow A$
2	$F = A \rightarrow \overline{B} \vee C$	17	$F = A \leftrightarrow \overline{C} \vee B \rightarrow \overline{A}$
3	$F = B \vee (\overline{A} \leftrightarrow C)$	18	$F = (A \leftrightarrow C) \vee (B \rightarrow A)$
4	$F = \overline{B} \vee (A \leftrightarrow C)$	19	$F = A \leftrightarrow C \vee (B \rightarrow \overline{A})$
5	$F = A \wedge B \rightarrow \overline{B} \wedge C$	20	$F = A \leftrightarrow (C \vee B \rightarrow A)$
6	$F = A \wedge B \leftrightarrow \overline{B} \vee C$	21	$F = (\overline{A} \leftrightarrow C) \vee B \rightarrow A$
7	$F = (A \vee \overline{B}) \vee (\overline{A} \rightarrow C)$	22	$F = \overline{A} \leftrightarrow (C \vee \overline{B} \rightarrow A)$
8	$F = (A \rightarrow \overline{B}) \vee C$	23	$F = A \wedge (B \rightarrow \overline{C}) \wedge C$
9	$F = B \vee C \leftrightarrow \overline{A} \vee \overline{C}$	24	$F = A \wedge (B \leftrightarrow \overline{A}) \vee C$
10	$F = \overline{B} \vee (A \wedge C \rightarrow B)$	25	$F = (C \vee \overline{B}) \vee (\overline{A} \vee C)$
11	$F = A \vee B \rightarrow \overline{B} \vee C$	26	$F = A \rightarrow \overline{B} \vee (C \rightarrow B)$
12	$F = A \wedge B \leftrightarrow \overline{B} \vee C$	27	$F = (A \wedge B \rightarrow \overline{B}) \wedge (C \vee \overline{A})$
13	$F = A \rightarrow \overline{B} \vee (\overline{A} \vee C)$	28	$F = \overline{B} \vee (A \leftrightarrow C) \wedge C$
14	$F = \overline{A} \wedge B \rightarrow \overline{B} \vee C$	29	$F = A \wedge B \rightarrow \overline{B} \wedge C$
15	$F = B \vee (\overline{A} \leftrightarrow C) \wedge A$	30	$F = A \wedge B \leftrightarrow \overline{B} \vee C$

Логический элемент компьютера – это часть электронной логической схемы, которая реализует элементарную логическую функцию.

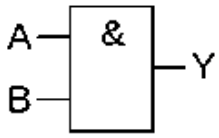
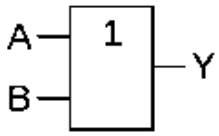
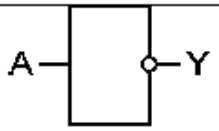
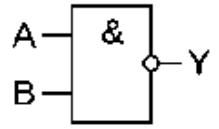
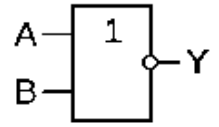
Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ и др. (называемые также вентилями), а также триггер.

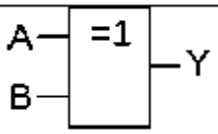
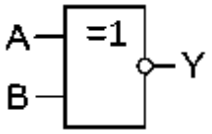
С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у элементов бывает от 2 до 8 входов и один или два выхода. Чтобы представить два логических состояния 1 и 0, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения, например 5 и 0 В. Высокий уровень обычно соответствует значению «истинна» (1), а низкий – значению «ложь» (0).

Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая электронная схема в нем реализована. Это упрощает запись и понимание сложных схем.

Работу логических элементов описывают с помощью таблиц истинности. Основные структурные схемы логических элементов компьютера и их таблицы истинности, представлены в таблице 3.1.

Таблица 3.1. - Структурные схемы логических элементов компьютера

Условное обозначение	Структурная схема (отечественное обозначение)	Структурная схема (зарубежное обозначение)	Таблица истинности
И			A B Y (A&B) 0 0 0 0 1 0 1 0 0 1 1 1
ИЛИ			A B Y (A∨B) 0 0 0 0 1 1 1 0 1 1 1 1
НЕ			A Y ($\bar{A}$) 0 1 1 0
И-НЕ			A B Y 0 0 1 0 1 1 1 0 1 1 1 0
ИЛИ-НЕ			A B Y 0 0 1 0 1 0 1 0 0 1 1 0
Исключающее ИЛИ			A B Y 0 0 0 0 1 1

			1 0 1 1 1 0
Исключающее ИЛИ-НЕ			A B Y 0 0 1 0 1 0 1 0 0 1 1 1

Почему необходимо уметь строить логические схемы?

Дело в том, что из вентилей составляют более сложные схемы, которые позволяют выполнить арифметические операции и хранить информацию. Причем схему, выполняющую определенные функции, можно построить из различных по сочетанию и количеству вентилей. Поэтому значение формального представления логической схемы чрезвычайно велико. Оно необходимо для того, чтобы разработчик имел возможность выбрать наиболее подходящий ему вариант построения схемы из вентилей. Процесс разработки общей логической схемы устройства (в том числе и компьютера в целом) таким образом становится иерархическим, причем на каждом следующем уровне в качестве «кирпичиков» используются логические схемы, созданные на предыдущем этапе.

Алгебра логики дала в руки конструкторам мощное средство разработки, анализа и совершенствования логических схем. В самом деле, гораздо проще, быстрее и дешевле изучать свойства и доказывать правильность работы схемы с помощью выражающей ее формулы, чем создавать реальное техническое устройство. Именно в этом состоит смысл любого математического моделирования.

Логические схемы необходимо строить из минимально возможного количества элементов, что в свою очередь, обеспечивает большую скорость работы и увеличивает надежность устройства.

Алгоритм построения логических схем:

1) Определить число логических переменных.

2) Определить количество базовых логических операций и их порядок.

3) Изобразить для каждой логической операции соответствующий ей вентиль.

4) Соединить вентили в порядке выполнения логических операций.

Пример 1

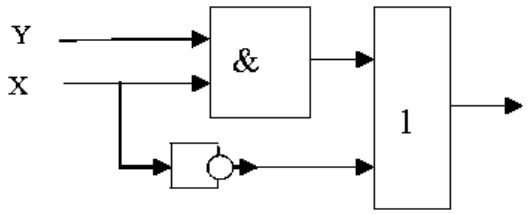
Составить логическую схему для логического выражения: F=¬XvY&X.

1) Две переменные – X и Y.

2) Две логические операции: 1 3 2

¬XvY&X.

3) Строим схему, соединяя вентили в порядке выполнения логических операций:



Пример 2

Постройте логическую схему, соответствующую логическому выражению F=X&Yv¬(YvX).

Вычислить значения выражения для X=1, Y=0.

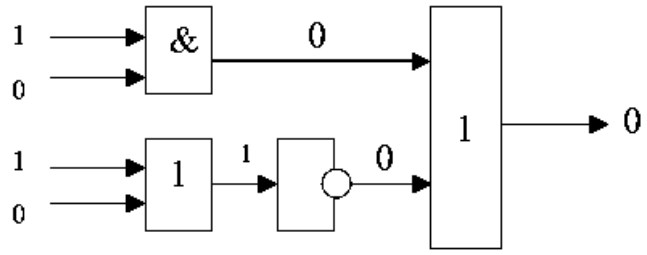
1) Переменных две: X и Y.

2) Логических операций четыре: конъюнкция, две дизъюнкции и отрицание. Определяем порядок выполнения операций:

1 4 3 2

X&Yv ¬(YvX).

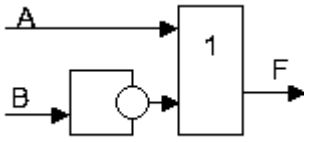
3) Схему строим слева направо в соответствии с порядком выполнения логических операций:



4) Вычислим значение выражения: F=1&0v¬(0v1)=0.

Выполним обратное задание. Дана логическая схема. Необходимо построить по ней логическое выражение.

Пример - Задание 1.

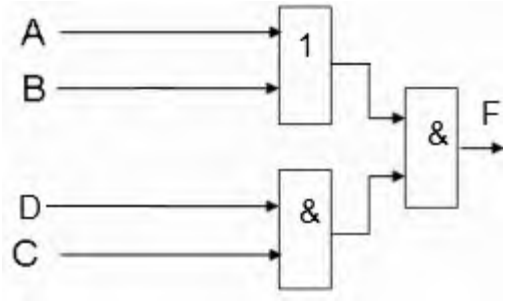


Логических переменных на схеме две (A и B), вентилей два: один инвертор, один дизъюнктор. Строим логическую схему в порядке выполнения логических операций.

F = ¬ B V A

Пример - Задание 2.

Постройте логическую схему, соответствующую логическому выражению и найдите значение логического выражения.



F = (A V B) & (D& C)

Задание 1

Постройте логическую схему, соответствующую логическому выражению, и найдите значение логического выражения:

1) $F = (A\&B)\vee(B\&C)\vee(\overline{A}\&C)$

2) $F = \overline{(A\&B)\vee(B\vee C)}\&(\overline{A}\&C)$

3) $F = \overline{(A\&B)\vee(B\&C)}\vee(\overline{A\&C})$

4) $F = \overline{(A\&B)\vee(B\&C)}\vee(\overline{A}\&C)$

5) $F = (\overline{A\&B})\vee(B\&C)\vee(\overline{A}\&C)$

6) $F = \overline{(A\&\bar{B})\vee(B\vee C)}\&(\overline{A}\&C)$

7) $F = \overline{(A\&B)\vee(B\&C)}\vee(A\&C)$

8) $F = \overline{(A\&B)\vee(B\&C)}\vee(\overline{A}\&\bar{C})$

9) $F = (A\&B)\vee(\overline{B\&C})\vee(\overline{A}\&C)$

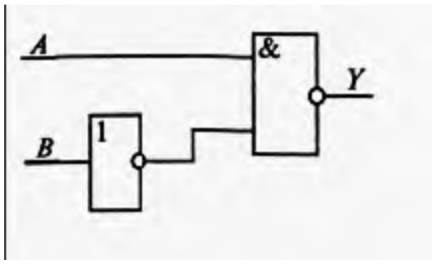
10) $F = \overline{(A\&B)\vee(B\vee\bar{C})}\&(\overline{A}\&C)$

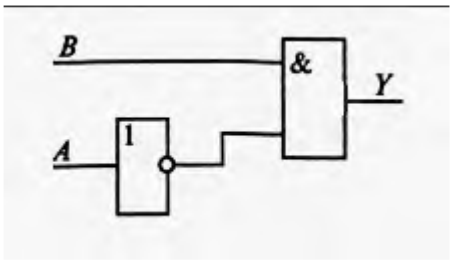
11) $F = (A\&B)\vee(B\&C)\vee(\overline{A\&C})$

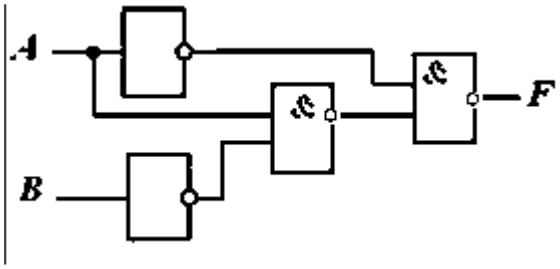
12) $F = \overline{(\bar{A}\&B)\vee(B\&C)}\vee(\overline{A}\&C)$

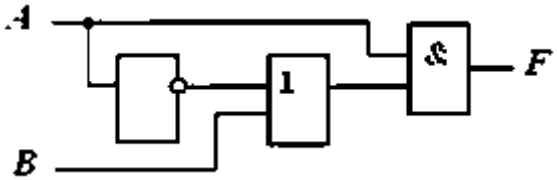
Задание 2

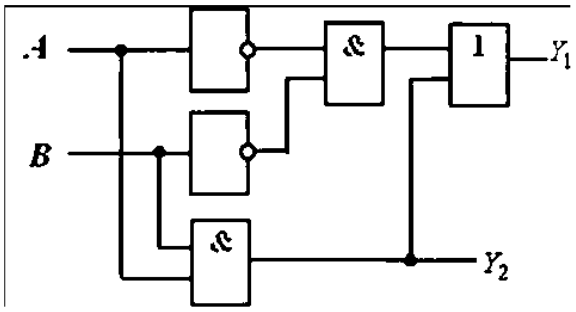
По логической схеме составить логическую функцию

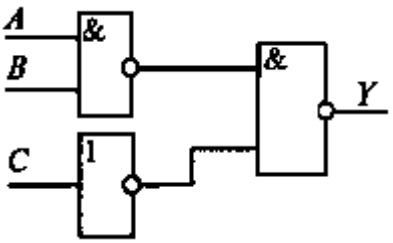


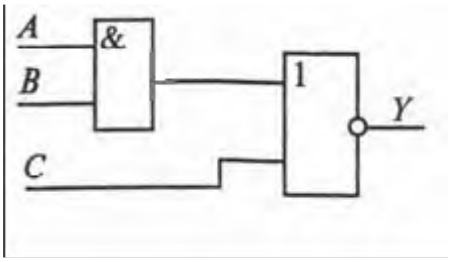


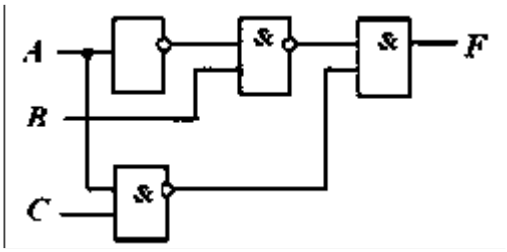


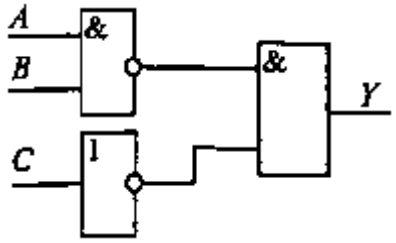


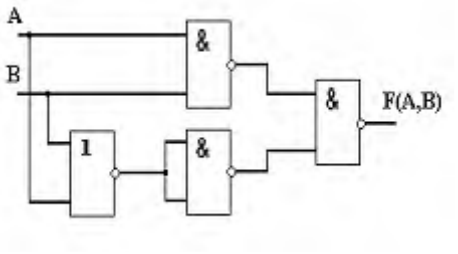


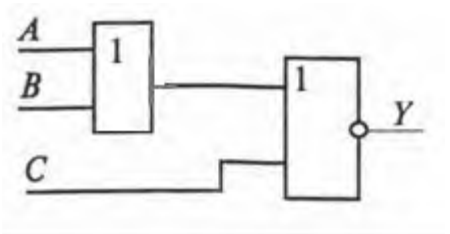


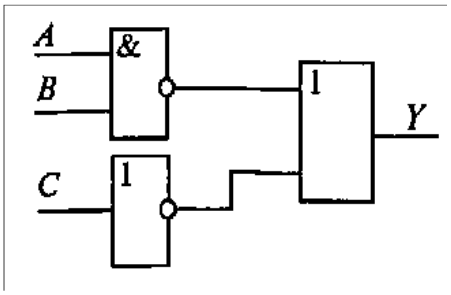












Выводы и предложения о проделанной работе:

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Список используемых источников
5. Выводы и предложения

Вопросы для самопроверки:

1.Что такое логическое высказывание?

2. Перечислите основные бинарные логические операции и связки.

3. Опишите операцию инверсии – НЕ.

4. Опишите операцию конъюнкции – И.

5. Опишите операцию дизъюнкции-ИЛИ.

6 Какие бывают логические выражения?

7 Что такое алгебра логики?

8 Понятие и обозначение инверсии.

9 Таблицы истинности инверсии

10 Понятие и обозначение конъюнкции.

11 Таблицы истинности конъюнкции.

12 Понятие и обозначение дизъюнкции.

13 Таблицы истинности дизъюнкции.

14 Способ изменения порядка действий в логических выражениях.

Практическое занятие №4 Кодирование данных различного вида. Их представление в компьютере.

Цель занятия:

Изучить способы представления текстовой, графической, звуковой информации и видеоинформации, научиться записывать информацию в различных кодировках.

Исходные данные:

Раздаточный материал

Содержание и порядок выполнения задания:

Изучите теоретическую часть

Выполните задания.

Теоретическая часть

Вся информация, которую обрабатывает компьютер, должна быть представлена двоичным кодом с помощью двух цифр 0 и 1. Эти два символа принято называть двоичными цифрами или битами. С помощью двух цифр 0 и 1 можно закодировать любое сообщение. Это явилось причиной того, что в компьютере обязательно должно быть организованно два важных процесса: кодирование и декодирование.

Кодирование – преобразование входной информации в форму, воспринимаемую компьютером, то есть двоичный код.

Декодирование – преобразование данных из двоичного кода в форму, понятную человеку.

С точки зрения технической реализации использование двоичной системы счисления для кодирования информации оказалось намного более простым, чем применение других способов. Действительно, удобно кодировать информацию в виде последовательности нулей и единиц, если представить эти значения как два возможных устойчивых состояния электронного элемента:

0 – отсутствие электрического сигнала;

1 – наличие электрического сигнала.

Эти состояния легко различать. Недостаток двоичного кодирования – длинные коды. Но в технике легче иметь дело с большим количеством простых элементов, чем с небольшим числом сложных.

Способы кодирования и декодирования информации в компьютере, в первую очередь, зависит от вида информации, а именно, что должно кодироваться: числа, текст, графические изображения или звук.

Аналоговый и дискретный способ кодирования

Человек способен воспринимать и хранить информацию в форме образов (зрительных, звуковых, осязательных, вкусовых и обонятельных). Зрительные образы могут быть сохранены в виде изображений (рисунков, фотографий и так далее), а звуковые - зафиксированы на пластинках, магнитных лентах, лазерных дисках и так далее.

Информация, в том числе графическая и звуковая, может быть представлена в аналоговой или дискретной форме. При аналоговом представлении физическая величина принимает бесконечное множество значений, причем ее значения изменяются непрерывно. При дискретном представлении физическая величина принимает конечное множество значений, причем ее величина изменяется скачкообразно.

Примером аналогового представления графической информации может служить, например, живописное полотно, цвет которого изменяется непрерывно, а дискретного – изображение, напечатанное с помощью струйного принтера и состоящее из отдельных точек разного цвета. Примером аналогового хранения звуковой информации является виниловая пластинка (звуковая дорожка изменяет

свою форму непрерывно), а дискретного – аудио компакт-диск (звуковая дорожка которого содержит участки с различной отражающей способностью).

Преобразование графической и звуковой информации из аналоговой формы в дискретную производится путем дискретизации, то есть разбиения непрерывного графического изображения и непрерывного (аналогового) звукового сигнала на отдельные элементы. В процессе дискретизации производится кодирование, то есть присвоение каждому элементу конкретного значения в форме кода.

Дискретизация – это преобразование непрерывных изображений и звука в набор дискретных значений в форме кодов.

Кодирование графических изображений

Создавать и хранить графические объекты в компьютере можно двумя способами – как растровое или как векторное изображение. Для каждого типа изображений используется свой способ кодирования.

Кодирование растровых изображений

Растровое изображение представляет собой совокупность точек (пикселей) разных цветов. Пиксель – минимальный участок изображения, цвет которого можно задать независимым образом.

В процессе кодирования изображения производится его пространственная дискретизация. Пространственную дискретизацию изображения можно сравнить с построением изображения из мозаики (большого количества маленьких разноцветных стекол). Изображение разбивается на отдельные маленькие фрагменты (точки), причем каждому фрагменту присваивается значение его цвета, то есть код цвета (красный, зеленый, синий и так далее).

Для черно-белого изображения информационный объем одной точки равен одному биту (либо черная, либо белая – либо 1, либо 0).

Для четырех цветного – 2 бита.

Для 8 цветов необходимо – 3 бита.

Для 16 цветов – 4 бита.

Для 256 цветов – 8 бит (1 байт).

Качество изображения зависит от количества точек (чем меньше размер точки и, соответственно, больше их количество, тем лучше качество) и количества

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

используемых цветов (чем больше цветов, тем качественнее кодируется изображение).

Для представления цвета в виде числового кода используются две обратных друг другу цветовые модели: RGB или CMYK. Модель RGB используется в телевизорах, мониторах, проекторах, сканерах, цифровых фотоаппаратах. Основные цвета в этой модели: красный (Red), зеленый (Green), синий (Blue). Цветовая модель CMYK используется в полиграфии при формировании изображений, предназначенных для печати на бумаге.

Цветные изображения могут иметь различную глубину цвета, которая задается количеством битов, используемых для кодирования цвета точки.

Если кодировать цвет одной точки изображения тремя битами (по одному биту на каждый цвет RGB), то мы получим все восемь различных цветов.

R	G	B	Цвет
1	1	1	Белый
1	1	0	Желтый
1	0	1	Пурпурный
1	0	0	Красный
0	1	1	Голубой
0	1	0	Зеленый
0	0	1	Синий
0	0	0	Черный

На практике же, для сохранения информации о цвете каждой точки цветного изображения в модели RGB обычно отводится 3 байта (то есть 24 бита) - по 1 байту (то есть по 8 бит) под значение цвета каждой составляющей. Таким образом, каждая RGB-составляющая может принимать значение в диапазоне от 0 до 255 (всего 28=256 значений), а каждая точка изображения, при такой системе кодирования может быть окрашена в один из 16 777 216 цветов. Такой набор цветов принято называть True Color (правдивые цвета), потому что человеческий глаз все равно не в состоянии различить большего разнообразия.

Для того чтобы на экране монитора формировалось изображение, информация о каждой точке (код цвета точки) должна храниться в видеопамяти компьютера. Рассчитаем необходимый объем видеопамяти для одного из графических режимов. В современных компьютерах разрешение экрана обычно составляет 1280x1024 точек. Т.е. всего 1280 * 1024 = 1310720 точек. При глубине цвета 32 бита на точку необходимый объем видеопамяти:

32 * 1310720 = 41943040 бит = 5242880 байт = 5120 Кб = 5 Мб.

Растровые изображения очень чувствительны к масштабированию (увеличению или уменьшению). При уменьшении растрового изображения несколько соседних точек преобразуются в одну, поэтому теряется различимость мелких деталей изображения. При увеличении изображения увеличивается размер каждой точки и появляется ступенчатый эффект, который можно увидеть невооруженным глазом.

Кодирование векторных изображений

Векторное изображение представляет собой совокупность графических примитивов (точка, отрезок, эллипс…). Каждый примитив описывается математическими формулами. Кодирование зависит от прикладной среды.

Достоинством векторной графики является то, что файлы, хранящие векторные графические изображения, имеют сравнительно небольшой объем.

Важно также, что векторные графические изображения могут быть увеличены или уменьшены без потери качества.

Графические форматы файлов

Форматы графических файлов определяют способ хранения информации в файле (растровый или векторный), а также форму хранения информации (используемый алгоритм сжатия). Наиболее популярные растровые форматы:

Bit MaP image (BMP) – универсальный формат растровых графических файлов, используется в операционной системе Windows. Этот формат поддерживается многими графическими редакторами, в том числе редактором Paint. Рекомендуется для хранения и обмена данными с другими приложениями.

Tagged Image File Format (TIFF) – формат растровых графических файлов, поддерживается всеми основными графическими редакторами и компьютерными платформами. Включает в себя алгоритм сжатия без потерь информации. Используется для обмена документами между различными программами. Рекомендуется для использования при работе с издательскими системами.

Graphics Interchange Format (GIF) – формат растровых графических файлов, поддерживается приложениями для различных операционных систем. Включает алгоритм сжатия без потерь информации, позволяющий уменьшить объем файла в несколько раз. Рекомендуется для хранения изображений, создаваемых

программным путем (диаграмм, графиков и так далее) и рисунков (типа аппликации) с ограниченным количеством цветов (до 256). Используется для размещения графических изображений на Web-страницах в Интернете.

Portable Network Graphic (PNG) – формат растровых графических файлов, аналогичный формату GIF. Рекомендуется для размещения графических изображений на Web-страницах в Интернете.

Joint Photographic Expert Group (JPEG) – формат растровых графических файлов, который реализует эффективный алгоритм сжатия (метод JPEG) для отсканированных фотографий и иллюстраций. Алгоритм сжатия позволяет уменьшить объем файла в десятки раз, однако приводит к необратимой потере части информации. Поддерживается приложениями для различных операционных систем. Используется для размещения графических изображений на Web-страницах в Интернете.

Двоичное кодирование звука

Использование компьютера для обработки звука началось позднее, нежели чисел, текстов и графики.

Звук – волна с непрерывно изменяющейся амплитудой и частотой. Чем больше амплитуда, тем он громче для человека, чем больше частота, тем выше тон.

Звуковые сигналы в окружающем нас мире необычайно разнообразны. Сложные непрерывные сигналы можно с достаточной точностью представлять в виде суммы некоторого числа простейших синусоидальных колебаний.

Причем каждое слагаемое, то есть каждая синусоида, может быть точно задана некоторым набором числовых параметров – амплитуды, фазы и частоты, которые можно рассматривать как код звука в некоторый момент времени.

В процессе кодирования звукового сигнала производится его временная дискретизация– непрерывная волна разбивается на отдельные маленькие временные участки и для каждого такого участка устанавливается определенная величина амплитуды.

Таким образом непрерывная зависимость амплитуды сигнала от времени заменяется на дискретную последовательность уровней громкости.

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

Каждому уровню громкости присваивается его код. Чем большее количество уровней громкости будет выделено в процессе кодирования, тем большее количество информации будет нести значение каждого уровня и тем более качественным будет звучание.

Качество двоичного кодирования звука определяется глубиной кодирования и частотой дискретизации.

Частота дискретизации – количество измерений уровня сигнала в единицу времени.

Количество уровней громкости определяет глубину кодирования. Современные звуковые карты обеспечивают 16-битную глубину кодирования звука. При этом количество уровней громкости равно N = 216 = 65536.

Представление видеоинформации

В последнее время компьютер все чаще используется для работы с видеоинформацией. Простейшей такой работой является просмотр кинофильмов и видеоклипов. Следует четко представлять, что обработка видеоинформации требует очень высокого быстродействия компьютерной системы.

Что представляет собой фильм с точки зрения информатики? Прежде всего, это сочетание звуковой и графической информации. Кроме того, для создания на экране эффекта движения используется дискретная по своей сути технология быстрой смены статических картинок. Исследования показали, что если за одну секунду сменяется более 10-12 кадров, то человеческий глаз воспринимает изменения на них как непрерывные.

Казалось бы, если проблемы кодирования статической графики и звука решены, то сохранить видеоизображение уже не составит труда. Но это только на первый взгляд, поскольку, как показывает разобранный выше пример, при использовании традиционных методов сохранения информации электронная версия фильма получится слишком большой. Достаточно очевидное усовершенствование состоит в том, чтобы первый кадр запомнить целиком (в литературе его принято называть ключевым), а в следующих сохранять лишь отличия от начального кадра (разностные кадры).

Существует множество различных форматов представления видеоданных.

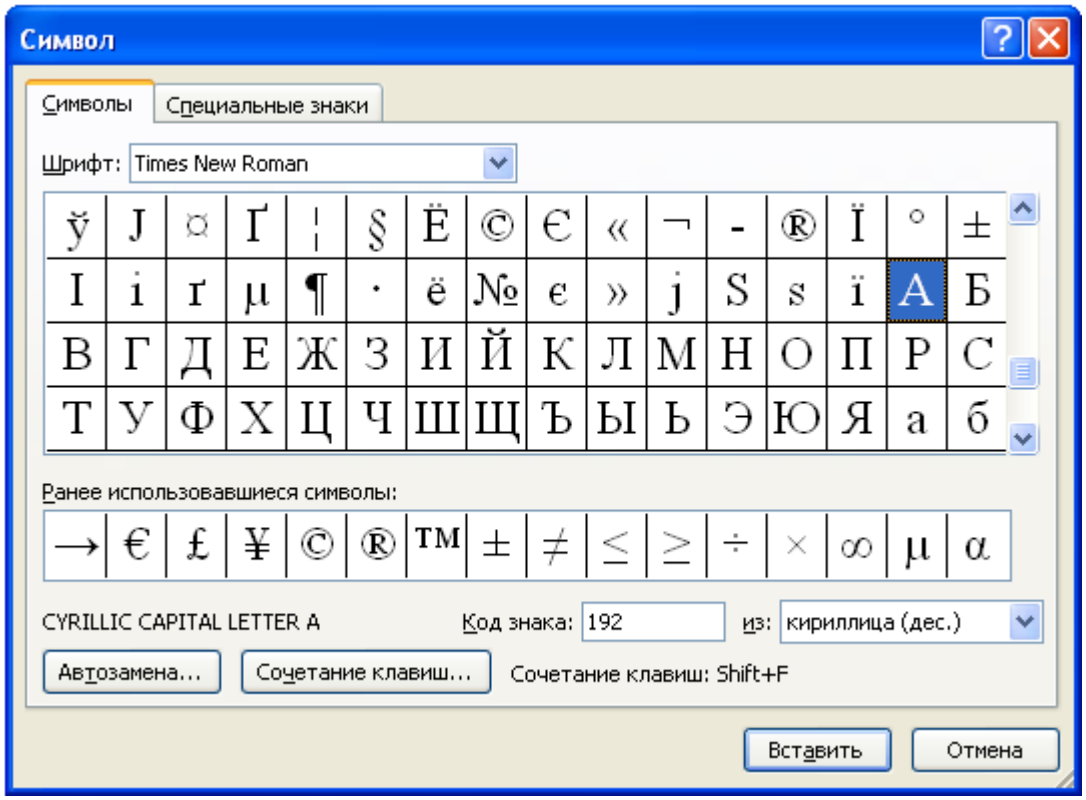
В среде Windows, например, уже более 10 лет (начиная с версии 3.1) применяется формат Video for Windows, базирующийся на универсальных файлах с расширением AVI (Audio Video Interleave – чередование аудио и видео).

Более универсальным является мультимедийный формат Quick Time, первоначально возникший на компьютерах Apple.

Задания

Задача 1. Используя таблицу символов, записать последовательность десятичных числовых кодов в кодировке Windows для своих ФИО, названия улицы, по которой проживаете. Таблица символов отображается в редакторе MS Word с помощью команды: вкладка Вставка>Символ>Другие символы.

В поле Шрифт выбираете Times New Roman, в поле из выбираете кириллица. Например, для буквы «А» (русской заглавной) код знака– 192.



Пример:

И	В	А	Н	О	В		А	Р	Т	Е	М		П	Е	Т	Р	О	В	И	
200	194	192	205	206	194		192	208	210	197	204		207	197	210	208	206	194	200	2

Задача 2.

1) Используя стандартную программу БЛОКНОТ, определить, какая фраза в кодировке Windows задана последовательностью числовых кодов и продолжить код. Запустить БЛОКНОТ. С помощью дополнительной цифровой клавиатуры при нажатой клавише ALT ввести код, отпустить клавишу ALT. В документе появиться соответствующий символ.

0255		0243	0247	0243	0241	0252		0226		0208	0232	0234		0239	0238	
0241	0239	0229	0246	0232	0224	0235	0252	0237	0238	0241	0242	0232				

2)* В кодировке Unicod запишите название своей специальности.

Задача 3. По вариантам

Вариант 1

3.1 Закодируйте в кодировке Unicode следующую фразу Азбука - к мудрости ступенька

Полученный код запишите в тетрадь

3.2 Фраза закодирована в кодировке Юникод (шестн.). Расшифруйте ее:

041D 0410 0020 0413 0420 0415 0411 041D 0415 0020 0412 041E 041B 041D 042B 0020 042D 0422 041E 0020 041A 041D 0418 0413 0410

Ответ запишите в тетрадь

Для определения кода символа: запустить текстовый редактор. Введите команду ВСТАВКА- СПЕЦИАЛЬНЫЕ СИМВОЛЫ. На экране появится диалоговая панель ВЫБОР СИМВОЛА. Для определения десятичного числового кода символа выбрать кодировку из списка, выбрать символ. Появиться код знака. Возможен и обратный вариант, только буквы в кодах – латинские.

3.3 Разведчиком была получена шифрованная радиограмма, переданная с использованием азбуки Морзе:

— • — • — • • — — • • • —

При передаче радиограммы было потеряно разбиение на буквы, но известно, что в радиограмме использовались только следующие буквы:

И	А	Н	Р	Т
••	• —	— •	• — •	—

Вариант 2

3.1 Закодируйте в кодировке Кириллица (шестн.) следующую фразу Время дороже золота

Полученный код запишите в тетрадь

3.2 Фраза закодирована в кодировке Юникод (шестн.). Расшифруйте.

041B 0415 041D 042C 0020 041B 0401 0416 0410 0020 041D 0410 0020 041F 0415 0427 0418 0020 0417 0410 041C 0401 0420 0417 041B 0410. Ответ запишите в тетрадь

Для определения кода символа: запустить текстовый редактор. Введите команду ВСТАВКА- СПЕЦИАЛЬНЫЕ СИМВОЛЫ. На экране появится диалоговая панель ВЫБОР СИМВОЛА. Для определения десятичного числового кода символа выбрать кодировку из списка, выбрать символ. Появиться код знака. Возможен и обратный вариант, только буквы в кодах – латинские.

3.3 Разведчиком была получена шифрованная радиограмма, переданная с использованием азбуки Морзе: •— —•• — •• —• — — —••• —

При передаче радиограммы было потеряно разбиение на буквы, но известно, что в радиограмме использовались только следующие буквы:

П	А	В	Р	Д
•— —•	• —	• — —	• — •	—••

Вариант 3

3.1 Закодируйте в кодировке Кириллица (шестн.) следующую фразу Век долог, да час дорог

Полученный код запишите в тетрадь

3.2 Фраза закодирована в кодировке Юникод (шестн.). Расшифруйте. 0412 0415 041A 0020 041F 0420 041E 0422 042F 041D 0415 0422 0421 042F 002C 0020 0412 0421 0415 0413 041E 0020 0414 041E 0421 0422 0410 041D 0415 0422 0421 042F. Ответ запишите в тетрадь

Для определения кода символа: запустить текстовый редактор. Введите команду ВСТАВКА- СПЕЦИАЛЬНЫЕ СИМВОЛЫ. На экране появится диалоговая панель ВЫБОР СИМВОЛА. Для определения десятичного числового кода символа выбрать

кодировку из списка, выбрать символ. Появиться код знака. Возможен и обратный вариант, только буквы в кодах – латинские.

3.3 Разведчиком была получена шифрованная радиограмма, переданная с использованием азбуки Морзе:

••— •••—•— — —• — •— —• — —•—••••—•—

При передаче радиограммы было потеряно разбиение на буквы, но известно, что в радиограмме использовались только следующие буквы:

И	А	Н	Р	Ц	Ф	О	М	Я
••	• —	— •	• — •	—•—•	••—•	— — —	— —	•—•—

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список используемых источников
5. Выводы и предложения

Вопросы для самопроверки:

1. Чем отличается непрерывный сигнал от дискретного?
2. Что такое частота дискретизации и на что она влияет?
3. Опишите как происходит кодирование звука?
4. Какие звуковые форматы вы знаете?
5. Какие этапы кодирования видеоинформации вам известны?
6. Какие форматы видео файлов вы знаете?

Практическое занятие №5 Системы счисления. Переводы чисел из одной системы счисления в другую. Арифметические действия в различных системах счисления.

Цель занятия:

1.Изучение систем счисления и получение практических навыков перевода чисел из одной системы счисления в другую.

2.Ознакомиться с арифметические операции над числами, записанными в двоичной, восьмеричной и шестнадцатеричной системе счисления

Исходные данные:

Таблица, правила перевода систем счисления.

Содержание и порядок выполнения задания:

Изучите теоретическую часть

Выполните задания.

Теоретическая часть 1

Представление числовой информации с помощью систем счисления

Для записи информации о количестве объектов используются числа. Числа записываются с использованием особых знаковых систем, которые называются системами счисления. Алфавит систем счисления состоит из символов, которые называются цифрами. Например, в десятичной системе счисления числа записываются с помощью десяти всем хорошо известных цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Система счисления - это знаковая система, в которой числа записываются по определенным правилам с помощью символов некоторого алфавита, называемых цифрами.

Все системы счисления делятся на две большие группы: *позиционные* и *непозиционные* системы счисления. В позиционных системах счисления значение цифры зависит от ее положения в числе, а в непозиционных - не зависит.

Римская непозиционная система счисления. Самой распространенной из непозиционных систем счисления является римская. В качестве цифр в ней используются **X={I (1), V (5), X (10), L (50), C (100), D (500), M (1000)}**, где в скобках указаны веса символов (не зависящие от позиции символа)

Примеры римских чисел (в скобках – обычные десятичные эквиваленты):

III (3), IV (4), V (5), VI (6), IX (9), XI (11), DCL (650).

Значение цифры не зависит от ее положения в числе. Например, в числе XXX (30) цифра X встречается трижды и в каждом случае обозначает одну и ту же величину - число 10, три числа по 10 в сумме дают 30.

Позиционные системы счисления. Первая позиционная система счисления была придумана еще в Древнем Вавилоне, причем вавилонская нумерация была шестидесятеричной, то есть в ней использовалось шестьдесят цифр! Интересно, что до сих пор при измерении времени мы используем основание, равное 60 (в 1 минуте содержится 60 секунд, а в 1 часе - 60 минут).

В XIX веке довольно широкое распространение получила двенадцатеричная система счисления. До сих пор мы часто употребляем дюжину (число 12): в сутках две дюжины часов, круг содержит тридцать дюжин градусов и так далее.

*В **позиционных системах счисления** количественное значение цифры зависит от ее позиции в числе.*

Наиболее распространенными в настоящее время позиционными системами счисления являются десятичная, двоичная, восьмеричная и шестнадцатеричная. Каждая позиционная система имеет определенный *алфавит цифр* и *основание*.

*В **позиционных системах счисления** основание системы равно количеству цифр (знаков в ее алфавите) и определяет, во сколько раз различаются значения одинаковых цифр, стоящих в соседних позициях числа.*

Десятичная система счисления имеет алфавит цифр, который состоит из десяти всем известных, так называемых арабских, цифр, и основание, равное 10, двоичная - две цифры и основание 2, восьмеричная - восемь цифр и основание 8, шестнадцатеричная - шестнадцать цифр (в качестве цифр используются и буквы латинского алфавита) и основание 16 (табл.).

Таблица – Позиционные системы счисления

Основание	**Система счисления**	**Знаки**
2	*Двоичная*	над алфавитом **X = {0,1}**;
3	*Троичная*	над **X = {0,1,2}**;
8	*Восьмеричная*	над **X = {0, 1, 2, 3, 4, 5, 6, 7}**;
10	Десятеричная	над **X = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};**

16	*Шестнадцатеричная*	над **X = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}**, где символы **A, B, C, D, E, F** имеют, соответственно, десятичные

В позиционных системах чем больше основание системы, тем меньшее количество разрядов (то есть записываемых цифр) требуется при записи числа.

Арабская система счисления, которым мы пользуемся в повседневной жизни, является позиционной. В позиционных системах счисления позиция числа однозначно определяет величину числа. Рассмотрим это на примере числа 6372 в десятичной системе счисления. Пронумеруем это число, справа, налево начиная с нуля:

Число	6	3	7	2
Позиция	3	2	1	0

Тогда число 6372 можно представить в развернутом виде:

$6372=6000+300+70+2 =6\cdot10^{3}+3\cdot10^{2}+7\cdot10^{1}+2\cdot10^{0}$.

Число 10 определяет систему счисления. В качестве степеней взяты значения позиции данного числа.

Рассмотрим вещественное десятичное число 1287,923. Пронумеруем его, начиная с нуля позиции числа от десятичной точки влево и вправо:

Число	1	2	8	7	.	9	2	3
Позиция	3	2	1	0	.	-1	-2	-3

Тогда число 1287.923 можно представить в виде:

$1287.923=1000+200+80+7+0.9+0.02+0.003=1\cdot10^{3}+2\cdot10^{2}+8\cdot10^{1}+7\cdot10^{0}+9\cdot10^{-1}+2\cdot10^{-2}+ +3\cdot10^{-3}$.

В вычислительных машинах используется двоичная система счисления, её основание - число 2. Для записи чисел в этой системе используют только две цифры - 0 и 1.

Выбор двоичной системы для применения в вычислительной технике объясняется тем, что электронные элементы - триггеры, из которых состоят микросхемы ЭВМ, могут находиться только в двух рабочих состояниях.

Двоичная система удобна для компьютера, но неудобна для человека: числа получаются длинными и их трудно записывать и запоминать. Конечно, можно перевести число в десятичную систему и записывать в таком виде, а потом, когда понадобится перевести обратно, но все эти переводы трудоёмки. Поэтому применяются системы счисления, родственные двоичной - восьмеричная и шестнадцатеричная. Для записи чисел в этих системах требуется соответственно 8 и 16 цифр. В шестнадцатеричной системе счисления первые десять цифр общие, а дальше используют заглавные латинские буквы. Шестнадцатеричная цифра A соответствует десятеричному числу 10, шестнадцатеричная B – десятичному числу 11 и т. д. Использование этих систем объясняется тем, что переход к записи числа в любой из этих систем от его двоичной записи очень прост. В таблице 1 приведена таблица соответствия чисел, записанных в разных системах.

Таблица 1

Двоичные числа	Восьмеричные числа	Десятичные числа	Шестнадцатеричные числа
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

Перевод чисел из произвольной системы счисления в десятичную систему счисления и обратно

Перевод чисел из одной системы счисления в другую составляет важную часть машинной арифметики. В вычислительной технике применяют позиционные

системы счисления с недесятичным основанием: двоичную, восьмеричную, шестнадцатеричную. Для обозначения используемой системы счисления число снабжают верхним или нижним индексом, в котором записывают основание системы счисления.

16-ричная и 8-ричная система счисления используются при составлении программ на языке машинных кодов для более короткой и удобной записи двоичных кодов – команд, данных, адресов и операндов. Задача перевода из одной системы счисления в другую часто встречается при программировании.

Если основание используемой системы счисления больше десяти, то для цифр вводят буквенное обозначение.

В шестнадцатеричной системе счисления основа - это цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 с соответствующими обозначениями 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Примеры чисел: 17D, CA, F12A.

Двоичная система счисления - это система, в которой для записи чисел используются две цифры 0 и 1. Основанием двоичной системы счисления является число 2. Двоичный или бинарный код числа - запись этого числа в двоичной системе счисления. Например,

$0=0_2$

$1=1_2$

$2=10_2$

$3=11_2$

$7=111_2$

$120=1111000_2$.

Для перевода целого числа из десятичной системы счисления в s-ичную необходимо последовательно делить это число и получаемые частные на s (по правилам системы счисления с основанием s) до тех пор, пока частное не станет равным нулю. Старшей цифрой в записи числа с основанием s служит последний остаток, а следующие за ней цифры образуют остатки от предшествующих делений, выписываемые в последовательности, обратной их получению.

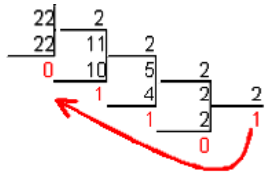
Рассмотрим основные правила перевода:

✓Для перевода десятичного числа в двоичную систему его необходимо последовательно делить на 2 до тех пор, пока не останется остаток, меньший или равный 1. Число в двоичной системе записывается как последовательность последнего результата деления и остатков от деления в обратном порядке.

Пример 1. Число 22_{10} перевести в двоичную систему счисления.

Решение:

I способ:



$22_{10}=10110_2$

II способ:

$22_{10}=11\cdot2(0)$

$11 = 5\cdot2(1)$

$5 = 2\cdot2(1)$

$2=1\cdot2(0)$

$22_{10} = 10110_2$

✓Для перевода двоичного числа в десятичное необходимо его записать в виде многочлена (то есть в развернутом виде), состоящего из произведений цифр числа и соответствующей степени числа 2, и вычислить по правилам десятичной арифметики:

$X_2=a_n\cdot2^{n-1} + a_{n-1}\cdot2^{n-2} + a_{n-2}\cdot2^{n-3} + a_n\cdot2^{n-1}+ \dots + a_2\cdot2^1 + +a_1\cdot2^0$

Пример2. Число 11101000_2перевести в десятичную систему счисления.

Решение:

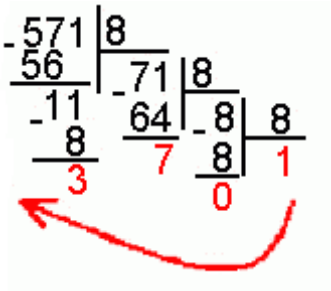
$11101000_2=1\cdot2^7+1\cdot2^6+1\cdot2^5+0\cdot2^4+1\cdot2^3+0\cdot2^2+0\cdot2^1+0\cdot2^0=232_{10}$

✓Для перевода десятичного числа в восьмеричную систему его необходимо последовательно делить на 8 до тех пор, пока не останется остаток, меньший 8. Число в восьмеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример3. Число 57110 перевести в восьмеричную систему счисления.

Решение:

I способ:



$571_{10} = 1073_8$

II способ:

57110 = 71·8(3)

71 = 8 ·8(7)

8 = 1·8(0)

$571_{10} = 1073_8$

✓Для перевода восьмеричного числа в десятичное число необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 8, и вычислить по правилам десятичной арифметики:

$X_8=a_n \cdot 8^{n-1} + a_{n-1} \cdot 8^{n-2} + a_{n-2} \cdot 8^{n-3} + a_n \cdot 8^{n-1} + \ldots + a_2 \cdot 8^1 + a_1 \cdot 8^0$

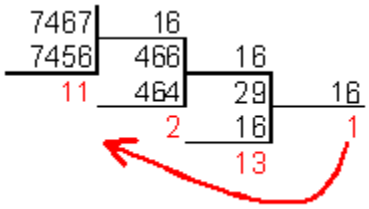
Пример 4. Число 5013_8 перевести в десятичную систему счисления.

Решение: $5013_8 = 5 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 = 2571_{10}$

Для перевода десятичного числа в шестнадцатеричную систему его необходимо последовательно делить на 16 до тех пор, пока не останется остаток, меньший 16. Число в шестнадцатеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример 5. Число 7467_{10} перевести в шестнадцатеричную систему счисления.

Решение: I способ:



$7467_{10}=1D2B_{16}$

II способ:

$7467_{10} = 466 \cdot 16(11)$

$466 = 29 \cdot 16(1)$

$29=13 \cdot 16(0)$

$7467_{10}=1D2B_{16}$

Для перевода шестнадцатеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 16, и вычислить по правилам десятичной арифметики:

$$X_{16}=a_n \cdot 16^{n-1} + a_{n-1} \cdot 16^{n-2} + a_{n-2} \cdot 16^{n-3} + a_n \cdot 16^{n-1} + \ldots + a_2 \cdot 16^{1} + + a_1 \cdot 16^{0}$$

Пример 6. Число $FDA1_{16}$ перевести в десятичную систему счисления.

Решение:

$$FDA1_{16}=15 \cdot 16^{3}+13 \cdot 16^{2}+10 \cdot 16^{1}+1 \cdot 16^{0}=64929_{10}$$

✓Чтобы перевести число из двоичной системы в восьмеричную, его нужно разбить на триады (тройки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую триаду нулями, и каждую триаду заменить соответствующей восьмеричной цифрой (таблица 1).

Пример 7. Число1001011_2 перевести в восьмеричную систему счисления.

Решение: делим число, начиная с конца на триады и заменяем их восьмеричными числами используя таблицу 1

$001\ 001\ 011_2 = 113_8$

✓Чтобы перевести число из двоичной системы в шестнадцатеричную, его нужно разбить на тетрады (четверки цифр), начиная с младшего разряда, в

случае необходимости дополнив старшую тетраду нулями, и каждую тетраду заменить соответствующей шестнадцатеричной цифрой (таблица 1).

Пример 8. Число1011100011_2 перевести в шестнадцатеричную систему счисления.

Решение: $0010\ 1110\ 0011_2=2E3_{16}$

Для перевода восьмеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной триадой (таблица 1).

Пример 9. Число531_8 перевести в двоичную систему счисления.

Решение: $527_8=101\ 010\ 111_2$

Для перевода шестнадцатеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной тетрадой (таблица 1).

Пример 10. Число $EF8_{16}$ перевести в двоичную систему счисления

Решение: $EE8_{16}=111011101000_2$

При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно, необходим промежуточный перевод чисел в десятичную или двоичную систему.

Пример11. Число FEA_{16} перевести в восьмеричную систему счисления.

Решение:

I способ:

$FEA_{16}=15\cdot16^2+14\cdot16^1+10\cdot16^0=3840+224+10=4074_{10}$

4074:8=509(2)

509:8=63(5)

63:8=7(7)

$FEA_{16}=4074_{10}=7752_8$

II способ:

$FEA_{16}=1111\ 1110\ 1010_2$ (по таблице1)

$111\ 111\ 101\ 010_2=7752_8$

Пример12. Число 635_8 перевести в шестнадцатеричную систему счисления.

Решение:

I способ:

$635_8=6\cdot8^2+3\cdot8^1+5\cdot8^0=384+24+5=413_{10}$

413:16=25(13)

25:16=1(9)

$635_8=413_{10}=19D_{16}$

II способ:

$635_8=110\ 011101_2$ (по таблице1)

$0001\ 1001\ 1101_2=19D_{16}$

Задания:

Варианты заданий.

Вариант 1.

1) Преобразуйте следующие двоичные числа в десятичные:
   1101, 110110, 10111, 10010, 1001.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   1, 510, 57, 15, 653.
3) Преобразуйте следующие десятичные числа в двоичные:
   256, 13, 72, 11, 101.
4) Преобразуйте следующие десятичные числа в восьмеричные:
   39, 7, 162, 41, 418.
5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
   764, 66, 10, 763, 14.
6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
   69, 8, 2E9, 3FC, 55.

Вариант 2

1) Преобразуйте следующие двоичные числа в десятичные:
   11100, 1110, 100111, 11110, 100101.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   576, 5, 42, 3, 71.
3) Преобразуйте следующие десятичные числа в двоичные:
   296, 13, 44, 62, 2.
4) Преобразуйте следующие десятичные числа в восьмеричные:

2, 34, 305, 22, 7.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
912, 11, 75, 15, 64.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
61, 268, D, 23B, 67.

Вариант 3.

1) Преобразуйте следующие двоичные числа в десятичные:
1000, 111001, 10011, 1101, 101110.

2) Преобразуйте следующие восьмеричные числа в десятичные:
2, 20, 244, 7, 24.

3) Преобразуйте следующие десятичные числа в двоичные:
52, 21, 342, 6, 112.

4) Преобразуйте следующие десятичные числа в восьмеричные:
6, 43, 389, 7, 140.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
8, 112, 798, 643, 11.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
248, 41, E, 45, 23C.

Вариант 4.

1) Преобразуйте следующие двоичные числа в десятичные:
1111, 110001, 11000, 1000, 11011.

2) Преобразуйте следующие восьмеричные числа в десятичные:
54, 714, 7, 6, 777.

3) Преобразуйте следующие десятичные числа в двоичные:
40, 12, 199, 91, 29.

4) Преобразуйте следующие десятичные числа в восьмеричные:
288, 3, 19, 336, 2.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
117, 11, 972, 15, 541.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
75, 360, 8, 7A, C.

Вариант 5.

1) Преобразуйте следующие двоичные числа в десятичные:
   1111, 10110, 100000, 111001, 10000.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   322, 1, 43, 402, 37.
3) Преобразуйте следующие десятичные числа в двоичные:
   16, 73, 255, 19, 90.
4) Преобразуйте следующие десятичные числа в восьмеричные:
   2, 30, 241, 62, 3.
5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
   949, 89, 9, 120, 12.
6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
   F, 4F, 348, 219, 6C.

Вариант 6.

1) Преобразуйте следующие двоичные числа в десятичные:
   1111, 100111, 10100, 1110, 11111.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   4, 676, 11, 70, 7.
3) Преобразуйте следующие десятичные числа в двоичные:
   97, 7, 479, 155, 41.
4) Преобразуйте следующие десятичные числа в восьмеричные:
   294, 1, 20, 197, 7.
5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
   815, 14, 108, 895, 13.
6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
   374, 72, 9, 5F, 2B6.

Вариант 7.

1) Преобразуйте следующие двоичные числа в десятичные:
   1011, 10000, 101101, 1100, 100101.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   3, 16, 215, 177, 6.
3) Преобразуйте следующие десятичные числа в двоичные:

172, 94, 26, 80, 339.

4) Преобразуйте следующие десятичные числа в восьмеричные:
484, 9, 3, 2, 110.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
639, 66, 10, 11, 77.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
4F, 347, 8, 6E, C.

Вариант 8.

1) Преобразуйте следующие двоичные числа в десятичные:
110000, 1000, 10110, 11000, 111110.

2) Преобразуйте следующие восьмеричные числа в десятичные:
40, 536, 2, 57, 114.

3) Преобразуйте следующие десятичные числа в двоичные:
121, 63, 22, 78, 154.

4) Преобразуйте следующие десятичные числа в восьмеричные:
7, 10, 75, 6, 336.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
76, 14, 898, 646, 70.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
9, 3FC, 7E, 77, 3FD.

Вариант 9.

1) Преобразуйте следующие двоичные числа в десятичные:
111010, 1000, 11011, 10100, 101010.

2) Преобразуйте следующие восьмеричные числа в десятичные:
7, 552, 45, 314, 6.

3) Преобразуйте следующие десятичные числа в двоичные:
97, 149, 19, 5, 90.

4) Преобразуйте следующие десятичные числа в восьмеричные:
4, 437, 13, 7, 47.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
114, 575, 12, 8, 922.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:

B, 67, 222, 234, C.

Вариант 10

1) Преобразуйте следующие двоичные числа в десятичные:
   10011, 1001, 100010, 10110, 110000.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   732, 3, 17, 7, 641.
3) Преобразуйте следующие десятичные числа в двоичные:
   27, 341, 93, 40, 5.
4) Преобразуйте следующие десятичные числа в восьмеричные:
   31, 1, 403, 117, 29.
5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
   620, 10, 100, 66, 760.
6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
   73, 233, C, F, 78.

Вариант 11.

1) Преобразуйте следующие двоичные числа в десятичные:
   111100, 1110, 11001, 101100, 1111.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   7, 21, 521, 730, 67.
3) Преобразуйте следующие десятичные числа в двоичные:
   8, 55, 215, 88, 7.
4) Преобразуйте следующие десятичные числа в восьмеричные:
   211, 5, 24, 6, 16.
5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
   874, 13, 68, 9, 725.
6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
   5A, 8, 244, A, 34F.

Вариант 12.

1) Преобразуйте следующие двоичные числа в десятичные:
   111000, 1000, 10111, 101101, 10101.
2) Преобразуйте следующие восьмеричные числа в десятичные:
   7, 175, 51, 3, 767.

3) Преобразуйте следующие десятичные числа в двоичные:
5, 463, 99, 76, 110.

4) Преобразуйте следующие десятичные числа в восьмеричные:
18, 306, 6, 43, 3.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
111, 584, 9, 671, 117.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
257, 7A, 9, 2FB, 73.

Вариант 13.

1) Преобразуйте следующие двоичные числа в десятичные:
11100, 101100, 1101, 100100, 10011.

2) Преобразуйте следующие восьмеричные числа в десятичные:
230, 16, 5, 607, 27.

3) Преобразуйте следующие десятичные числа в двоичные:
7, 71, 278, 172, 32.

4) Преобразуйте следующие десятичные числа в восьмеричные:
57, 1, 216, 226, 34.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
794, 126, 11, 591, 15.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
D, 350, 4C, F, 389.

Вариант 14.

1) Преобразуйте следующие двоичные числа в десятичные:
101101, 10010, 1000, 110001, 11001.

2) Преобразуйте следующие восьмеричные числа в десятичные:
660, 4, 40, 552, 2.

3) Преобразуйте следующие десятичные числа в двоичные:
128, 4, 45, 5, 297.

4) Преобразуйте следующие десятичные числа в восьмеричные:
104, 7, 61, 2, 223.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
117, 14, 728, 10, 563.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
77, 227, C, F, 204.

Вариант 15.

1) Преобразуйте следующие двоичные числа в десятичные:
110110, 10010, 1101, 1000, 11010.
2) Преобразуйте следующие восьмеричные числа в десятичные:
4, 17, 637, 1, 663.
3) Преобразуйте следующие десятичные числа в двоичные:
32, 373, 64, 67, 154.
4) Преобразуйте следующие десятичные числа в восьмеричные:
40, 80, 1, 61, 3.
5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
12, 733, 81, 603, 10.
6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
7F, 343, 9, 4E, 8.

Вариант 16.

1) Преобразуйте следующие двоичные числа в десятичные:
1111, 10110, 101100, 1101, 101111.
2) Преобразуйте следующие восьмеричные числа в десятичные:
577, 14, 2, 631, 11.
3) Преобразуйте следующие десятичные числа в двоичные:
301, 18, 97, 63, 16.
4) Преобразуйте следующие десятичные числа в восьмеричные:
4, 444, 22, 294, 7.
5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
98, 14, 972, 12, 83.
6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
E, 2EC, 70, 43, 20D.

Вариант 17.

1) Преобразуйте следующие двоичные числа в десятичные:
1001, 10001, 100110, 10110, 1011.

2) Преобразуйте следующие восьмеричные числа в десятичные:
20, 445, 6, 5, 607.

3) Преобразуйте следующие десятичные числа в двоичные:
468, 25, 53, 167, 18.

4) Преобразуйте следующие десятичные числа в восьмеричные:
196, 1, 40, 486, 5.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
13, 642, 86, 902, 8.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
72, 8, 35B, B, 355.

Вариант 18.

1) Преобразуйте следующие двоичные числа в десятичные:
1110, 111011, 11101, 1100, 100000.

2) Преобразуйте следующие восьмеричные числа в десятичные:
6, 71, 216, 53, 463.

3) Преобразуйте следующие десятичные числа в двоичные:
8, 392, 76, 25, 41.

4) Преобразуйте следующие десятичные числа в восьмеричные:
37, 7, 433, 301, 4.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
12, 991, 124, 11, 67.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
265, 6B, E, 71, C.

Вариант 19.

1) Преобразуйте следующие двоичные числа в десятичные:
10111, 1010, 111010, 1110, 110111.

2) Преобразуйте следующие восьмеричные числа в десятичные:
1, 151, 43, 355, 16.

3) Преобразуйте следующие десятичные числа в двоичные:
13, 91, 466, 75, 4.

4) Преобразуйте следующие десятичные числа в восьмеричные:
5, 497, 14, 45, 128.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
925, 8, 68, 109, 854.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
313, 7A, F, 337, A.

Вариант 20.

1) Преобразуйте следующие двоичные числа в десятичные:
100010, 11110, 1011, 1101, 10110.

2) Преобразуйте следующие восьмеричные числа в десятичные:
4, 204, 20, 1, 17.

3) Преобразуйте следующие десятичные числа в двоичные:
66, 207, 3, 90, 279.

4) Преобразуйте следующие десятичные числа в восьмеричные:
50, 1, 114, 277, 13.

5) Преобразуйте следующие десятичные числа в шестнадцатеричные:
11, 528, 126, 89, 9.

6) Преобразуйте следующие шестнадцатеричные числа в десятичные:
A, 29F, 59, 268, 8.

Изучите теоретическую часть

Выполните задания.

Теоретическая часть 2:

Над числами, записанными в любой системе счисления, можно производить различные арифметические операции. Многоразрядные числа складываются, вычитаются, умножаются и делятся по тем же правилам, что и в десятичной системе счисления.

Так, для сложения двоичных чисел необходимо использовать следующие правила:

0 + 0 = 0; 1 + 0 = 1;

0 + 1 = 1; 1 + 1 = 10.

В последнем случае, при сложении двух единиц, происходит переполнение младшего разряда, и единица переносится в старший разряд. Переполнение

возникает в случае, если сумма равна основанию системы счисления (в данном случае это число 2) или больше его (для двоичной системы счисления это не актуально).

Пример сложения двоичных чисел:

```
  10110      1001       1111
+   101    + 1010     +    1
-------    ------     ------
  11011      10011      10000
```

Вычитание одноразрядных двоичных чисел выполняется по следующим правилам:

0 - 0 = 0; 0 - 1 = 1 (заем из старшего разряда);

1 - 0 = 1; 1 - 1 = 0.

*Пример*вычитания двоичных чисел:

```
  1011     (0)
-  111     10110
------     01100
   100     -----
           01010
```

Умножение одноразрядных двоичных чисел выполняется по следующим правилам:

0 · 0 = 0; 1 · 0 = 0;
0 · 1 = 0; 1 · 1 = 1.

Пример умножения двоичных чисел:

```
   1011          10101
  x 101         x  111
-------       -------
   1011          10101
+ 1011        + 10101
-------       + 10101
1011          -------
-------       1001001 1
110111
```

Деление в любой позиционной системе счисления производится по тем же правилам, как и деление углом в десятичной системе. В двоичной системе деление выполняется особенно просто, ведь очередная цифра частного может быть только нулем или единицей.

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел

в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо заимствовать из соответствующих таблиц умножения и сложения.

Таблица сложения в восьмеричной системе счисления:

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Таблица умножения в восьмеричной системе счисления:

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Таблица сложения в 16-ричной системе счисления:

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Таблица умножения в 16-ричной системе счисления:

×	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

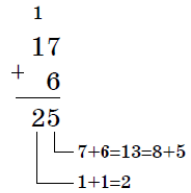
При выполнении действий сложения и вычитания в 8-ной системе счисления необходимо помнить (если не пользоваться таблицей):

- в записи результатов сложения и вычитания могут быть использованы только цифры восьмеричного алфавита;

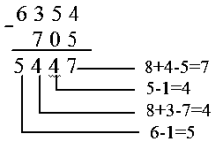
- основание восьмеричной системы счисления равен 8, т.е. переполнение наступает, когда результат сложения больше или равен 8. В этом случае для записи результата надо вычесть 8, записать остаток, а к старшему разряду прибавить единицу переполнения;

- если при вычитании приходится занимать единицу в старшем разряде, эта единица переносится в младший разряд в виде 8 единиц.

Пример сложения и вычитания восьмеричных чисел:

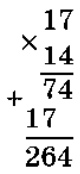


1) $17_8+6_8=25_8$



2) $6354_8-705_8=5447_8$

Пример умножения восьмеричных чисел:



$7 \cdot 4 = 28 = 8 \cdot 3 + \mathbf{4}$

$4 \cdot 1 + 3 = \mathbf{7}$

$17_8 \cdot 14_8 = 264_8$

✓При выполнении действий сложения и вычитания в 16-ной системе счисления необходимо помнить:

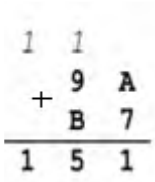
- в записи результатов сложения и вычитания могут быть использованы только цифры шестнадцатеричного алфавита (0-9, A-F)

- основание шестнадцатеричной системы счисления равно 16, т.е. переполнение наступает, когда результат сложения больше или равен 16. В этом случае для записи результата надо вычесть 16, записать остаток, а к старшему разряду прибавить единицу переполнения;

- если при вычитании приходится занимать единицу в старшем разряде, эта единица переносится в младший разряд в виде 16 единиц.

Пример сложения и вычитания 16-ных чисел:

1)



A+7 =10+7=17=17-16 =1→1

9+B +1=10+11=21=21- 16=5→1

$9A_{16} + B7_{16} = 151_{16}$

2)

$$\begin{array}{r} BC4_{16} \\ -\ AF_{16} \\ \hline B15_{16} \end{array}$$

Из 4 нельзя вычесть F, значит, из левого разряда мы займем 16. Теперь F надо вычитать из 20. В результате - 5, записываем его под разрядом единиц. Цифра С уменьшилась на 1,теперь это В. Значит надо А вычесть из В, это будет 1.

Записываем этот результат в разряде десятков. Из сотен в этот раз мы ничего не занимали и в вычитаемом только 2 цифры — сотен нет, то есть сносим В из уменьшаемого в результат.

Итак: $BC4_{16} - AF_{16} = B15_{16}$.

Пример умножения 16-ных чисел:

$$\begin{array}{r} 7 \\ \cdot\ 5\ D \\ \hline 5\ B \\ +\ 2\ 3\ \ \\ \hline 2\ 8\ B \end{array}$$

$7 \cdot D = 7 \cdot 13 = 91 = 16 \cdot 5 + 11 = 16 \cdot \mathbf{5} + \mathbf{B}$
$7 \cdot 5 = 35 = 16 \cdot \mathbf{2} + \mathbf{3}$

✓Операция деления выполняется по алгоритму, подобному алгоритму выполнения операции деления в десятичной системе счисления. Следует только грамотно пользоваться теми цифрами, которые входят в алфавит используемой системы счисления.

Наиболее просто деление столбиком осуществить в двоичной системе счисления, так как в двоичной системе счисления придется лишь сравнить два числа и вычитать из большего меньшее.

$$\begin{array}{l|l} 1000011_2 & 101_2 \\ \underline{-\ 101_2\ \ \ \ \ } & \underline{111_2} \\ \ \ 111_2 & \\ \ \ \underline{-\ 101_2} & \\ \ \ \ \ 101_2 & \\ \ \ \ \underline{-\ 101_2} & \\ \ \ \ \ \ 101_2 & \\ \ \ \ \ \underline{-\ 101_2} & \\ \ \ \ \ \ \ 0 & \end{array}$$

При делении мы не всегда получим целое число. Разделим число 5 на число 2, предварительно переведя их в двоичную систему счисления:

$$\begin{array}{l} 101_2 \;\big|\; 10_2 \\ \underline{10_2} \quad\; 10,1_2 \\ \;\;010_2 \\ \;\underline{-\;10_2} \\ \;\;\;0_2 \end{array}$$

Деление в восьмеричной системе счисления:

$$\begin{array}{l} 330 \;\big|\; 22 \\ \underline{22} \quad 14_8 \\ 110 \\ \underline{110} \\ 0 \end{array}$$

Процесс деления в шестнадцатеричной системе счисления аналогичен.

✓При выполнении любых арифметических операций над числами, представленными в разных системах счисления, следует предварительно перевести их в одну и ту же систему.

Пример 1. Выполнить арифметические операции с числами в различных системах счисления:

$101110_2 + 1263_8 \rightarrow x_{10}$

$101110_2 = 1\cdot2^5 + 0\cdot2^4 + 1\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 0\cdot2^0 = 32 + 0 + 8 + 4 + 2 + 0 = 46_{10}$

$1263_8 = 1\cdot8^3 + 2\cdot8^2 + 6\cdot8^1 + 3\cdot8^0 = 512 + 128 + 48 + 3 = 691_{10}$

$46_{10} + 691_{10} = 737_{10}$

Задание№1

Расположите следующие числа в порядке возрастания:

а) 74_8, 110010_2, 70_{10}, 38_{16};

б) $6E_{16}$, 142_8, 1101001_2, 100_{10};

в) 777_8, 101111111_2, $2FF_{16}$, 500_{10};

г) 100_{10}, 1100000_2, 60_{16}, 141_8.

Задание№2

Выполнить арифметические операции в 2-й СС:

1) $1110_2 + 1001_2 = 10111_2$

2) $1110_2 – 1001_2 = 101_2$

3) $1110_2 * 1001_2 = 1111110_2$

4) $1110_2 / 11_2 = 100_2$

Задание№3

Выполнить арифметические операции в 8-й СС:

5) $67_8 + 23_8 = 112_8$

6) $67_8 – 23_8 = 44_8$

7) $67_8 * 23_8 = 2025_8$

8) $74_8 / 24_8 = 3_8$

Задание№4

Выполнить арифметические операции в 16-й СС:

9) $AF_{16} + 97_{16} = 146_{16}$

10) $AF_{16}– 97_{16} = 18_{16}$

11) $AF_{16} * 97_{16} = 6739_{16}$

12) $5A_{16} / 1E_{16} = 3_{16}$

Задание№5

Сложить числа $5E_{16}$ и 12_8. Сумму представить в десятичной системе счисления.

13) ($94_{10} + 10_{10} = 104_{10}$)

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Список используемых источников
5. Выводы и предложения
6. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что называют системой счисления?
2. Какая система счисления используется в компьютерной технике?
3. Почему человеку удобно пользоваться десятичной системой счисления?

4. Какие правила применяются для выполнения арифметических операций в позиционных системах счисления?
5. Что необходимо помнить при выполнении операций в различных позиционных системах счисления?
6. Как нужно поступить, если операции производятся над числами, представленными в различных позиционных системах счисления?

Практическое занятие №6 Элементы комбинаторики, теории множеств и математической логики.

Цель занятия:

1.Познакомить обучающихся с основными понятиями теории множеств, элементами комбинаторики и математической логики;

Исходные Теоретический материал, схемы, задания по вариантам

Содержание и порядок выполнения:

Изучить теоретическую часть

Выполнить задания

Теоретическая часть

Множество — это совокупность объектов произвольной природы, которая рассматривается как единое целое.

Примерами множеств могут служить: множество всех курсантов вашей группы, множество всех жителей Калининграда, множество всех натуральных чисел, множество всех решений не которого уравнения и т. п.

Множества принято обозначать прописными буквами латинского алфавита (A, B, C, …). Объекты, входящие в состав множества, называются его элементами.

Множество можно задать следующими способами:

1.перечислением всех его элементов;

2.характеристическим свойством его элементов.

В первом случае внутри фигурных скобок перечисляются все объекты, составляющие множество. Каждый объект, входящий в множество, указывается в фигурных скобках лишь один раз.

Например, запись M = {1, 3, 5, 7, 9} означает, что множество M состоит из чисел 1, 3, 5, 7 и 9.

Точно такой же смысл будет иметь запись M = {3, 1, 5, 9, 7}. Иначе говоря, порядок расположения элементов в фигурных скобках значения не имеет. Важно точно указать, какие именно объекты являются элементами множества.

число 5 является элементом множества M: $5 \in M1$);

число 4 не является элементом множества M: $4 \notin M$.

Это же множество можно задать с помощью характеристического свойства образующих его элементов — такого свойства, которым обладает каждый элемент, принадлежащий множеству, и не обладает ни один элемент, который ему не принадлежит. В нашем примере можно говорить о множестве натуральных однозначных нечётных чисел.

В рассматриваемом множестве M содержится 5 элементов. Это обозначают так: |M| = 5. Можно составить множество, содержащее любое число элементов. Например, множество всех корней уравнения $x^2 – 4x – 5 = 0$ конечно (два элемента), а множество всех точек прямой бесконечно. Множество, не содержащее ни одного элемента, называется пустым и обозначается символом ∅

Первый способ задания множеств применим только для конечных множеств, да и то при условии, что число элементов множества невелико. Вторым способом можно задавать как конечные, так и бесконечные множества.

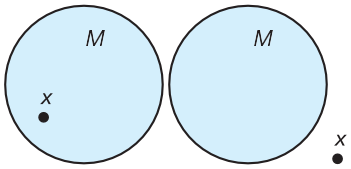
Из некоторых элементов множества M можно составить новое множество, например P: P = {1, 3, 5}.

Если каждый элемент множества P принадлежит множеству M, то говорят, что P есть подмножество M, и записывают: $P \subset M$.

Само множество M является своим подмножеством, т. к. каждый элемент M принадлежит множеству M. Пустое множество также является подмножеством M.

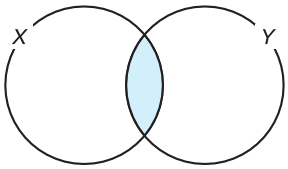
Работая с объектами какой-то определённой природы, всегда можно выделить «самое большое» или универсальное множество, содержащее все возможные подмножества. Пусть A — множество чётных чисел, B — множество натуральных чисел, C — множество чисел, кратных пяти. Тогда самым большим множеством, содержащим в себе множества A, B и C, а также другие подобные множества, будет множество целых чисел. Универсальное множество будем обозначать буквой U.

Для наглядного изображения множеств используются круги Эйлера (рис.). Точки внутри круга считаются элементами множества.



1) **пересечением** двух множеств X и Y называется множество их общих элементов.

Пересечение множеств обозначают с помощью знака ∩: X ∩ Y.



Пусть множества X и Y состоят из букв:

X = {ш, к, о, л, а};

Y = {у, р, о, к}.

Эти множества имеют общие элементы: к, о.

$X \cap Y$ = {к, о}.

Множества M и X не имеют общих элементов, их пересечение — пустое множество:

$M \cap X = \varnothing$.

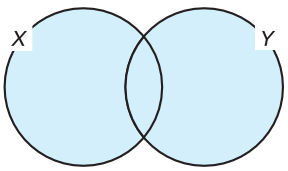
Пересечение множеств M и P есть множество P, а пересечение множеств M и M есть множество M:

$M \cap P = P$; $M \cap M = M$.

объединением двух множеств X и Y называется множество, состоящее из всех элементов этих множеств и не содержащее никаких других элементов.

Объединение множеств обозначают с помощью знака ∪:

$X \cup Y$.



Для наших примеров:

$X \cup Y$ = {ш, к, о, л, а, у, р};

$M \cup X$ = {1, 3, 5, 7, 9, ш, к, о, л, а};

$M \cup P = M$; $M \cup M = M$.

Подумайте, возможно ли равенство: $A \cup B = A \cap B$.

Пересечение и объединение выполняются для любой пары множеств.

дополнение — имеет смысл не для всех множеств, а только тогда, когда второе множество является подмножеством первого.

Пусть множество P является подмножеством множества M. дополнением P до M называется множество, состоящее из тех элементов M, которые не вошли в P. Дополнение P до M обозначают $\bar{P}$: $\bar{P}$ = {7, 9}.

Дополнение M до M есть пустое множество, дополнение пустого множества до M есть M: $\bar{M} = \varnothing$; $\bar{\varnothing} = M$.

Мощность множества

Мощностью конечного множества называется число его элементов. Мощность множества X обозначается |X|.

В рассмотренных выше примерах |X| = 5, |M| = 5.

Число элементов объединения двух непересекающихся множеств равно сумме чисел элементов этих множеств. Так, в объединении множеств M и X содержится 10 элементов: $|M \cup X| = 10$. Если же множества пересекаются, то число элементов объединения находится сложнее. Так, X состоит из 5 элементов, множество Y — из 4, а их объединение — из 7. Сложение чисел 5 и 4 даёт нам число 9. Но в эту сумму дважды вошло число элементов пересечения. Чтобы получить правильный результат, надо к числу элементов X прибавить число элементов Y и из суммы вычесть число элементов пересечения. Полученная формула подходит для любых двух множеств: $|X \cup Y| = |X| + |Y| - |X \cap Y|$. Это частный случай так называемого принципа включений-исключений.

Принципом включений-исключений называется формула, позволяющая вычислить мощность объединения (пересечения) множеств, если известны их мощности и мощности всех их пересечений (объединений).

Для случая объединения трёх множеств формула имеет вид:

$|X \cup Y \cup Z| = |X| + |Y| + |Z| - |X \cap Y| - |X \cap Z| - |Y \cap Z| + |X \cap Y \cap Z|$.

Аналогичные формулы справедливы и для пересечения множеств:

$|X \cap Y| = |X| + |Y| - |X \cup Y|$;

$|X \cap Y \cap Z| = |X| + |Y| + |Z| - |X \cup Y| - |X \cup Z| - |Y \cup Z| + |X \cup Y \cup Z|$.

Пример. В зимний оздоровительный лагерь отправляется 100 старшеклассников. Почти все они увлекаются сноубордом, коньками или лыжами. При этом многие из них занимаются не одним, а двумя и даже тремя видами спорта. Организаторы выяснили, что всего кататься на сноуборде умеют 30 ребят, на лыжах — 28, на коньках — 42. Всего умением кататься на лыжах и сноуборде из них могут похвастаться 8 ребят, на лыжах и коньках — 10, на сноуборде и коньках — 5, но только трое из них владеют всеми тремя видами спорта. Сколько ребят не умеет кататься ни на сноуборде, ни на лыжах, ни на коньках?

Обозначим через S, L и K множества сноубордистов, лыжников и любителей коньков соответственно. Тогда $|S| = 30$, $|L| = 28$ и $|K| = 42$. При этом $|S \cap L| = 8$, $|K \cap L| = 10$, $|S \cap K| = 5$, $|S \cap L \cap K| = 3$.

Объединение множеств S, L и K — это множество ребят, увлекающихся хотя бы каким-то видом спорта.

По формуле включений-исключений находим:

$|S \cup L \cup K| = 30 + 28 + 42 - 8 - 10 - 5 + 3 = 80$.

Таким образом, из 100 старшеклассников 20 не умеют кататься ни на сноуборде, ни на лыжах, ни на коньках.

Задания

1.Если множество X — это множество натуральных чисел, делящихся нацело на 2, а Y — множество натуральных чисел, делящихся нацело на 3, то что будет: пересечением этих множеств;

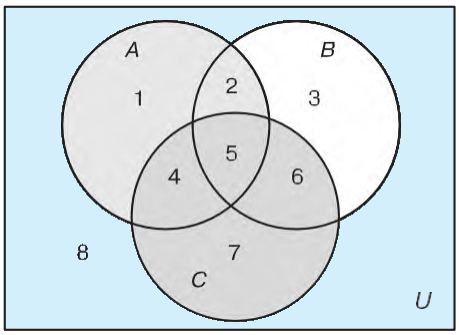
объединением этих множеств?

2.Пусть множество X — это множество натуральных чисел, делящихся нацело на 18, а Y — множество натуральных чисел, делящихся нацело на 14. Укажите наименьшее число, входящее:

в пересечение этих множеств;

в объединение этих множеств?

3.Пусть A, B и C — некоторые множества, обозначенные кругами, U — универсальное множество.



С помощью операций объединения, пересечения и дополнения до универсального множества выразите через A, B и C следующие множества:

1)1 ∪ 2∪3 ∪ 4 ∪5 ∪6;

2)2 ∪ 5;

3)5;

4)2 ∪ 4 ∪ 5 ∪ 6;

5)1 ∪ 2 ∪ 3;

6)8.

4.В первую смену в лагере «Дубки» отдыхали: 30 отличников,
28 победителей олимпиад и 42 спортсмена. При этом 10 человек были и отличниками, и победителями олимпиад, 5 — отличниками и спортсменами, 8 —

спортсменами и победителями олимпиад, 3 — и отличниками, и спортсменами, и победителями олимпиад. Сколько ребят отдыхало в лагере?

5.Старшеклассники заполняли анкету с вопросами об экзаменах по выбору. Оказалось, что выбрали они информатику, физику и обществознание. В классе 38 учеников. Обществознание выбрал 21 ученик, причём трое из них выбрали ещё и информатику, а шестеро — ещё и физику. Один ученик выбрал все три предмета. Всего информатику выбрали 13 учеников, пятеро из которых указали в анкете два предмета. Надо определить, сколько же учеников выбрали физику.

Выводы и предложения о проделанной работе

Содержание отчета:

Наименование практического занятия

Цель занятия

Вариант задания

Список используемых источников

Выводы и предложения

Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

Что называют множеством?

Что называют пересечением двух множеств?

Что называют объединением двух множеств?

Что называют подмножеством множества?

Что называют мощностью множества?

Что позволяет вычислить формула включений-исключений?

Практическое занятие №7 Структура информации. Графы. Введение и понятия. Способы задания графов. Алгоритм построения дерева решений.

Цель занятия:

1.Познакомить обучающихся с основными понятиями теории графов;

Исходные Теоретический материал, схемы, задания о вариантам

Содержание и порядок выполнения:

Изучить теоретическую часть

Выполнить задания

Теоретическая часть

Понятие графа

Граф – это множество точек или вершин и множество линий или ребер, соединяющих между собой все или часть этих точек.

Вершины, прилегающие к одному и тому же ребру, называются смежными.

Два ребра, у которых есть общая вершина, также называютсясмежными(или соседними).

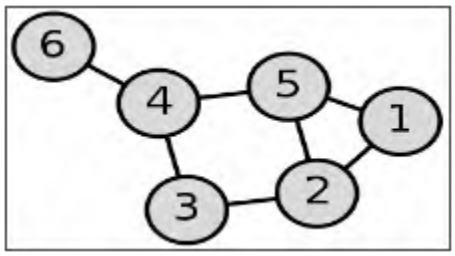


Рис. 1. Граф с шестью вершинами и семью ребрами

Элементы графа

Петля – это дуга, начальная и конечная вершина которой совпадают.

Пустым (нулевым) называется граф без ребер.

Полным называется граф, в котором каждые две вершины смежные.

Нулевой граф – граф, состоящий из «изолированных» вершин.

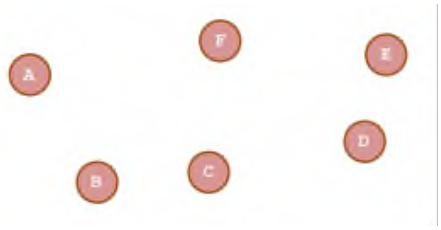


Рис. 2. Нулевой граф

Неполный граф – графы, в которых не построены все возможные ребра.

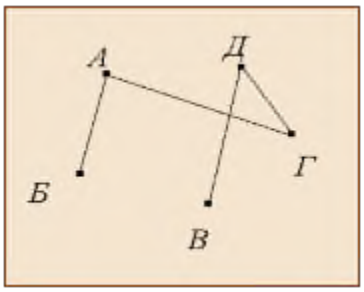


Рис. 3 Неполный граф

Степень графа

Количество рёбер, выходящих из вершины графа, называется степенью вершины. Вершина графа, имеющая нечётную степень, называется нечетной, а чётную степень – чётной.

Если степени всех вершин графа равны, то граф называется однородным. Таким образом, любой полный граф — однородный.

Кружки называются вершинами графа, линии со стрелками - дугами, без стрелок - ребрами. Граф, в котором направление линий не выделяется (все линии являются ребрами), называется неориентированным (рис. 1, А); граф, в котором направление линий принципиально (линии являются дугами) называется ориентированным (рис. 1, Б).

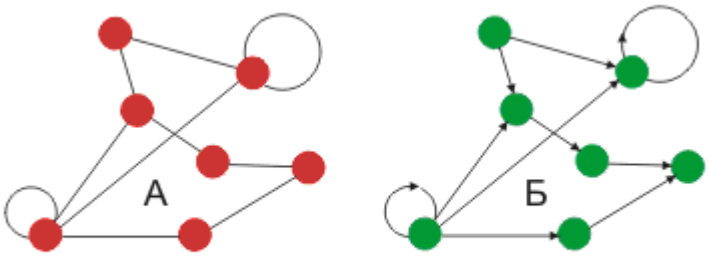


рис.1

Ориентированный граф

Граф называется ориентированным (или орграфом), если некоторые ребра имеют направление. Это означает, что в орграфе некоторая вершина может быть соединена с другой вершиной, а обратного соединения нет. Если ребра ориентированы, что обычно показывают стрелками, то они называются дугами.

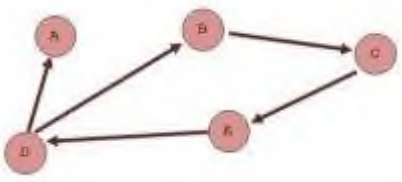


Рис. 4. Ориентированный граф

Ориентированный и неориентированный графы

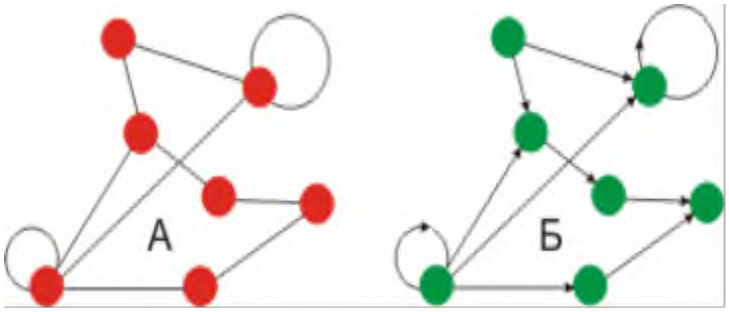


Рис. 5. Примеры неориентированного и ориентированного графов (А и Б)

Задание 1. Построить граф по заданному условию:

В соревнованиях по футболу участвуют 6 команд. Каждую из команд обозначили буквами A, B, C, D, E и F. Через несколько недель некоторые из команд уже сыграли друг с другом:

A с C, D, F;

B с C, E, F;

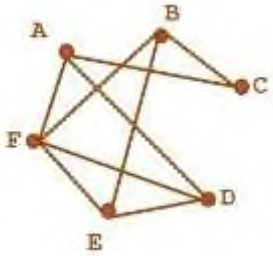
C с A, B;

D с A, E, F;

E с B, D, F;

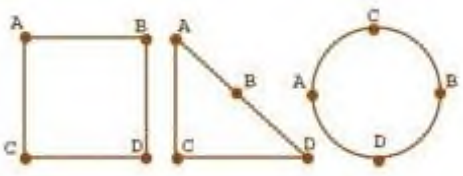
F с A, B, D.

Ответ:



Задание 2.

Определить изображают ли фигуры на рисунке один и тот же граф или нет.



1. 2) 3)

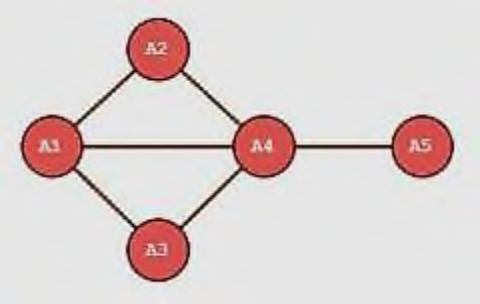
Ответ: Рисунок 1 и рисунок 2 являются изображениями одного графа. Рисунок 3 изображением другого графа.

Путь в графе

Путём в графе называется такая последовательность ребер, в которой каждые два соседних ребра имеют общую вершину и никакое ребро не встречается более одного раза.

Задание 3. Определить какая из перечисленных последовательностей путём не является.

1. (A1 A4); (A4 A5).
2. (A1 A2); (A2 A4); (A4 A5).
3. (A1 A4); (A4 A2); (A2 A1); (A1 A4); (A4, A5).
4. (A1 A4); (A4 A2); (A2 A1); (A1 A3); (A3 A4); (A4, A5).

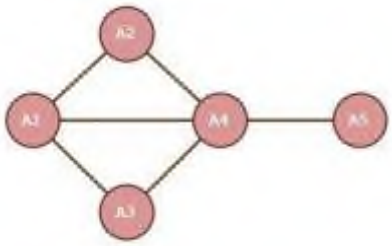


Ответ: Третья последовательность (A1 A4); (A4 A2); (A2 A1); (A1 A4); (A4, A5)

Путь называется простым, если он не проходит ни через одну из вершин графа более одного раза.

Задание 4. Определите, какие последовательности ребер являются путями, и какие из них простые. Если последовательность не является путем укажите почему.

1. (A1 A4); (A4 A5).
2. (A1 A2); (A2 A4); (A4 A5).
3. (A1 A4); (A4 A2); (A2 A1); (A1 A4); (A4, A5).
4. (A1 A4); (A4 A2); (A2 A1); (A1 A3); (A3 A4); (A4, A5).



Ответ: Первая, вторая и четвертая последовательности являются путями, а третья нет, т.к. ребро (A1, A4) повторяется.

Первая и вторая последовательность являются простыми путями, а четвертая нет, т.к. вершины A1 и A4 повторяются.

Понятие цикла в графе

Циклом называется путь, в котором совпадают его начальная и конечная вершины.

Простым циклом в графе называется цикл, не проходящий ни через одну из вершин графа более одного раза.

Задание 5. Назовите в графе циклы, содержащие

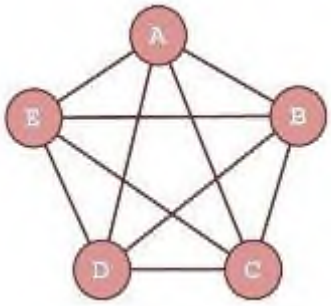
a) 4 ребра;

b) 6 ребер;

c) 5 ребер;

d) 10 ребер.

Какие из этих циклов являются простыми?



Ответ:

Решение:

a. (AB, BC, CE, EA), (CD, DA, AB, BC), (EB, BC, CD, DE) и т.д. – простые циклы.

b. (DB, BE, EA, AB, BC, CD), (EC, CA, AB, BC, CD, DE) ит.д. – циклы.

с. (AB, BC, CD, DE, EA), (AC, CE, EB, BD, DA) ит.д. – простыециклы.

(AC, CE, EB, BD, DA, AB, BC, CD, DE, EA), (EB, BD, DA, AC, CE, EA, AB, BC, CD, DE) и т.д. – циклы.

1 Граф G - совокупность двух множеств: вершин V и ребер E, между которыми определено отношение инцидентности. Если |V(G)|=n, |E(G)|=m, то граф G есть (n,m) граф, где n - порядок графа, m - размер графа.

2 Каждое ребро e из E инцидентно ровно двум вершинам v', v'', которые оно соединяет. При этом вершина v' и ребро e называются инцидентными друг другу, а вершины v' и v'' называются смежными.

3 Ребро (v',v'') может быть ориентированным и иметь начало (v') и конец (v'') (дуга в орграфе).

4 Ребро (v,v) называется петлей (концевые вершины совпадают).

5 Граф, содержащий ориентированные ребра (дуги), называется орграфом.

6 Граф, не содержащий ориентированные ребра (дуги), называется неографом.

7 Ребра, инцидентные одной паре вершин, называются параллельными или кратными.

8 Конечный граф - число вершин и ребер конечно.

9 Пустой граф - множество ребер пусто (число вершин может быть произвольным).

10 Полный граф - граф без петель и кратных ребер, каждая пара вершин соединена ребром.

11 Локальная степень вершины - число ребер ей инцидентных.

12 Внеографе сумма степеней всех вершин равна удвоенному числу ребер (лемма о рукопожатиях). Петля дает вклад, равный 2 в степень вершины.

13В орграфе сумма входящих ребер всех вершин равна сумме исходящих ребер всех вершин и равна числу ребер графа.

14 Графы равны, если множества вершин и инцидентных им ребер совпадают.

15 Графы, отличающиеся только нумерацией вершин и ребер, называются изоморфными.

16 Способы задания графов: − явное задание графа как алгебраической системы; − геометрический; − матрица смежности; − матрица инцидентности

17 Матрица инцидентности: По вертикали указываются вершины, по горизонтали - ребра. aij=1 если вершина i инцидентна ребру j, в противном случае aij=0. Если ребро - петля, то aij=2. Матрицей инцидентности (инциденций) ориентированного графа называется матрица, для которой aij=1, если вершина является началом дуги ,aij= – 1, если является концом дуги, в остальных случаях aij=0.

18 Матрица смежности - квадратная симметричная матрица. По горизонтали и вертикали - все вершины.aij= число ребер, соединяющее вершины i,j. Матрицей смежности ориентированного графа называется матрица, для которой aij=1, если вершина является началом дуги, в остальных случаях aij=0.

19 Маршрут - последовательность ребер, в которых каждые два соседних ребра имеют общую вершину.

20 Маршрут, в котором начало и конец совпадают - циклический.

21 Маршрут в неографе, в котором все ребра разные - цепь.

22 Маршрут в орграфе, в котором все дуги разные - путь.

23 Вершины связанные, если существует маршрут из одной вершины в другую.

24 Связанный граф - если все его вершины связаны.

25 Плоский граф - граф с вершинами, расположенными на плоскости и непересекающимися ребрами.

26 Вершины графа, которые не принадлежат ни одному ребру, называются изолированными.

27 Дерево - связный граф без циклов.

Пример выполнения:

Исходные данные:

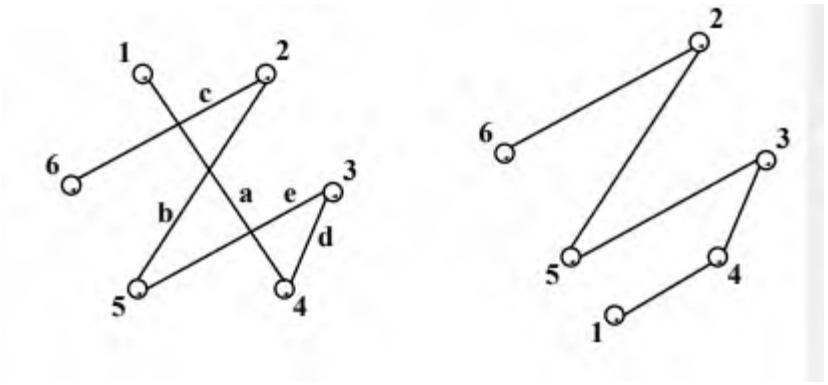
1 Задать неограф, представленный множеством вершин и ребер, графически и матрицами, преобразовать граф в плоский, вычислить степени его вершин.

V = {1; 2; 3; 4; 5; 6};E = {a; b; c; d; e}

E = {(1; 4); (2; 5); (2; 6); (3; 4); (3; 5)}

Решение:

1Изобразим граф, соединив вершины: Ребро a соединяет вершины 1 и 4, b соединяет вершины 2 и 5 и т. д. Затем преобразуем этот граф в плоский:



2 Составим матрицу смежности. В первом столбце и первой строке выпишем вершины. Ребру a инцидентны вершины 1 и 4, следовательно в колонке 1 и строке 4 ставим 1, а также колонке 4 и строке 1 ставим 1. Ребру b инцидентны вершины 2 и 5, следовательно в колонке 2 в строке 5 и колонке 5 строке 2 ставим 1 и т.д. Остальные ячейки таблицы содержат нули.

3 Составим матрицу инцидентности. В первом столбце выпишем вершины, первой строке – ребра. Ребру a инцидентны вершины 1 и 4, следовательно в колонке a в строке 1 и строке 4 ставим 1. Ребру b инцидентны вершины 2 и 5, следовательно в колонке b в строке 2 и строке 5 ставим 1и т.д. Остальные ячейки таблицы заполняем нулями

Матрица смежности

	1	2	3	4	5	6
1	0	0	0	1	0	0
2	0	0	0	0	1	1
3	0	0	0	1	1	0
4	1	0	1	0	0	0
5	0	1	1	0	0	0
6	0	1	0	0	0	0

Матрица инцидентности

	a	b	c	d	e
1	1	0	0	0	0
2	0	1	1	0	0
3	0	0	0	1	1
4	1	0	0	1	0
5	0	1	0	0	1
6	0	0	1	0	0

4 Вычислим степени вершин:

ρ (1) = 1 ρ (2) = 2 ρ (3) = 2 ρ (4) = 2 ρ (5) = 2 ρ (6) = 1

ρ (1) + ρ (2) + ρ (3) + ρ (4) + ρ (5) + ρ (6) = 10 = 2 * q

q = 5 (ребер 5)

Исходные данные:

2Задать граф, представленный матрицей инцидентности, алгебраически, графически и матрицей смежности, преобразовать граф в плоский, вычислить степени его вершин

	a	b	c	d	e	f
1	-1	-1	0	0	0	0
2	1	0	-1	1	0	0
3	0	0	0	-1	0	0
4	0	0	1	0	1	0
5	0	0	0	0	-1	-1
6	0	1	0	0	0	1

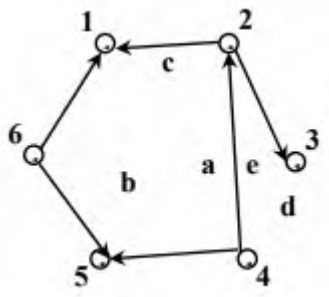
Решение:

1 Количество вершин – 6. V = {1; 2; 3; 4; 5; 6}.

2 Ребро a выходит из вершины 2, т.к. в ячейке (2; 1) стоит 1, а приходит в вершину 1 (в ячейке (1; 1) находится -1) и т.д.

Получим множество E = {(2; 1); (6; 1); (4; 2); (2; 3); (4; 5); (6; 5)}

3Изобразим граф, соединив вершины, этот граф уже плоский, т.к. ребра не пересекаются:



4 Составим матрицу смежности.

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	1	0	0	0
3	0	0	0	0	0	0
4	0	1	0	0	1	0
5	0	0	0	0	0	0
6	1	0	0	0	1	0

5 Вычислим степени вершин:

ρ1 (1) = 0

ρ1 (2) = 2

ρ1 (3) = 0

ρ1 (4) = 2

ρ1 (5) = 0

ρ1 (6) = 2

ρ2 (1) = 2

ρ2 (2) = 1

ρ2 (3) = 1

ρ2 (4) = 0

ρ2 (5) = 2

ρ2 (6) = 0

ρ1 (1) + ρ1 (2) + ρ1 (3) + ρ1 (4) + ρ1 (5) + ρ1 (6) = 6

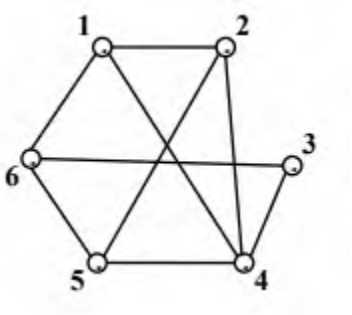
ρ2 (1) + ρ2 (2) + ρ2 (3) + ρ2 (4) + ρ2 (5) + ρ2 (6) = 6

q = 6(ребер 6)

Задания

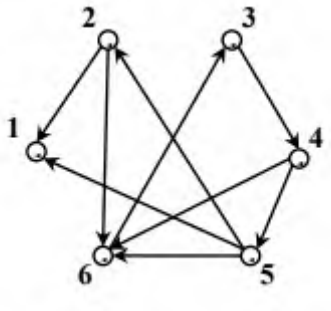
1. Орграф V = {1; 2; 3; 4; 5; 6}

E = {(1; 3); (1; 6); (2; 5); (3; 2); (3; 4); (4; 1); (4; 5); (5; 3); (6; 2)}

2. 

3 НеографV = {1; 2; 3; 4; 5; 6}

E = {(1; 2); (1; 5); (2; 3); (3; 1); (3; 4); (4; 2); (4; 5); (4; 6); (5; 3)}

4. 

5.

	a	b	c	d	e	f	g	h	i
1	1	1	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	1	0
3	0	0	0	0	1	1	0	0	0
4	0	1	0	0	0	0	0	1	1
5	0	0	0	1	1	0	1	0	0
6	0	0	1	0	0	1	1	0	1

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Список используемых источников
4. Выводы и предложения
5. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что такое граф?

2. Что такое инцидентное ребро или инцидентная вершина?

3 Какое ребро называется ориентированным?

4 Что такое кратные ребра?

5 Что такое неограф?

6 Что такое оргаф?

Изобразите графически:

7. Неориентированное и ориентированное ребро;

8. Неориентированный граф G(V,E), заданный множеством

V={v0, v1, v2, v3, v4, v5}

E = {e1=(v0, v1), e2=(v0, v2}, e3=(v1, v2), e4=(v1,v4), e5=(v2, v5), e6=(v3, v4)}

9. Плоский граф;

10. Полный неориентированный граф на трех, четырех и пяти вершинах;

11. Неполный ориентированный граф на пяти вершинах;

12. Петлю графа.

13. Цикл графа.

Практическое занятие № 8 Решение логических задач с помощью графов. Анализ алгоритмов в профессиональной области.

Цель занятия:

1.обобщить и систематизировать знания о графах, их видах и свойствах;

2.рассмотреть типы задач, которые можно решить с использованием графов;

3.отработать навыки преобразования ориентированного графа в дерево;

4.сформировать навыки построение путей в графе, поиска кратчайшего пути;

5.Формировать ОК5, ОК6, ЛР26, ЛР30 .

Содержание и порядок выполнения задания:

Изучить теоретическую часть

Выполнить задания

Теоретическая часть

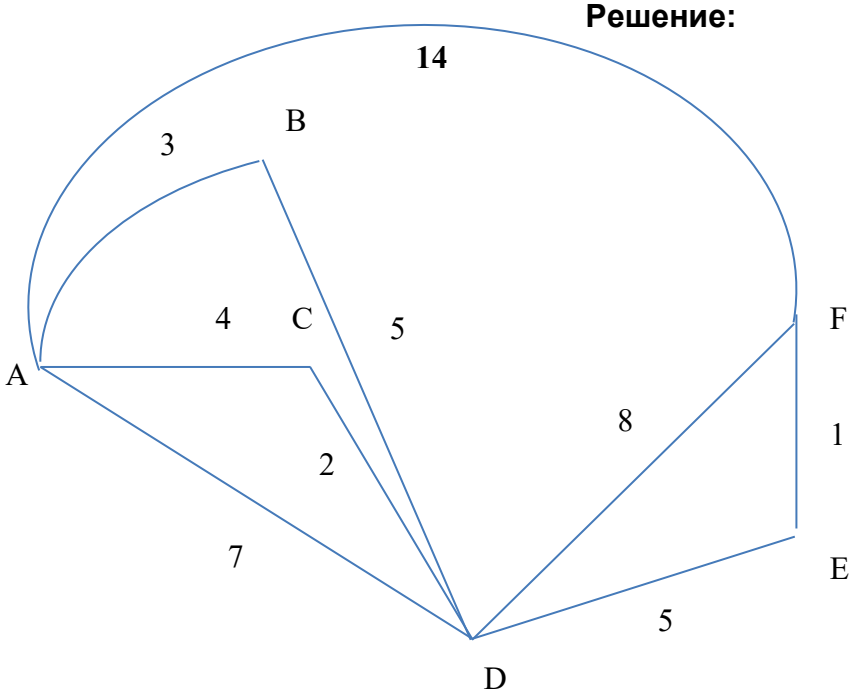
При решении логических задач по информатике часто приходится сталкиваться с заданиями, которые вызывает затруднение при решении. Если же построить граф, то задача существенно упрощается. Теория графов позволяет решать наиболее легким способом, наглядно многие логические задачи, которые способствуют развитию мышления и интеллекта. Слово «граф» означает картинку, где нарисовано несколько точек, некоторые из которых соединены линиями. Графами являются блок – схемы программ для ЭВМ, сетевые графики строительства, где вершины – события, означающие окончания работ на некотором участке, а ребра, связывающие эти вершины, - работы, которые возможно начать по совершении одного события и необходимо выполнить для совершения следующего.

Задача 1. Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.

	A	B	C	D	E	F
A		3	4	7		14
B	3			5		
C	4			2		
D	7	5	2		5	8
E				5		1
F	14			8	1	

Определите длину кратчайшего пути между пунктами A и F при условии, что передвигаться можно только по указанным в таблице дорогам.

Решение:



AF=14

AF=AB+BD+DF=3+5+8=16

AF=AC+CD+DF=4+2+8=14

AF=AD+DE+EF=7+5+1=13

AF= AC+ CD+ DE+ EF=4+2+5+1=12

Ответ: 12

Задача 2

Сколькими способами можно рассадить в ряд на три стула трёх студентов? Выписать все возможные случаи.

Решение этой задачи удобнее всего представить в виде дерева. За его корневую вершину возьмём произвольную точку плоскости О.

На первый стул можно посадить любого из трёх студентов — обозначим их А, В и С. На схеме это соответствует трём ветвям, исходящим из точки О (рис. 1).

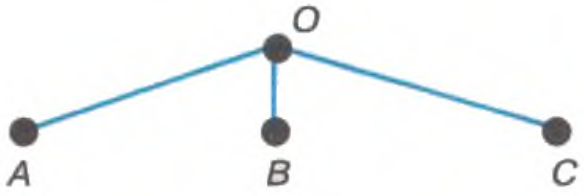


Рис. 1

Посадив на первый стул студента А, на второй стул можно посадить студента В или С. Если же на первый стул сядет В, то на второй можно посадить А или С. Если на первый стул сядет С, то на второй можно будет посадить А или В. Это соответствует на схеме двум ветвям, исходящим из каждой вершины первого уровня (рис. 2).

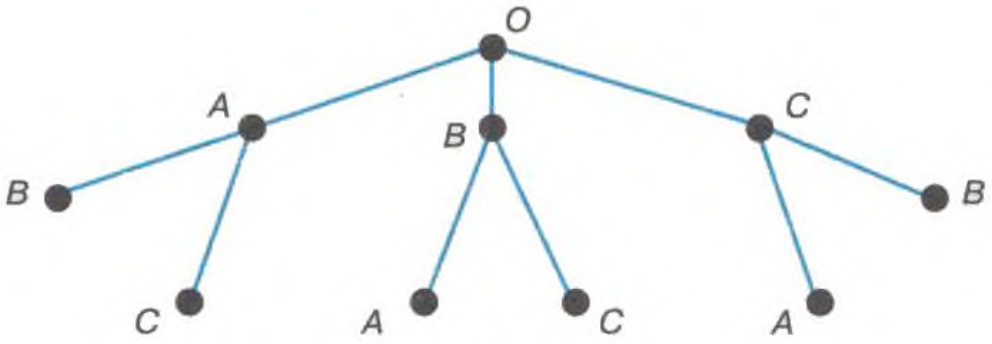


Рис. 2

Очевидно, что третий стул в каждом случае займёт оставшийся студент. Это соответствует одной ветви дерева, которая «вырастает» на каждой из предыдущих ветвей (рис. 3).

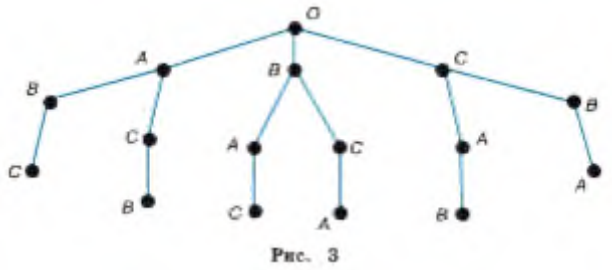


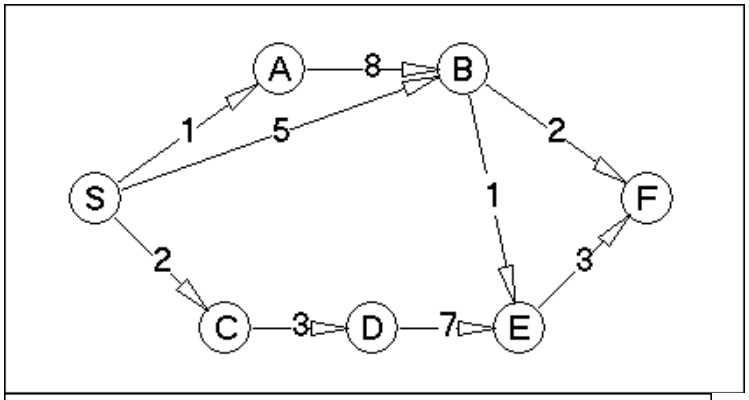
Рис. 3

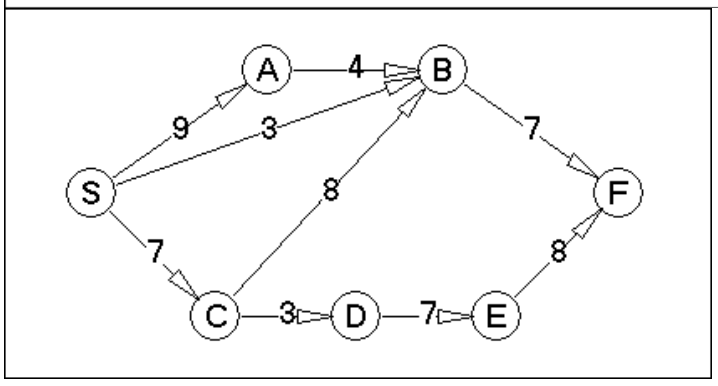
Выпишем все пути от вершин первого уровня к вершинам третьего уровня: А-В-Су А-С-В, В-А-С, В-С-А, С-А-Б, С-В-А. Каждый из выписанных путей определяет один из вариантов рассаживания учеников на стулья. Так как других путей нет, то искомое число способов — 6.

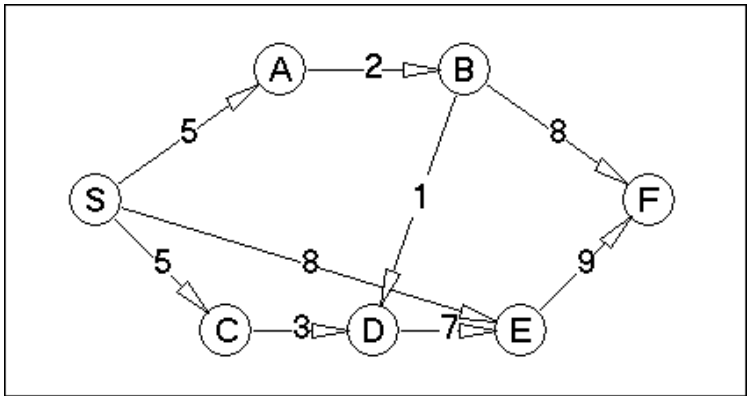
Дерево можно не строить, если не требуется выписывать все возможные варианты, а нужно просто указать их число. В этом случае рассуждать нужно так: на первый стул можно усадить одного из трёх человек, на второй — одного из двух оставшихся, на третий — одного оставшегося: 3-2-1 = 6.

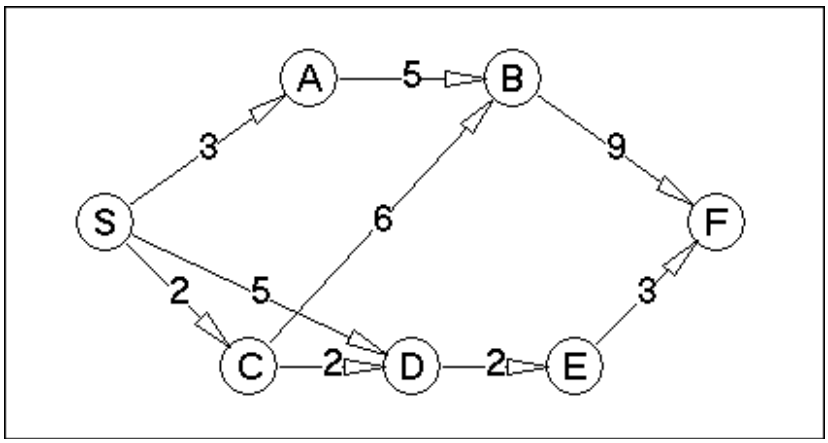
Задание1

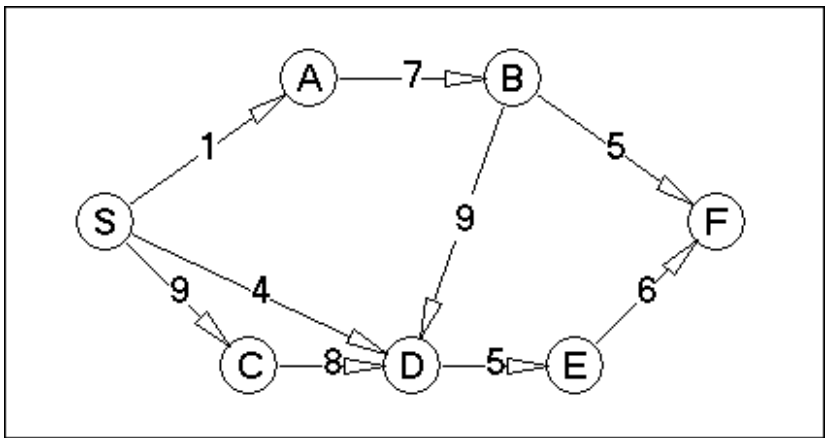
Для данного графа найти наикратчайшее расстояние от S до F











Задание 2

Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице:

	A	B	C	D	E	F
A		6	4	2	1	
B	6		1			
C	4	1		3		2
D	2		3		2	
E	1			2		6
F			2		6	

Определите длину кратчайшего пути между пунктами A и F. Передвигаться можно только по дорогам, протяжённость которых указана в таблице.

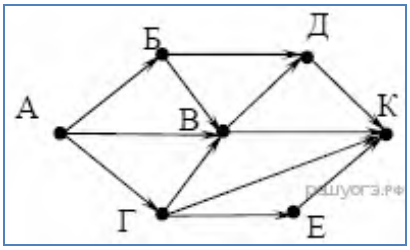
Задание 3

В таблице приведена стоимость перевозок между пятью железнодорожными станциями, обозначенными буквами A, B, C, D и E. Построить схему, соответствующую таблице.

	A	**B**	**C**	**D**	**E**
A		1	4		1
B	1			3	
C	4				2
D		3			
E	1		2		

Задание 4

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Список используемых источников
5. Выводы и предложения
6. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Скажите, что такое граф?
2. Что является вершинами и ребрами графа?
3. Какой граф можно назвать взвешенным?
4. Какой граф можно назвать сематической сетью?

Практическое занятие № 9 Алгоритмы и способы их описания. Линейные и условные алгоритмы. (составление трассировочных таблиц) Описание алгоритмов с помощью блок-схем

Цели занятия:

1.Сформировать представление об алгоритме и его свойствах;

2. Сформировать представление о способах их описания алгоритмов;

3.Сформировать представление о типах алгоритмов;

4. Сформировать представление о линейной и условных алгоритмических конструкциях;

5.Научить создавать трассировочные таблицы линейной и условных алгоритмических конструкций.

6. Формировать ОК5, ОК6, ЛР26, ЛР30

Исходные данные: теоретический материал

Содержание и порядок выполнения задания

Изучите теоретическую часть, данную в практическом занятии.

Изучите примеры и выполните задания. Результат работы алгоритма определяется с помощью трассировочных таблиц.

Теоретическая часть

Слово ***алгоритм*** происходит от латинской формы написания имени великого математика IX века **Аль-Хорезми**, который сформулировал правила выполнения арифметических действий.

Первоначально под алгоритмами понимали только правила выполнения четырёх арифметических действий над многозначными числами.

Алгоритм – описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.

Алгоритмизация – процесс разработки алгоритма (плана действий) для решения задачи.

Шаг алгоритма – это каждое отдельное действие алгоритма.

Исполнитель – это объект, умеющий выполнять определенный набор действий. Исполнителем может быть человек, робот, животное, компьютер.

Система команд исполнителя (СКИ) – это все команды, которые исполнитель умеет выполнять.

Среда исполнителя – обстановка, в которой функционирует исполнитель.

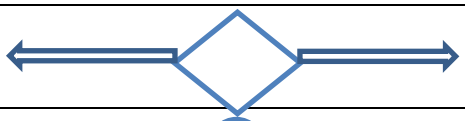
Свойства алгоритма:

- Дискретность - (прерывность, раздельность) – разбиение алгоритма на шаги
- Результативность - получение результата за конечное количество шагов
- Массовость - использование алгоритма для решения однотипных задач
- Конечность - каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения
- Детерминированность - (определенность, точность) – каждое действие должно быть строго и недвусмысленно определено

Способы записи алгоритмов (блок-схема)

Условное обозначение	Назначение блока
	Начало или **конец** алгоритма
	Ввод или **вывод** данных.

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

	Внутри блока перечисляются данные через запятую.
	Процесс.Действие. Внутри блока записываются математические формулы и операции для обработки данных.
	**Проверка условия.** Внутри блока записываются логические условия. Имеет два выхода **Да(+)**и**Нет(-)**.
	Соединительный блок
	Блок вывода информации на печатающее устройство
	Блок вывода информации на экран дисплея
	Направление.

Алгоритмы могут быть заданы:

Словесно- ***Словесное*** задание описывает алгоритм с помощью слов и предложений естественного языка.

Таблично- ***Табличное*** задание служит для представления алгоритма в форме таблиц

Формульно – ***Формульное*** *задание* расчётными формулами.

Программой- задание ***программы*** *на языке программирования*

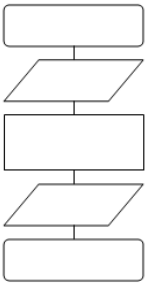
Графическое- задание или ***блок-схема*** – способ представления алгоритма с помощью геометрических фигур, называемых ***блоками***.

Типы алгоритмов

Алгоритмы бывают:

- *линейные*
- *разветвляющиеся*
- *циклические*

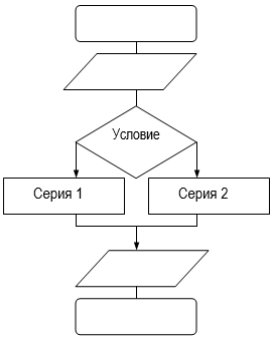
Алгоритм, в котором команды выполняются последовательно одна за другой, называется ***линейным алгоритмом.***



В ***разветвляющиеся алгоритмы*** входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (серий).

В алгоритмической структуре ***«ветвление»*** та или иная серия команд выполняется в зависимости от истинности ***условия.***

Условие может быть либо истинным, либо ложным.

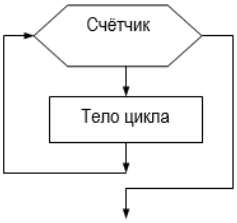


В ***циклические алгоритмы*** входит последовательность команд, выполняемая многократно. Такая последовательность команд называется ***телом цикла.***

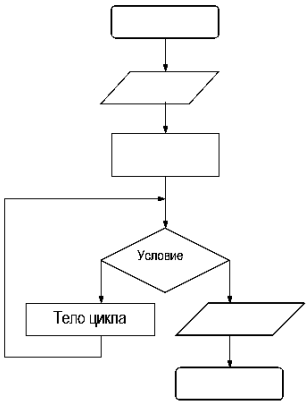
В алгоритмической структуре ***«цикл»*** серия команд (тело цикла) выполняется многократно.

Циклические алгоритмические структуры бывают двух типов:

- *циклы со счётчиком*, в которых тело цикла выполняется определённое количество раз;



- *циклы с условием, в которых тело цикла выполняется, пока условие истинно.*



Примеры выполнения заданий работы

Пример 1. Определить площадь трапеции по введенным значениям оснований (a и b) и высоты (h).

Запись алгоритма в виде блок-схемы (рис. 1):

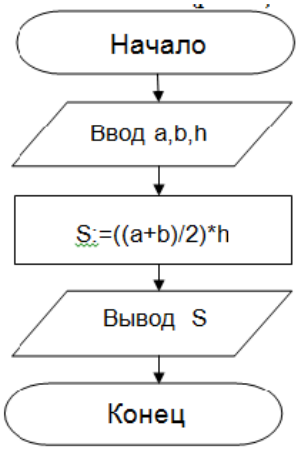
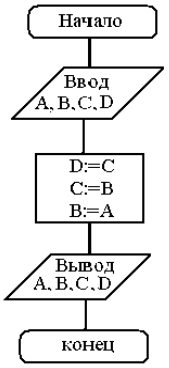


Рисунок 1. Блок-схема линейного алгоритма

Пример 2. Дан алгоритм в виде блок-схемы (рис. 11).



Найти A, B, C, D, если изначально:

a)A=0,B=0,C=5,D=10

Результат работы алгоритма определяется с помощью трассировочных таблиц:

а) A=0, B=0, C=5, D=10.

Шаг	1	
	A	0
	B	0

Исходные значения	C	5
	D	10
Результат выполнения	A	0
	B	0
	C	0
	D	5
Вывод значений	0,0,0,5	

Пример 2. Определить среднее арифметическое двух чисел, если **a** положительное и частное (a/b) в противном случае.

Запись алгоритма в виде блок-схемы (рис. 2):

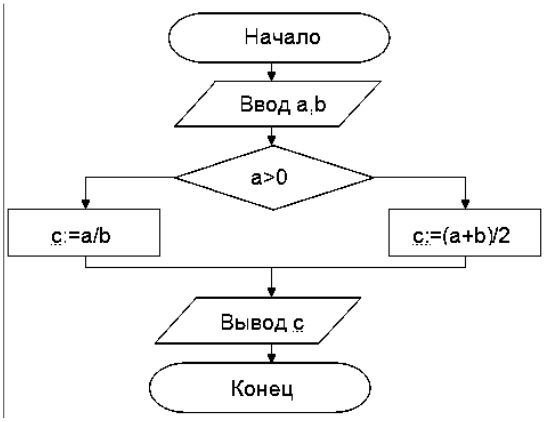


Рисунок 2. Блок-схема алгоритма с ветвлением

***Задание* 1**

Реализован некоторый алгоритм в виде блок- схемы. Найти A, B на выходе блок-схемы с заполнение трассировочных таблиц, если изначально, :

1 вариант A=0,B=0;

2вариант A=0,B=5;

3 вариант A=10,B=20;

4 вариант A=10,B=10.

5 вариант A=3, B=0;

6 вариант A=0, B=2;

10 вариант A=2, B=4;

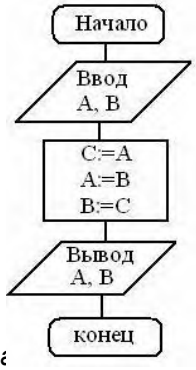
11 вариант A=12, B=24;

12 вариант A=11, B=19

13 вариант A=3, B=8;

14 вариант A=2, B=7;

15 вариант A=19, B=20;



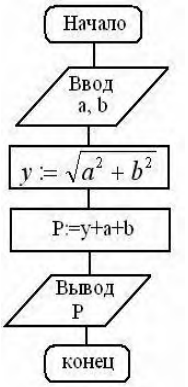
7вариант A=1,B=2; 16 вариант A=15, B=10

8 вариант A=12, B=15 17 вариант A=60, B=20;

9 вариант A=0, B=0; 18 вариант A=20, B=65;

***Задание* 2.**

Даны длины двух катетов (a, b) прямоугольного треугольника. Определить периметр этого треугольника (P)с заполнением трассировочных таблиц (см. блок-схему на рис.), если:



а) a=3,b=4;

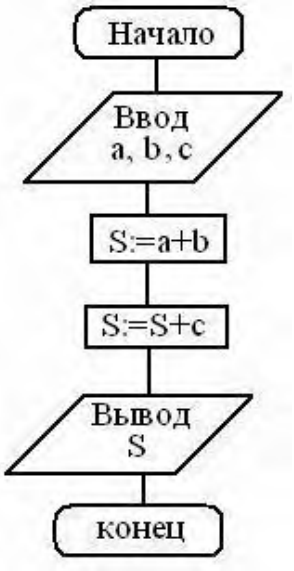
б) a=0,b=3;

в)a=6,b=8;

г)a=9,b=12.

***Задание*3.**

Реализован некоторый алгоритм в виде блок- схемы (рис.). По данной блок-схеме вычислить S, заполнить трассировочные таблицы, если:



а)a=1, b=2, c=3;

б)a=9, b=0, c=1;

в) a=5, b=6, c=9.

***Задание*4.**

Дана блок-схема (рис.19). Начальные условия: a=8, b=2. Тогда после исполнения алгоритмазначение переменной g будет равно …

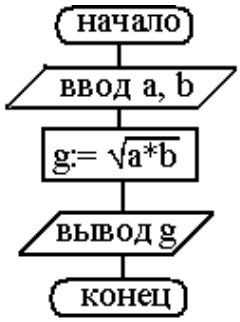
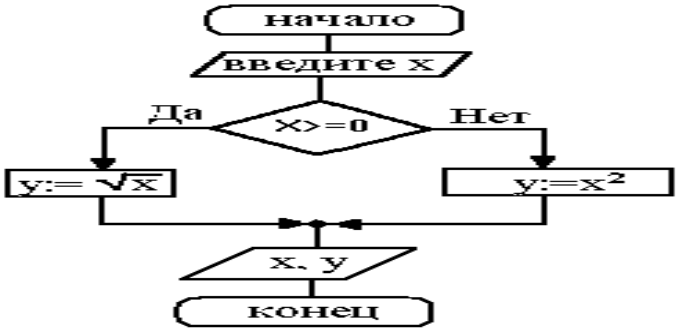


Рис. 19

Задание5. Вычислить значение функции y. Результат работы алгоритма определяется с помощью трассировочных таблиц:

(см. блок-схему на рис.), если: а) x=0; б) x=1; в) x= – 5.

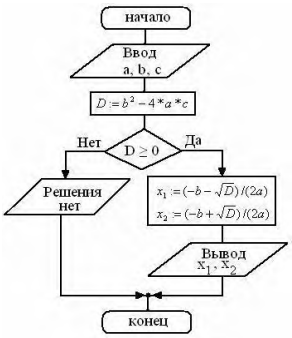


Задание6. Используя блок-схему (рис.), найти корни уравнения $ax^2 + bx + c = 0$ ($a \neq 0$) , если:

а)a=1, b=2,

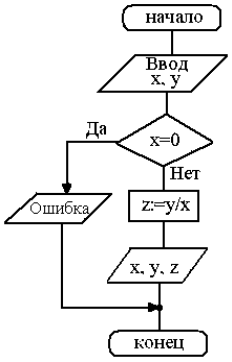
б)a=1, b=4,

в) a=3, b=–8,c=3.



Результат работы алгоритма определяется с помощью трассировочных таблиц

Задание7. Реализован некоторый алгоритм в виде блок- схемы (рис. 22). Что получится на выходе блок-схемы, если:



а) x=0, y=1; б) x=2, y=4; в) x=6, y=0?

Результат работы алгоритма определяется с помощью трассировочных таблиц

Задание8. На блок-схеме (рис. 23) представлен алгоритм вычисления стоимости покупки с учетом скидки, где a – цена, b

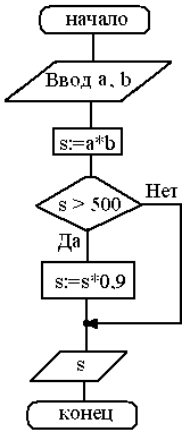
– количество, s – сумма. Какой будет результат на выходе блок- схемы, если:

а) a=50, b=8;

б) a=200, b=5;

в) a=300, b=1;

г) a=800, b=4?



Результат работы алгоритма определяется с помощью трассировочных таблиц

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Список используемых источников
4. Выводы и предложения
5. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что такое алгоритм?

2. Назовите свойства алгоритма и способы записи алгоритмов.

4. Изобразите элементы блок-схемы.

5. Какие существуют виды алгоритмов.

6.Как представляется блок-схемой циклический алгоритм с пост условием?

7.Как представляется блок-схемой циклический алгоритм с пред условием?

Практическое занятие № 10 Циклические алгоритмы (составление трассировочных таблиц). Описание алгоритма с помощью блок-схем

Цели занятия:

1. Сформировать представление о циклических алгоритмических конструкциях;

2.Научить создавать трассировочные таблицы циклических алгоритмических конструкций.

3. Формировать ОК5, ОК6, ЛР26, ЛР30

Исходные данные: теоретический материал

Содержание и порядок выполнения задания

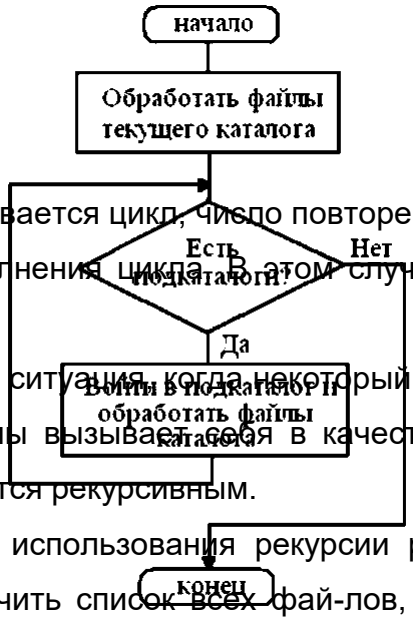
Изучите теоретическую часть, данную в практическом занятии.

Изучите примеры и выполните задания. Результат работы алгоритма определяется с помощью трассировочных таблиц.

Теоретическая часть

Циклическая структура (или повторение) предусматривает повторное выполнение некоторого набора действий.

Циклы позволяют записать длинные последовательности операций обработки данных с помощью небольшого числа повторяющихся команд.



Итерационным называется цикл, число повторений которого не задается, а определяется в ходе выполнения цикла. В этом случае одно повторение цикла называется итерацией.

Рекурсия – это такая ситуация, когда некоторый алгоритм непосредственно или через другие алгоритмы вызывает себя в качестве вспомогательного. Сам алгоритм при этом называется рекурсивным.

В качестве примера использования рекурсии рассмотрим задачу поиска файлов. Пусть нужно получить список всех фай-лов, например, с расширением bmp, которые находятся в указанном пользователем каталоге и во всех подкаталогах этого каталога.

Словесно алгоритм обработки каталога может быть представлен так:

1. Вывести список всех файлов, удовлетворяющих критерию запроса.

2. Если в каталоге есть подкаталоги, то обработать каждый из этих каталогов.

Приведенный алгоритм, блок-схема которого представлена на рис. является рекурсивным. Для того чтобы обработать подкаталог, процедура обработки текущего каталога должна вызвать сама себя.

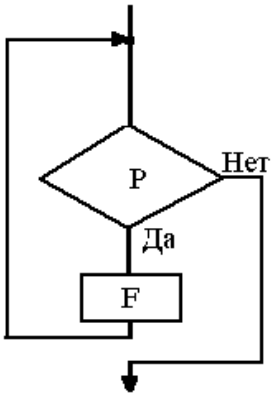
Различают цикл с **предусловием** и цикл с **постусловием.**

Цикл начинается с проверки логического выражения «P». Если оно истинно, то выполняется «F», затем все повторяется снова, до тех пор, пока логическое выражение сохраняет значение «истина». Как только оно становится ложным, выполнение операций «F» прекращается и управление передается по программе дальше.

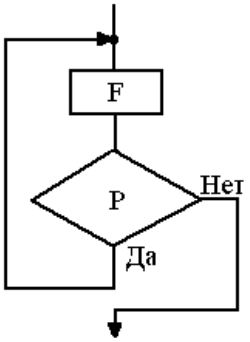
Так как выражение, управляющее циклом, проверяется в самом начале, то в случае, если условие сразу окажется ложным, операторы циклической части «F» могут вообще не выполняться. Операторы циклической части «F» должны изменять.

Переменную (или переменные), влияющую на значение логического выражения, иначе программа «зациклится» – будет выполняться бесконечно. Рассмотренная циклическая конструкция называется

цикл «**пока**», или цикл с **предусловием** (см. рис.)



Существует и иная конструкция цикла, которая предусматривает проверку условия после выполнения команд, встроенных внутрь цикла. Это цикл с **постусловием** (см. рис.)

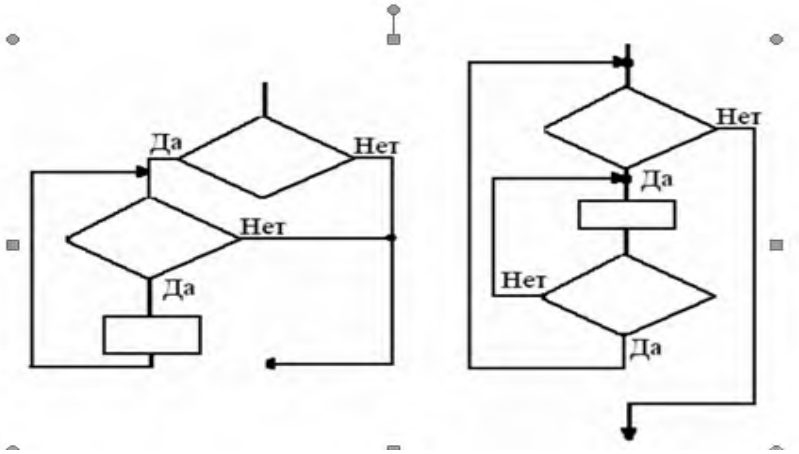


Циклические структуры можно комбинировать одну с другой – как путем организации их следований, так и путем создания суперпозиций (вложений одной структуры в другую).

Схематические изображения нескольких суперпозиций базовых алгоритмических структур представлены на рис..

Алгоритм типа «цикл, вложенный в неполную развилку»

Алгоритм типа «цикл в цикле»



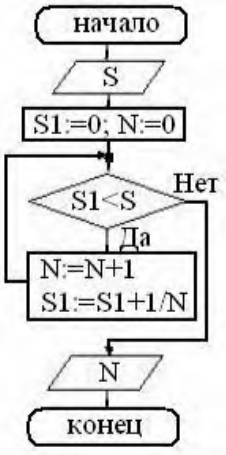
ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ НА АЛГОРИТМЫ

Цикл с предусловием

Пример 1. Дана блок-схема (рис. 14). Какое значение будет иметь N на выходе, если:

а) S=1,1; б) S=2,09?

Для определения результата воспользуемся трассировочными таблицами (а):



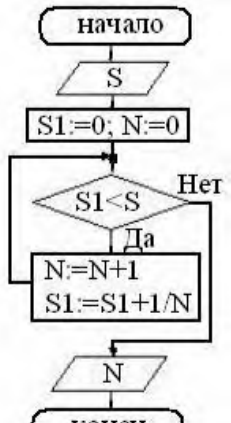
а) S=1,1:

Шаг	1	2	3
Значение S1	0	1	1,5
Значение N	0	1	2
Тело цикла	0 < 1,1 (Да)	1 < 1,1 (Да)	1,5 < 1,1 (Нет)
Результат выполнения	N=0+1=1 S1=0+1/1=1	N=1+1=2 S1=1+1/2=1,5	N = 2
Вывод значения N			2

Задание №1

Какое значение будет иметь N на выходе, если:

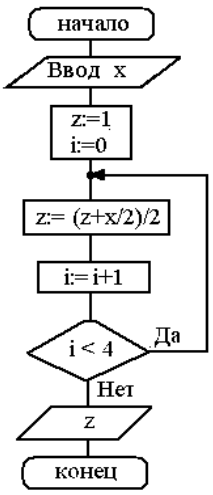
S=2,09. Для определения результата составьте трассировочную таблицу



Документ управляется программными средства
Проверь актуальность версии по оригиналу, храняще

Цикл с постусловием

Пример 2. Дана блок-схема (рис.) Какое значение будет иметь z на выходе, если: а) x=2



Для определения результата воспользуемся трассировочной таблицей

а) x=2:

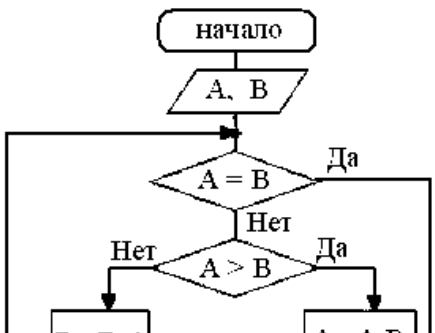
Шаг	1	2	3	4
Начальное значение z	1	1	1	1
Значение i	0	1	2	3
Результат выполнения z	z = (1+2/2)/2=1	z = (1+2/2)/2=1	z = (1+2/2)/2=1	z = (1+2/2)/2=1
Результат выполнения i	i = 0+1=1	i = 1+1=2	i = 2+1=3	i = 3+1=4
Тело цикла	1<4(Да)	2<4(Да)	3<4(Да)	4 < 4 (Нет)
Вывод z				1

Задание №2

Определите результат при x=4, x=6 , используя предыдущую блок-схему .Для определения результата воспользуйтесь трассировочной таблицей.

Задание №3

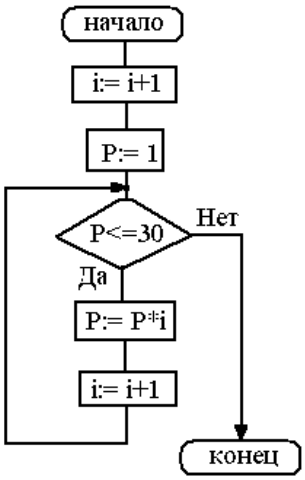
На блок-схеме (рис.)представлен алгоритм Евклида, определяющий наибольший общий делитель (НОД) для двух натуральных чисел A и B. Найти A на выходе блок-схемы, если: а) A=5, B=10; б) A=8, B=8



Документ управляется программн
Проверь актуальность версии по ориги

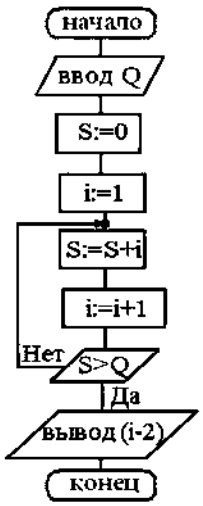
Задание №4

Дана блок-схема (рис. 26). Тогда после исполнения алгоритма переменная ***i*** примет значение …



Задание №5

Реализован некоторый алгоритм в виде блок-схемы (рис.) Что получится на выходе блок-схемы, если: Q=2;



Результат работы алгоритма определяется с помощью трассировочных таблиц

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия

2. Цель занятия
3. Вариант задания
4. Список используемых источников
5. Выводы и предложения
6. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что называется алгоритмом?
2. Для чего необходимы трассировочные таблицы?
3. Как представляется блок-схемой циклический алгоритм с пост условием?
4. Как представляется блок-схемой циклический алгоритм с пред условием?

Практическое занятие № 11 Введение в язык программирования Python. Ввод и вывод данных. Типы данных

Цель занятия:

1. Познакомиться со средой разработки Python;
2. Уметь запускать среду программирования Python, сохранять и открывать программы в среде программирования;
3. Знать основные типы данных, команды ввода и вывода данных;
4. Формировать ОК5, ОК6, ЛР26, ЛР30 .

Исходные материалы: теоретический материал, компьютер, среда программирования Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;
2. Выполнить задания.

Теоретическая часть

Python — активно развивающийся язык программирования, новые версии с добавлением и изменением языковых свойств выходят примерно раз в два с половиной года. Он находит применение во множестве сфер человеческой деятельности.

В комплекте вместе с интерпретатором Python идет IDLE (интегрированная среда разработки). По своей сути она подобна интерпретатору, запущенному в

интерактивном режиме с расширенным набором возможностей (подсветка синтаксиса, просмотр объектов, отладка и т.п.).

Скачать и установить интерпретатор Python можно совершенно бесплатно с официального сайта: http://python.org. Для работы нам понадобится интерпретатор **Python версии 3** или выше.

Для запуска **IDLE** в Windows необходимо перейти в папку **Python** в меню **“Пуск”** и найти там ярлык с именем “**IDLE** (Python 3.X XX-bit)”.

После установки программы запустите интерактивную графическую среду IDLE и дождитесь появления приглашения для ввода команд:

Type "copyright", "credits" or "license ()" for more information.

>>>

Три знака «больше» (>>>) называются *приглашением*. После приглашения можно вводить различные команды.

В интерактивном режиме IDLE можно с клавиатуры вводить математические выражения. После завершения набора выражения нажимается клавиша **Enter** для завершения ввода и вывода результата на экран. Такой режим работы Python называют интерактивным калькулятором.

Пример:

```
>>>1-5
- 4
>>> _+6   Нижним подчеркиванием обозначается последний полученный результат.
2
```

Любая Python-программа состоит из последовательности допустимых символов, записанных в определенном порядке и по определенным правилам.

Алфавит языка Python (набор допустимых символов) состоит из букв латинского алфавита (причём *заглавные и строчные буквы различаются*), цифр и специальных знаков (знаков препинания, арифметических и других). Русские буквы могут использоваться только при выводе текста на экран и в комментариях к программе.

Программа на языке Python включает в себя:

1)**комментарии**

обозначаются предваряющим их символом **#** *и продолжаются до конца строки*

2)**команды**

каждая команда пишется с новой строки или разделяется в одной строке **;**

3)**знаки пунктуации**

в алфавит Python входит достаточное количество знаков пунктуации, которые используются для различных целей. Например, знаки "+" или "" могут использоваться для сложения и умножения, а знак запятой "," - для разделения параметров функций*

4)**идентификаторы**

идентификаторы в Python - это имена используемые для обозначения переменной, функции, класса, модуля или другого объекта.

5)**ключевые слова**

некоторые слова имеют в Python специальное назначение и представляют собой управляющие конструкции языка, ключевые слова, которые для Python имеют определенный смысл.

Величины в программе представлены в виде констант и переменных.

Константы – величины, не изменяющие своего значения при выполнении программы.

Переменные – величины, которые могут изменять свое значение при выполнении программы. Каждая переменная имеет имя, тип и значение.

Имя переменной (идентификатор) – любая отличная от служебных слов последовательность латинских букв, цифр и символа подчеркивания "_", не может начинаться с цифры.

N, N1, massa, massa_tela – правильно;

1N, масса, massa tela – неправильно.

Имена переменным придумывает программист, но есть несколько ограничений, связанных с наименованием. В качестве имен переменных нельзя использовать ключевые слова, которые для Python имеют определенный смысл (эти слова подсвечиваются в IDLE оранжевым цветом):

and	as	assert	break	class	continue
def	del	elif	else	except	exec
finally	for	from	global	if	import
in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with
yield	True	False	None		

Значения переменных хранятся в ячейках оперативной памяти.

Тип переменной определяет способ хранения данных в памяти компьютера и допустимые операции над ними.

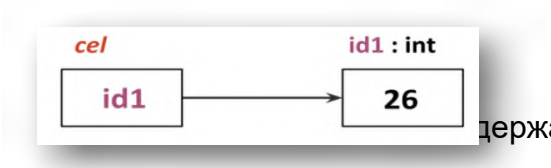
Типы данных

Название	Обозначение	Допустимые значения
Целочисленный	int («integer»)	Сколь угодно большие целые числа, размер ограничен оперативной памятью
Вещественный	float («floating point»)	Любые числа с дробной частью (с плавающей точкой)
Строковый	str («string»)	Произвольная последовательность символов из таблицы Unicode
Логический	bool («boolean»)	False («Ложь») или True («Истина»)

Команда присваивания

Знак **=** сопоставим в программировании с командой присваивания.

В момент выполнения присваивания




```
>>> cel = 26
>>>cel = 26 + 46
>>> cel
72
>>>
```

держат адреса объектов или можно сказать, что переменные ссылаются на объекты. Постоянно сохраняя в голове эту модель, для упрощения будем говорить, что переменная содержит значение.

В языке Python допускается множественное присваивание:

Запись оператора:	*Равносильная запись:*
`a, b = 0, 1`	`a = 0` `b = 1`
`a = b = 0`	`a = 0` `b = 0`

В математических выражениях в качестве операндов можно использовать целые числа (1, 4, -5) или вещественные (в программировании их еще называют числами с плавающей точкой): 4.111, -9.3. Если один из операндов является вещественным числом, то в результате получится вещественное число.

Простые арифметические операции над числами

оператор	описание
x + y	Сложение
x - y	Вычитание
x * y	Умножение
x / y	Деление
//	Деление с округлением вниз
**	Возведение в степень
%	Остаток от деления
abs(x)	Модуль числа

Пример:

- **>>> 5/3**
- **1.6666666666666667**
- **>>> 5//3**
- **1**
- **>>> 5%3**
- **2**
- **>>> 5**67**
- **677626357803440271254658000543713569641113281 25**
- **>>>**

Приоритеты выполнения операций

1)операции в скобках;

2) возведение в степень;

3) умножение и деление (в том числе // и %);

4) сложение и вычитание.

Операции одинакового приоритета выполняются в порядке записи слева направо.

- >>> -2**4
- -16
- >>> -(2**4)
- -16
- >>> (-2)**4
- 16

Если выражение слишком длинное и не помещается в одной строке, необходимо заключить всё выражение в скобки (перенос внутри скобок разрешён).

Ввод и вывод данных

Для ввода значений переменных с клавиатуры в процессе выполнения программы используется функция ввода

input («ввод»):

Формат записи функции

<имя_переменной> = **input**()

В скобках функции можно указать сообщение - комментарий к вводимым данным:

```
a = input ("Введите количество: ")
```

При выполнении оператора:

- компьютер переходит в режим ожидания данных;
- пользователь вводит с клавиатуры данные в виде строки символов;
- для завершения ввода пользователь нажимает клавишу Enter;

введенная строка записывается в указанную переменную.

Если вводится не строка, а число, необходимо выполнить преобразование типов с помощью функций **int** (для целых) и **float** (для вещественных).

Например:

```
print("Введите слово и два числа:")
x = input()
y = int(input())
z = float(input())
print(x, y, z)
```

На экране:

```
Введите слово и два числа:
ноль
1
2
ноль 1 2.0
```

Можно в одной строке ввести несколько значений через пробел. Для этого используется функция **split** («расщепить»). Затем данные необходимо преобразовать к нужному типу по отдельности.

Например:

```
a, b, c = input("Введите a,b,c через пробел: ").split()
a, b, c = int(a), int(b), int(c)
print (a, b, c)
```

На экране:

```
Введите a,b,c через пробел: 1 2 3
1 2 3
```

Вывод данных из оперативной памяти на экран осуществляется с помощью функции вывода

print («печатать»):

формат записи функции

print(<выражение1>, <выражение2>, ..., <выражениеN>)

- На экран выводятся значения переменных и выражений, строковые значения выводится на экран без кавычек.
- Выводимые значения разделяются пробелом (по умолчанию).
- После выполнения функции происходит автоматический переход на новую строку.

Например:

print ("Масса равна", m, "кг");

Для m=15 на экране появится:

Масса□равна□15□кг

Здесь и далее символом □ обозначен пробел.

Вместо пробела можно использовать другие символы в качестве разделителя, указав их после слова **sep** («separator»).

Чтобы убрать переход на новую строку после выполнения оператора, используется параметр **end**

Нужный вариант вывода	**Оператор**	**На экране**
По умолчанию	print (1, 20, 300)	1□20□300
Без разделителя	print (1, 20, 300, sep="")	120300
Через запятую и пробел	print (1, 20, 300, sep=", ")	1,□20,□300
Каждое с новой строки	print (1, 20, 300, sep="\n")	1 20 300
Без перехода на новую строку	print (1, end="") print (20)	120

Фрагмент программы	**На экране**
print ("{:3}{:3}{:3}".format(13, 7, 22))	□13□□7□22
a = 7 print ("{:4d}{:4d}".format(a, a*a))	□□□7□□49
a = 1/3; b = 1/9 print ("{:7.3f}{:7.4f}".format(a, b))	□□0.333□0.1111

a = 1/3 print ("{:10.3e}".format(a))	□3.333e-01

Пример

Напишите программу, которая запрашивала бы у пользователя:

Вариант 0 **Решение**

- ФИО ("Ваши фамилия, имя, отчество?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваше имя"

"Ваш возраст"

"Вы живете в"

```
a=input('Введите ваши фамилию, имя, отчество ')
b=input('Сколько вам лет? ')
c=input('Где вы живёте? ')
print('Ваше имя ',a)
print('Ваш возраст ',b)
print('Вы живете в ',c)
```

```
Введите ваши фамилию, имя, отчество Иванов Иван Иванович
Сколько вам лет? 15
Где вы живёте? Уссурийск
Ваше имя  Иванов Иван Иванович
Ваш возраст  15
Вы живете в  Уссурийск
```

Задания для самостоятельной работы (по вариантам)

Вариант 1

1) В режиме интерактивного калькулятора найти значение выражения

$$\left(5{,}1\cdot 10^{-3}\right)^2-\left(\frac{1}{7}\right)^2+\sqrt{5},$$

2) Напишите программу, которая запрашивала бы у пользователя:

Имя, Фамилия, Возраст, Место жительства

- фамилия, имя ("Ваши фамилия, имя?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваши фамилия, имя"

"Ваш возраст"

"Вы живете в"

Вариант 2

1) В режиме интерактивного калькулятора найти значение выражения

$$3{,}95+\left(\frac{1}{20}\right)^2+\frac{1}{900},$$

2) Напишите программу, которая запрашивала бы у пользователя:

Имя, Дата рождения, Образование

- имя ("Ваше, имя?")
- дата рождения ("Ваша дата рождения?")
- образование ("Где Вы учитесь?")

После этого выводила бы в одну строку через два пробела строки:

"Ваше имя" "Дата рождения" "Вы учитесь в "

Вариант 3

1) В режиме интерактивного калькулятора найти значение выражения

$$\frac{2{,}75\cdot 10^2}{\sqrt{14{,}3+\pi}},$$

2) Напишите программу, которая запрашивала бы у пользователя:

Фамилия, Место жительства

- Фамилия("Ваша фамилия?")
- место жительства ("Где Вы живете?")

После этого выводила бы две строки: "Ваша фамилия" и "Вы живете в"

Вариант 4

1) В режиме интерактивного калькулятора найти значение выражения

$$6{,}7\left(\frac{1}{\pi+5{,}04}+10^{0{,}198\sqrt{3{,}2}}\right)-0{,}36\cdot 10^{-3},$$

2)Напишите программу, которая запрашивала бы у пользователя:

Фамилия, Место рождения, любимая музыка

- Фамилия, ("Ваша фамилия?")
- место рождения ("Где Вы родились?")
- музыка("Какая музыка нравится? ")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы родились в"

"Ваша любимая музыка "

Вариант 5

1) В режиме интерактивного калькулятора найти значение выражения

$$\frac{10^{\frac{1}{3}} - \left(4{,}25 + \frac{1}{7}\right) + 0{,}89 \cdot 10^{-4} \cdot 1{,}2}{6},$$

2)Напишите программу, которая запрашивала бы у пользователя:

Имя, Фамилия, ФИО мамы, ФИО отца

- ФИО (например, "Ваши фамилия, имя, отчество?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где Вы живете?")

После этого выводила бы в одну строку через запятую : "Ваши имя, фамилия, отчество","Ваш возраст","Вы живете в"

Вариант 6

1) В режиме интерактивного калькулятора найти значение выражения

$$(3 \cdot (17{,}8^2 - 0{,}139 \cdot 10^2)) + \frac{2{,}3 \cdot 0{,}69}{1{,}5},$$

2)Напишите программу, которая запрашивала бы у пользователя:

Имя, Любимый предмет в школе, Номер класса

- имя ("Ваше имя?")
- любимый предмет ("Какой Ваш любимый предмет в школе?")
- номер класса ("В каком классе Вы учитесь?")

После этого выводила бы три строки:

"Ваше имя", "Ваш любимый предмет в школе","Вы учитесь в классе номер"

Вариант 7

1) В режиме интерактивного калькулятора найти значение выражения

$$2^{\left((0{,}3410^3 - \frac{0{,}16 \cdot 5{,}27}{0{,}029}\right)},$$

2)Напишите программу, которая запрашивала бы у пользователя:

Имя, Фамилия, Отчество, Хобби

- ФИО (например, "Ваши фамилия, имя, отчество?")

- хобби ("Чем Вы увлекаетесь?")

После этого выводила бы две строки:

"Ваши имя, фамилия, отчество"

"Ваше хобби"

Вариант 9

1) В режиме интерактивного калькулятора найти значение выражения

$$0{,}09 \quad (0{,}41 \cdot 10^{-3} + \sqrt{0{,}26 \cdot 10^{-5}}) + \sqrt{5{,}68 - \sqrt{3}} \qquad)$$

2)Напишите программу, которая запрашивала бы у пользователя:

Имя, Фамилия, любимый спорт

- Фамилия, имя ("Ваши фамилия, имя?")

- образование ("В какой школе Вы учитесь?")

- ФИО Вашего руководителя по информатики ("ФИО Вашего руководителя по информатики?")

После этого выводила бы в одну строку через пробел: "Ваши имя, фамилия"

"Вы учитесь в школе номер: " "ФИО Вашего руководителя по информатике "

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия;

2. Цель занятия;

3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

4. Список используемых источников;

5. Выводы и предложения;

6. Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1.Как в языке Python называются указания компьютеру, определяющие, какие операции выполнит компьютер над данными?

2.Определите порядок выполнения операций в указанной инструкции? `a = 3 - 5 * 4 ** (-3 + 2)`

3. Символ # в Python обозначает **…**

4.345, 6.8, КМРК - **…** типы данных.

5.Функция input() – предназначена для …

Практическое занятие № 12 Оператор присваивания. Математические операции с целыми и вещественными Стандартные функции. Математический модуль math. Составление программ профессиональной направленности.

Цель занятия:

1. Изучить основные математические операциями в Python;
2. Уметь писать коды программ для решения математических задач;
3. Изучить стандартные функции в Python;
4. Познакомиться с функциями математического модуля math;
5. Уметь писать коды программ для расчета математических функций;

Исходные материалы: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;
2. Выполнить задания.

Теоретическая часть

Язык Python, благодаря наличию огромного количества библиотек для решения разного рода вычислительных задач, сегодня является конкурентом таким пакетам как Matlab и Octave. Запущенный в интерактивном режиме, он, фактически, превращается в мощный калькулятор.

Задания для самостоятельной работы

Задание №1

Рассчитайте на калькуляторе Python

$3{,}95 + \left(\frac{1}{20}\right)^2 + \frac{1}{900}$, *Ответ*: 3,95.....

$\frac{2{,}75 \cdot 10^2}{\sqrt{14{,}3 + \pi}}$, *Ответ*: 65,842...

$$6{,}7\left(\frac{1}{\pi+5{,}04}+10^{0{,}198\sqrt{3{,}2}}\right)-0{,}36\cdot10^{-3}, \qquad \textit{Ответ}: \quad 15{,}96$$

$$\left(\sqrt{\frac{0{,}079}{3{,}5}-6{,}1\cdot10^{-5}}+0{,}4\right)\cdot\left(e^{\frac{0{,}7}{9}+0{,}8}-1{,}5\right), \qquad \textit{Ответ}: \quad 0{,}49\ldots$$

Ответы даны для того, чтобы вы убедились в правильности ваших действий в системе программирования.

В этом практическом занятии речь пойдет об арифметических операциях, доступных в данном языке. Арифметические операции изучим применительно к числам.

Если в качестве операндов некоторого арифметического выражения используются только целые числа, то результат тоже будет целое число. Исключением является операция деления, результатом которой является вещественное число. При совместном использовании целочисленных и вещественных переменных, результат будет вещественным.

Целые числа (int)

Числа в Python 3 поддерживают набор самых обычных математических операций:

x + y	Сложение
x - y	Вычитание
x * y	Умножение
x / y	Деление
x // y	Получение целой части от деления
x % y	Остаток от деления
-x	Смена знака числа
abs(x)	Модуль числа
divmod(x, y)	Пара (x // y, x % y)

x ** y	Возведение в степень
pow(x, y[, z])	x : Число, которое требуется возвести в степень. y : Число, являющееся степенью, в которую нужно возвести первый аргумент. Если число отрицательное или одно из чисел "x" или "y" не целые, то аргумент "z" не принимается. z : Число, на которое требуется произвести деление по модулю. Если число указано, ожидается, что "x" и "y" положительны и имеют тип int.

Пример применения выше описанных операций над целыми числами

```
x = 5
y = 2
z = 3
x+y = 7
x-y = 3
x*y = 10
x/y = 2.5
x//y = 2
x%y = 1
-x= -5
abs(-x) = 5
divmod(x,y) = (2, 1)
x**y = 25
pow(x,y,z) = 1
```

Вещественные числа (float)

Вещественные числа поддерживают те же операции, что и целые. Однако (из-за представления чисел в компьютере) вещественные числа неточны, и это может привести к ошибкам.

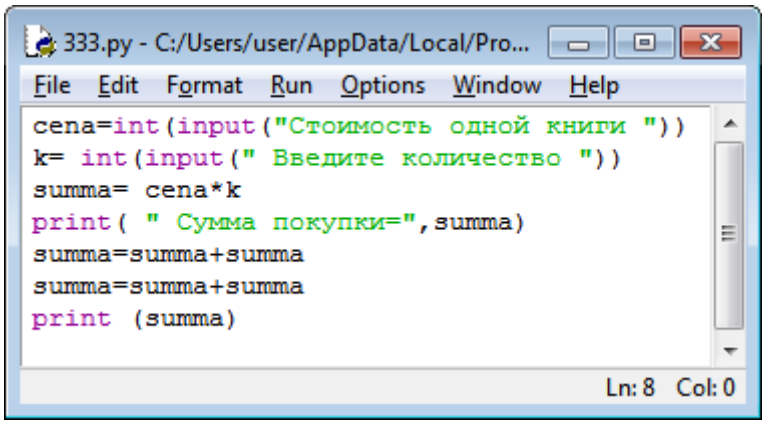
Пример применения вышеописанных операций над вещественными числами

```
x = 5.5
y = 2.3
x+y = 7.8
x-y = 3.2
x*y = 12.649999999999999
x/y = 2.3913043478260874
x//y = 2.0
x%y = 0.9000000000000004
-x= -5.5
abs(-x) = 5.5
divmod(x,y) = (2.0, 0.9000000000000004)
x**y = 50.44686540422945
```

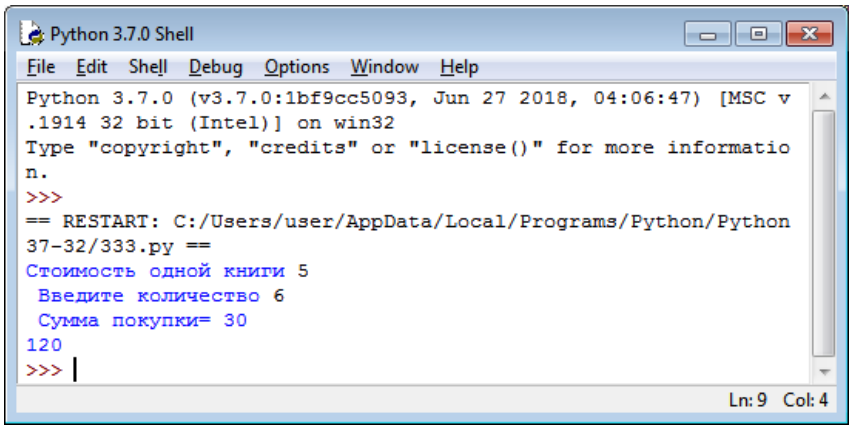
Пример составления программ для решения математической задачи

Задается стоимость одной книги и количество покупаемых книг. Написать программу вычисления стоимости покупки с выводом на экран.

Дописать в программу операторы присваивания, которые позволили бы просуммировать несколько значений стоимости покупки



Решение



Задания для самостоятельной работы

Задание №2

1. Составить программу, меняющую местами значения двух переменных
2. Составить программу для вычисления площади треугольника по известным длинам его сторон.

Формула Герона: $S = \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

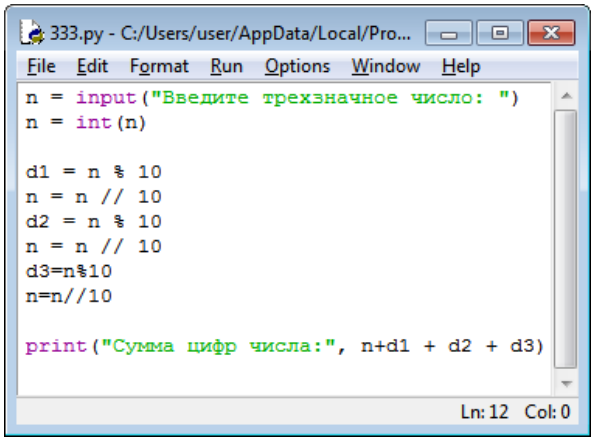
3. Задается значение скорости и времени. Напишите программу, которая запрашивала скорость, время и рассчитывала бы пройденный путь с выводом на экран.

4. Допишите в предыдущую программу операторы присваивания, которые позволяли бы просуммировать несколько значений пройденного пути.

5. Задается значение стоимости одного билета в кинотеатр и количество покупаемых билетов. Написать программу вычисления суммы к оплате.

6. Задается количество введенных оценок по различным предметам одного обучающего, написать программу нахождения среднего балла этого обучающего.

7. * Составить программу, вычисляющую сумму цифр введенного с клавиатуры целого трёхзначного числа



Большинство стандартных функций языка Python разбиты на группы по назначению, каждая группа записана в отдельном файле, который называется модулем.

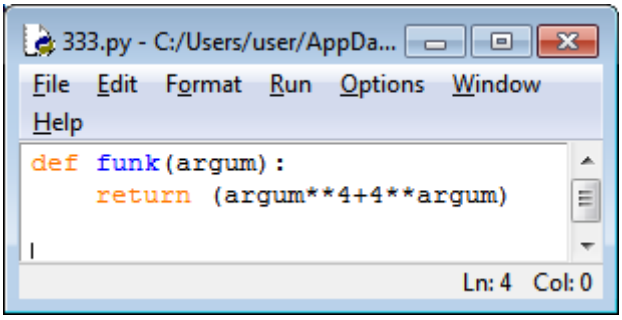
К примеру, вы написали несколько полезных функций, которые часто используете в своих программах. Чтобы к ним быстро обращаться, удобно все эти функции поместить в отдельный файл и загружать их оттуда.

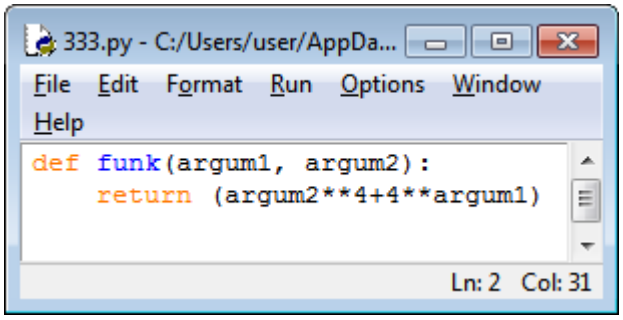
Пример

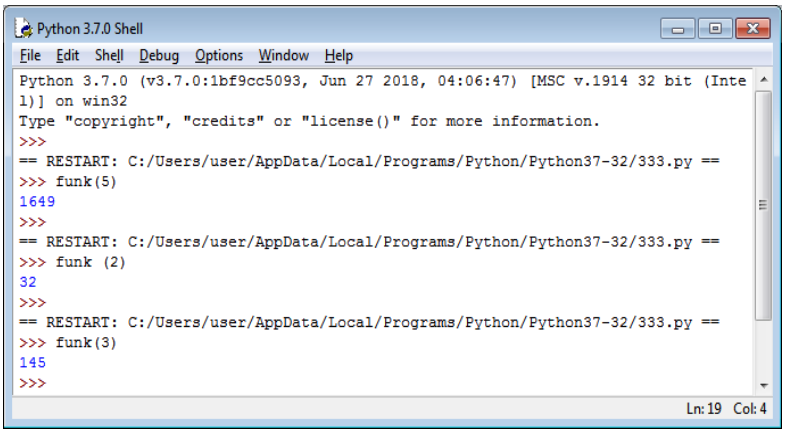
1. Создайте собственные функции для вычисления следующих выражений

$x^4 + 4^x$ $y^4 + 4^x$

Решение





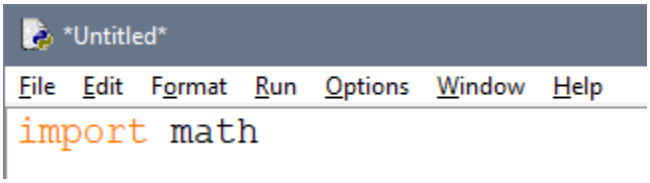


В Python такие файлы с набором функций и называются модулями. Для того чтобы воспользоваться функциями, которые находятся в этом модуле, его необходимо импортировать с помощью команды import:

Библиотека (модуль) math

В стандартную поставку Python входит библиотека math, в которой содержится большое количество часто используемых математических функций.

Для работы с данным модулем его предварительно нужно импортировать.




```
        # подключаем все функции из модуля math
                from math import *
```

```
from math import *

>>> math.sqrt(9)
 3.0

>>>
```

Рассмотрим наиболее часто используемые функции модуля math

math.ceil(x)	Возвращает ближайшее целое число большее, чем x
math.fabs(x)	Возвращает абсолютное значение числа x
math.factorial(x)	Вычисляет факториал x
math.floor(x)	Возвращает ближайшее целое число меньшее, чем x
math.exp(x)	Вычисляет e**x
math.log2(x)	Логарифм по основанию 2
math.log10(x)	Логарифм по основанию 10
math.log(x[, base])	По умолчанию вычисляет логарифм по основанию e, дополнительно можно указать основание логарифма
math.pow(x, y)	Вычисляет значение x в степени y
math.sqrt(x)	Корень квадратный от x

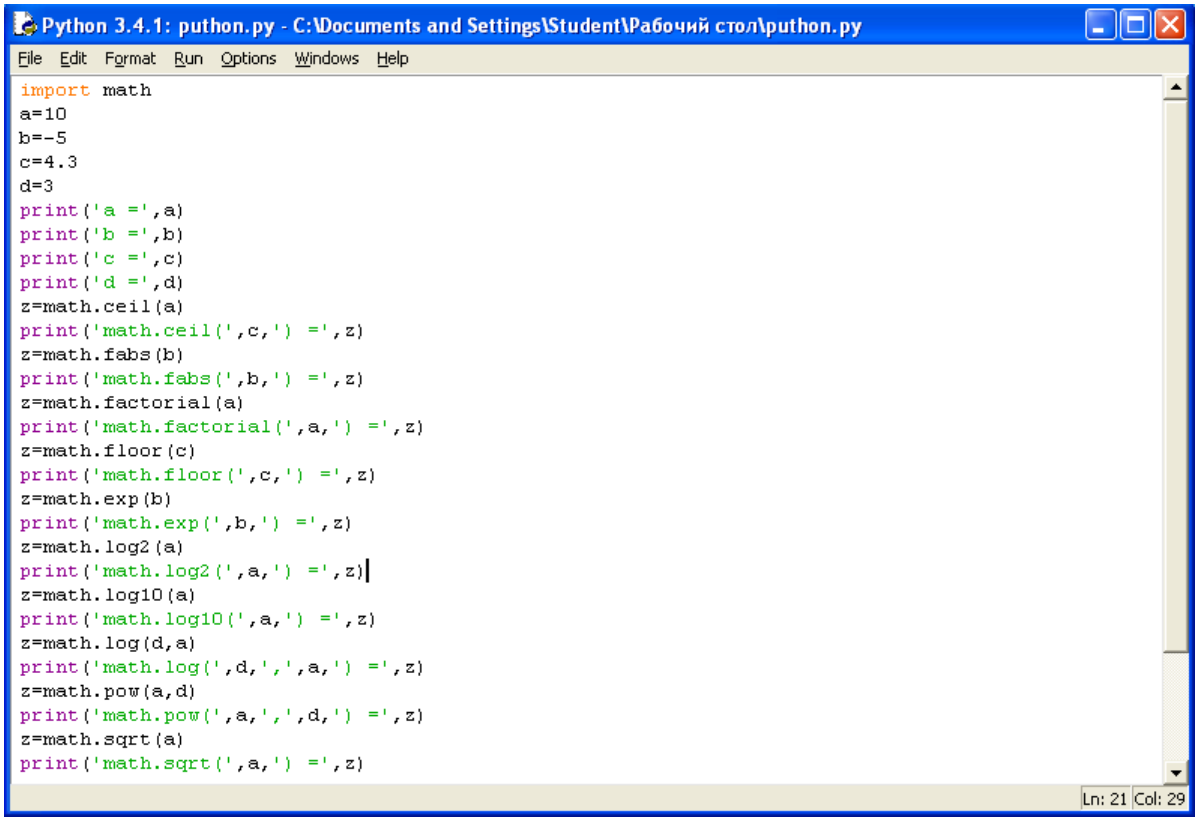
Пример применения вышеописанных функций над числами

В программе определены 4 переменные - a, b, c, d, каждая из которых является либо целым числом, либо вещественным, либо отрицательным.

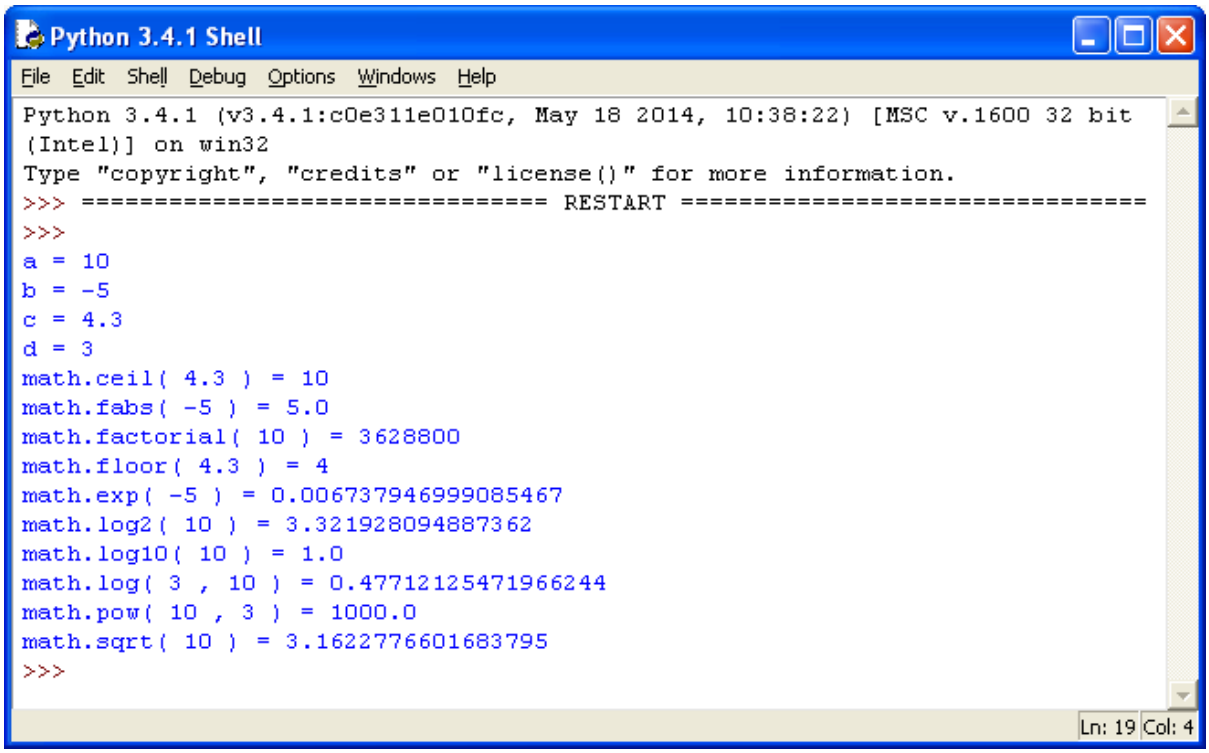
Командой print() выводится значение каждой переменной на экран при выполнении программы.

В переменную z помещается результат выполнения функции модуля math.

Затем командой print() выводится сообщение в виде используемой функции и её аргумента и результат её выполнения.



Пример программы на Python



Результат выполнения программы с применением функций модуля math

Тригонометрические функции модуля math

math.cos(x)	Возвращает cos числа X
math.sin(x)	Возвращает sin числа X
math.tan(x)	Возвращает tan числа X
math.acos(x)	Возвращает acos числа X
math.asin(x)	Возвращает asin числа X
math.atan(x)	Возвращает atan числа X

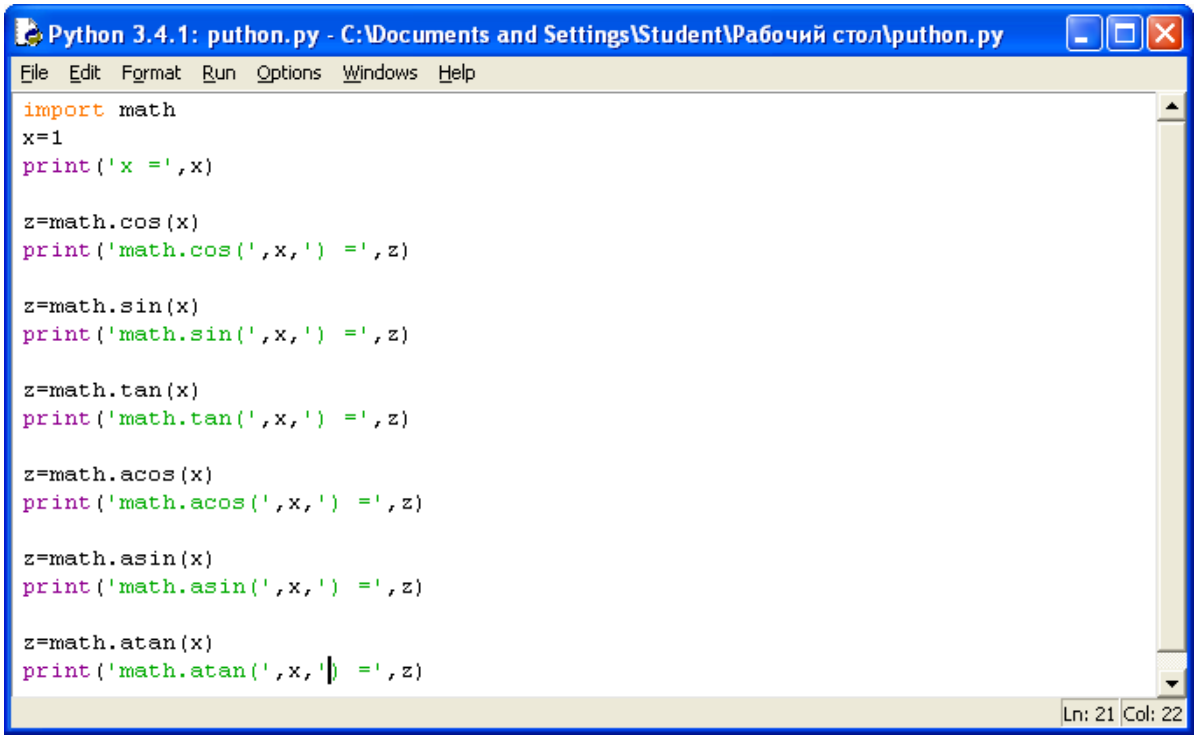
Пример применения вышеописанных функций над числами

В программе определена переменная x, содержащая целое число.

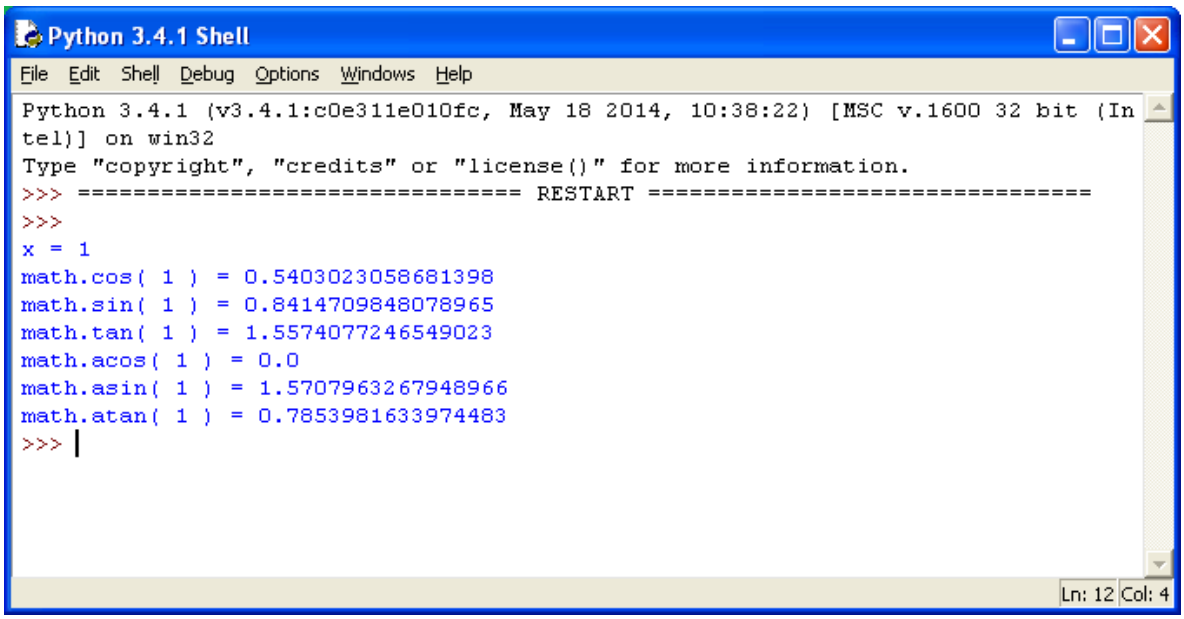
Значение переменной выводится командой print() на экран.

В переменную z помещается результат выполнения тригонометрической функции модуля math.

Затем командой print() выводится сообщение в виде используемой функции и её аргумента и результат её выполнения.



Пример программы с использованием тригонометрических функций модуля math



Результат выполнения программы с применением тригонометрических функций модуля math

Константы:

- **math.pi** - число Pi.
- **math.e** - число e (экспонента).

Пример

Напишите программу, которая бы вычисляла заданное арифметическое выражение при заданных переменных. Ввод переменных осуществляется с клавиатуры. Вывести результат с 2-мя знаками после запятой.

Вариант 0

$$Z = \frac{9\pi t + 10\cos(x)}{\sqrt{t} - |\sin(t)|} * e^{x}$$

x=10; t=1

Решение

Сначала импортируем модуль math. Для этого воспользуемся командой importmath.

Затем следует ввести значения двух переменных целого типа x и t.

Для ввода данных используется команда input, но так как в условии даны целые числа, то нужно сначала определить тип переменных: x=int(), t=int().

Определив тип переменных, следует их ввести, для этого в скобках команды int() нужно написать команду input().

Для переменной x это выглядит так: x=int(input("сообщение при вводе значения")).

Для переменной t аналогично: t=int(input("сообщение при вводе значения")).

Следующий шаг - это составление арифметического выражения, результат которого поместим в переменную z.

Сначала составим числитель. Выглядеть он будет так: 9*math.pi*t+10*math.cos(x).

Затем нужно составить знаменатель, при этом обратим внимание на то, что числитель делится на знаменатель, поэтому и числитель, и знаменатель нужно поместить в скобки (), а между ними написать знак деления /.

Выглядеть это будет так: (9*math.pi*t+10*math.cos(x))/(math.sqrt(t)-math.fabs(math.sin(t))).

Последним шагом является умножение дроби на экспоненту в степени x.

Так как умножается вся дробь, то следует составленное выражение поместить в скобки (), а уже потом написать функцию math.pow(math.e,x).

В результате выражение будет иметь вид:

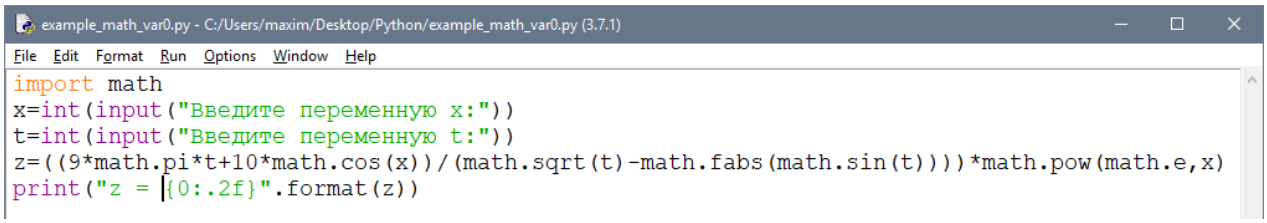
z=((9*math.pi*t+10*math.cos(x))/(math.sqrt(t)-math.fabs(math.sin(t))))*math.pow(math.e,x).

При составлении данного выражения следует обратить внимание на количество открывающихся и закрывающихся скобок.

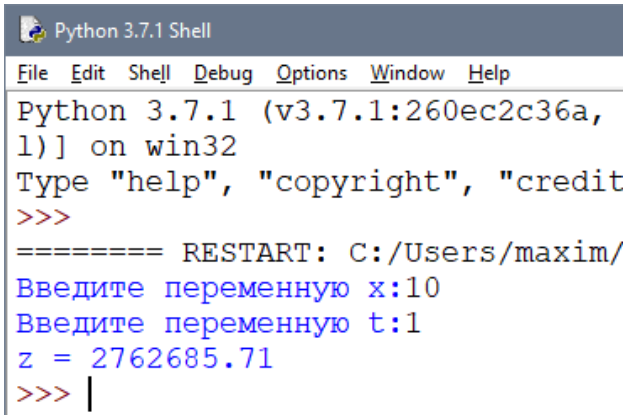
Командой print() выведем значение переменной, отформатировав его командой format.

Сам формат записывается в апострофах в фигурных скобках {}.

В задаче требуется вывести число с двумя знаками после запятой, значит вид формата будет выглядеть следующим образом: {0:.2f}, где 2 - это количество знаков после запятой, а f указывает на то, что форматируется вещественное число. При этом перед 2 нужно поставить точку, указав тем самым на то, что форматируем именно дробную часть числа.



Результат



ВАРИАНТЫ ЗАДАНИЙ

Вариант 1.

Вычислить. $a=\dfrac{\sqrt{|x-1|}-\sqrt[3]{|y|}}{1+\dfrac{x^2}{2}+\dfrac{z^2}{4}}$, и $b=x(arctgZ+e)^{-(y+3)}$; если *x = 2; y = 6; z = 7.*

Вариант 2.

Вычислить $a=\dfrac{3+e^{y-1}}{1+x(y-tgZ)}$, и $b=1+|y-x|+\cdot\dfrac{(y-x)^2}{2}+\dfrac{|y-z|^3}{3}$;

если *x = 5; y = 1; z = 4.*

Вариант 3.

Вычислить $a=(1+y)\dfrac{x+y/(x^2+4)}{e^{-x-2}+1/(z^2+4)}$, и $b=\dfrac{1+\cos(y-2)}{x^{4/2}+\sin^2 z}$; если *x =1; y=12; z = 6.*

Вариант 4.

Вычислить $a=y+\dfrac{x}{z^2+\left|\dfrac{x^2}{y+x^3/3}\right|}$, и $b=(1+\arcsin\ x^2+tg^2\dfrac{z}{2})$;

если *x = 11; y = 5; z = 10.*

Вариант 5.

Вычислить $a=\dfrac{2\cos(x-n/6)}{1/2+\sin^2 y}$, и $b=1+\dfrac{z^2}{3+z^2/5}$; если *x = 15; y = 7; z = 3.*

Вариант 6.

Вычислить $a = \frac{1+\sin^2(x-y)}{2+\left|x-2x/(1+x^2y^2)\right|}+z,$ и $b=\cos^2(arctg\frac{1}{z});$ если

x = 3; y = 4; z = 5.

Вариант 7.

Вычислить $a=\ln\left|(y-\sqrt{|x|})(x-\frac{y}{z+x^2/4})\right|,$ и $b=x-\frac{x^2}{3!}+\frac{x^5}{5};$

если *x =5; y =3; z =17.*

Вариант 8.

Вычислить $a=\left|\frac{\sin^3\left|3x^3+5y^2+15z\right|}{\sqrt{(12x^3+6y^2-z)^2+3.14}}\right|,$ и $b=tg(7x^2+e^{3y^2-z});$ если

x = 7; y = 4; z = 8.

Вариант 9.

Вычислить $a=\sqrt{\frac{|\ \sin 8y\ |+17x}{(1-\sin 4\cdot y\cdot\cos(z^2+18))^2}},$ и $b=y-\sqrt{\frac{3|z|^2}{3+\left|tg3x^2-\sin 5y\right|}};$ если

x = 17; y = 16; z = 15.

Вариант 10.

Вычислить $a=l^{(2x^2-y)}+\frac{3\arccos y^2}{(z^2+xy)^2},$ и $b=tg\left(\frac{|x-y|^2}{3Z+\cos x^2}\right);$ если

x = 6; y = 7; z = 8.

Вариант 11.

Вычислить $a=10\cdot\ln\left|(x-\sqrt{\cos y})(z-\frac{y^2}{x-x^2/2})\right|;$ и $b=2x^3+\frac{\left|y^2-3z^3\right|}{e^{3x+y^2}};$ есл

и *x = 3; y = 4; z = 5.*

Вариант 12.

Вычислить $a=\frac{\cos^2 x+5\sin^3 y}{\ln\left|2z+z^3\right|};$ и $b=\arcsin\left(\frac{4x^2+5y^3}{\sqrt{2z-y^2}}\right)/\ln(\sqrt{7x-7x/5y^2});$ ес

ли *x = 4; y = 5; z = 6.*

Вариант 13.

Вычислить $a = tg\left((x+y)^2 - \sqrt{\frac{\cos^2(z)}{tgz^2}}\right)$, и $b = \ln(x^2 + \frac{e^{-3y^2}}{\sin^2 y})$; если

x = 5; y = 6; z = 7.

Вариант 14.

Вычислить $a = \left(\frac{y}{\sin x} + \frac{y}{(\sin^2 x - 3\cos z)}\right) \cdot e^{5x^2}$, и $b = \frac{5 - e^{z-2}}{y + x^2\left|z^2 - tgz\right|}$;

если *x = 8; y = 9; z = 10.*

Вариант 15.

Вычислить $a = z + \left|y^2 - x^2\right| + y(\arcsin(z - e^{(z+5)}))$; и

$b = \sqrt{\frac{x - \cos^2(y+z)}{5 + \ln(y - 5x/\left|\ xy - \sqrt{7}\ \right|)}}$; если *x = 9; y = 10; z = 11*

Выводы и предложения о проделанной работе

Содержание отчета:

1 Наименование практического занятия;

2 Цель занятия;

3 Вариант задания;

4 Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

5 Список используемых источников;

6 Выводы и предложения;

7 Дата и подпись курсанта и преподавателя;

Вопросы для самопроверки

1 Перечислите математические операции для целых чисел.

2 Перечислите часто используемые функции модуля math.

3 Перечислите тригонометрические функции модуля math.

4 Как создать свою собственную функцию?

Практическое занятие № 13 Проверка условий в Python. Синтаксис If,If-else,if-elif-else. Составление программ с проверкой условий из профессиональной деятельности.

Цель занятия:

1. Познакомиться со структурой ветвление (if, if-else, if-elif-else);

2. Научиться работать с числами и строками используя структуру ветвление;

3. Уметь писать коды программ с использованием структуры ветвления;

4. Формировать ОК5, ОК6, ЛР26, ЛР30 .

Исходные данные: теоретический материал, программа Python

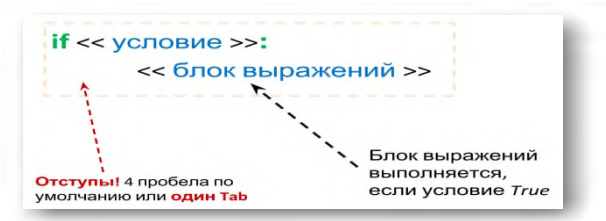
Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

Теоретическая часть

Наиболее часто логические выражения используются внутри условной инструкции **if:**



Условный оператор ветвления if, if-else, if-elif-else

Оператор ветвления if позволяет выполнить определенный набор инструкций в зависимости от некоторого условия. Возможны следующие варианты использования.

1. Конструкция if

Синтаксис оператора if выглядит так:

if логическое выражение:

команда_1

команда_2

...

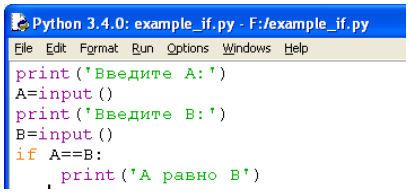
команда_n

После оператора if записывается **логическое выражение**.

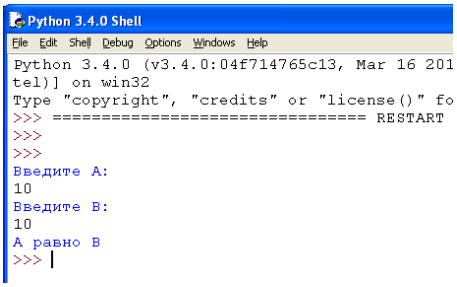
Напомним, **Логическое выражение** — конструкция языка программирования, результатом вычисления которой является «истина» или «ложь».

Если это выражение истинно, то выполняются инструкции, определяемые данным оператором. После условного выражения нужно поставить двоеточие "**:**".

Программа запрашивает у пользователя два числа, затем сравнивает их и если числа равны, то есть логическое выражение A==B истинно, то выводится соответствующее сообщение.



Пример программы на Python



Результат выполнения программы с использованием условного оператора if

2. Конструкция if – else

Бывают случаи, когда необходимо предусмотреть альтернативный вариант выполнения программы. Т.е. при истинном условии нужно выполнить один набор инструкций, при ложном – другой. Для этого используется конструкция if – else.

Синтаксис оператора if – else выглядит так:

if логическое выражение:

команда_1

команда_2

...

команда_n

else:

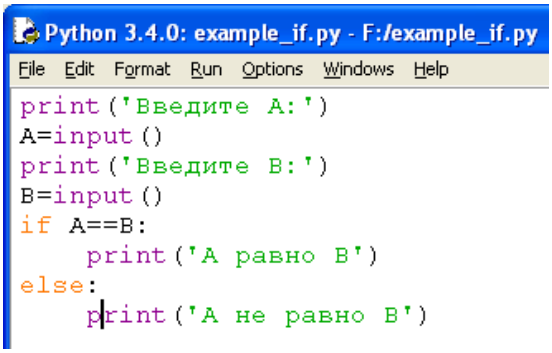
команда_1

команда_2

...

команда_n

Программа запрашивает у пользователя два числа, затем сравнивает их и если числа равны, то есть логическое выражение A==B истинно, то выводится соответствующее сообщение. В противном случае выводится сообщение, что числа не равны.



Пример программы на Python

```
>>> ================================ RESTART
>>>
Введите A:
10
Введите B:
5
A не равно B
>>>
```

Результат выполнения программы с использованием условного оператора if-else

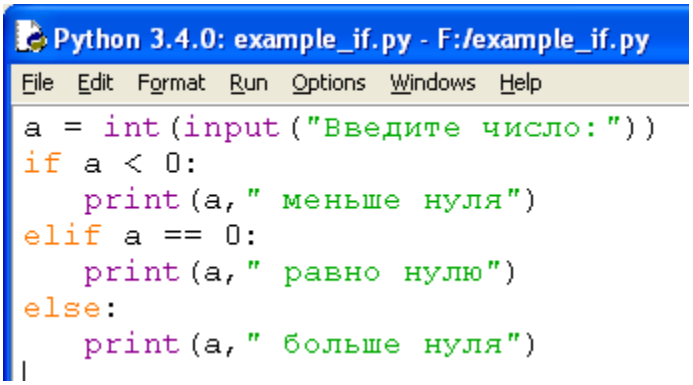
3. Конструкция if – elif – else

Для реализации выбора из нескольких альтернатив можно использовать конструкцию if – elif – else.

Синтаксис оператора if – elif – else выглядит так:

```
if  логическое выражение_1:
    команда_1
    команда_2
    ...
    команда_n
elif логическое выражение_2:
    команда_1
    команда_2
    ...
    команда_n
elif  логическое выражение_3:
    команда_1
    команда_2
    ...
    команда_n
else:
    команда_1
    команда_2
    ...
    команда_n
```

Программа запрашивает число у пользователя и сравнивает его с нулём a<0. Если оно меньше нуля, то выводится сообщение об этом. Если первое логическое выражение не истинно, то программа переходит ко второму - a==0. Если оно истинно, то программа выведет сообщение, что число равно нулю, в противном случае, если оба вышеуказанных логических выражения оказались ложными, то программа выведет сообщение, что введённое число больше нуля.



Пример программы на Python

```
Введите число:41
41  больше нуля
>>> ================================ RESTART =
>>>
Введите число:-5
-5  меньше нуля
>>> ================================ RESTART =
>>>
Введите число:0
0  равно нулю
>>>
```

Результат выполнения программы с использованием условного оператора if-elif-else

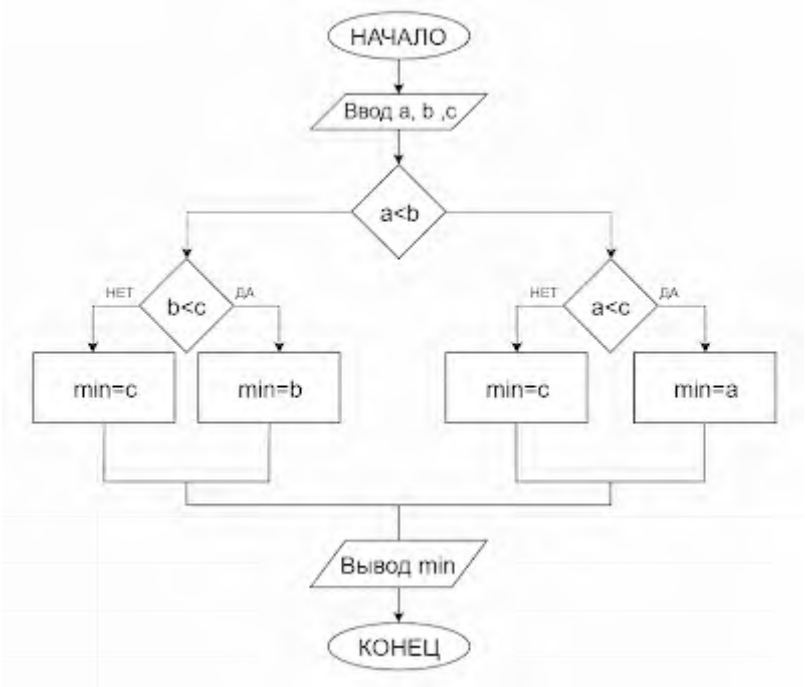
Пример

Вариант 0

Дано 3 числа. Найти минимальное среди них и вывести на экран.

Решение

Для простоты построим блок-схему задачи.



Командами

a=input('')

b=input('')

c=input('')

введём три числа, присвоив значения переменным a, b, c.

Условной конструкцией if-else проверим на истинность логическое выражение a<b. Если оно истинно, то переходим на проверку логического выражения a<c. Если оно истинно, то переменной "y" присвоим значение переменной "a", т.е. "a" будет минимальным, а иначе "y" присвоится значение переменной "c".

Если в начале логическое выражение a<b оказалось ложным, то переходим на проверку другого логического выражения b<c.

Если оно истинно, то "y" присвоится значение переменной "b", иначе "c".

Командой print() выводим минимальное значение.

Пример программы:

```
#нахождение минимального из 3-х чисел
a=input('Введите целое число \n')
b=input('Введите целое число \n')
c=input('Введите целое число \n')
if a<b:
    if a<c:
        y=a
    else:
        y=c
else:
    if b<c:
        y=b
    else:
        y=c
print('Минимальное:',y)
```

Результат выполнения программы:

```
Введите целое число
2
Введите целое число
5
Введите целое число
1
Минимальное: 1
```

Задания для самостоятельной работы (по вариантам)

Вариант 1

Даны три целых числа. Выбрать из них те, которые принадлежат интервалу [1,3].

Вариант 2

Дан номер года (положительное целое число). Определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).

Вариант 3

Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется в том случае, если сумма покупки больше 500 руб., в 5% - если сумма больше 1000 руб.

Вариант 4

Написать программу, которая бы по введенному номеру единицы измерения (1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер) и массе M выдавала соответствующее значение массы в килограммах.

Вариант 5

Найти косинус минимального из 4 заданных чисел.

Вариант 6

Вывести на экран синус максимального из 3 заданных чисел.

Вариант 7

Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, т. е. имеют ли они равные площади. Если это не так, то вывести «Foul!!!»

Вариант 8

Составьте программу подсчёта площади равнобедренного треугольника. Если площадь треугольника чётная, разделить её на 2, в противном случае вывести сообщение «Не могу делить на 2!»

Вариант 9

Составить программу, которая по данному числу (1-12) выводит название соответствующего ему месяца на английском языке.

Вариант 10

Составить программу, осуществляющую перевод величин из радианной меры в градусную или наоборот. Программа должна запрашивать, какой перевод нужно осуществить, и выполнять указанное действие.

Вариант 11

Дано три числа. Найти количество положительных чисел среди них;

Вариант 12

Если действительные числа x и y – одного знака, найти их среднее геометрическое, в противном случае найти их среднее арифметическое.

Вариант 13

Определить, существует ли прямоугольный треугольник со сторонами x,y,z. Если – да, вычислить его площадь.

Вариант 14

Определить, существует ли треугольник с длинами сторон a, b, c. Если – да, вычислить его площадь по формуле Герона.

Формула Герона имеет вид:

S=p(p-a)(p-b)(p-c), где p=12(a+b+c)

Вариант 15

Вычислить значение функции f(x), если

$$f(x)=\begin{cases} 0{,}5-\sqrt[4]{|x|}, & x\geq 0 \\ \dfrac{\sin^2 x^2}{|x+1}, & x<0 \end{cases}$$

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия;
2. Цель занятия;
3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;
4. Список используемых источников;
5. Выводы и предложения;
6. Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1. Какой вычислительный процесс называется разветвляющимся?
2. Опишите синтаксис оператора if.
3. Дайте определение понятию «логическое выражение»
4. Опишите конструкцию if-elif-else.
5. Что позволяет выполнить оператор ветвления if.

Практическое занятие № 14 Реализация циклических алгоритмов в Python. Синтаксис цикла с предусловием и постусловием. Составление программ профессиональной направленности.

*Цель занятия***:**

1. Познакомиться с циклическими конструкциями;
2. Уметь писать коды программ, содержащие циклические конструкции с while;
3. Формировать ОК5, ОК6, ЛР26, ЛР30 .

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

Теоретическая часть

В Python существуют два типа цикличных выражений:

- Цикл while с предусловием и можно организовать while с постусловием
- Цикл for – (изучаем на следующем практическом занятии).

1. Цикл while в Python *Цикл с предусловием (с заданным условием продолжения работы, цикл «ПОКА»)*

Инструкция while в Python повторяет указанный блок кода до тех пор, пока указанное в цикле логическое выражение будет оставаться истинным.

Синтаксис цикла while:

while логическое выражение:

команда 1

команда 2

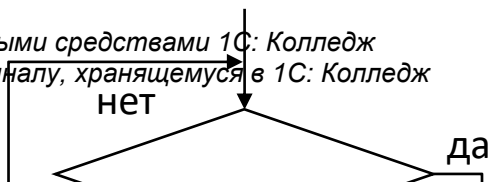
...

команда n

После ключевого слова while указывается условное выражение, и пока это выражение возвращает значение True, будет выполняться блок инструкций, который идет далее.

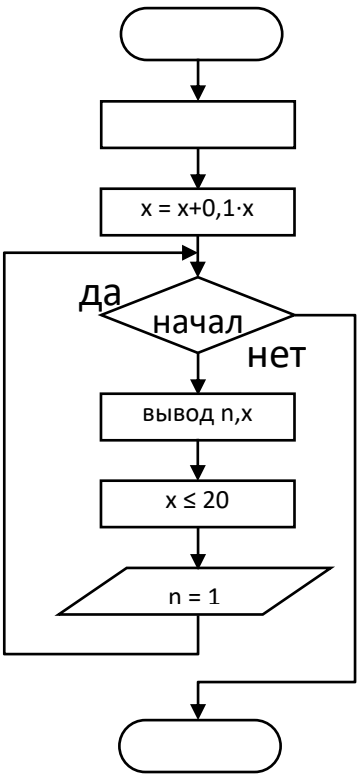
```
while <условие>:
    <блок_операторов>
```

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж



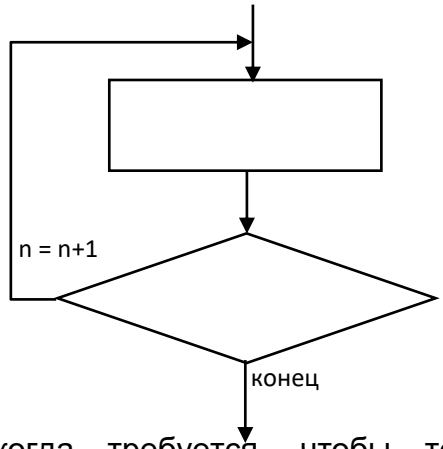
Задача

Лыжник в первый день тренировок пробежал 10 км. Каждый следующий день он увеличивал пройденное расстояние на 10% от пройденного в предыдущий день. В какой день он пробежит больше 20 км?

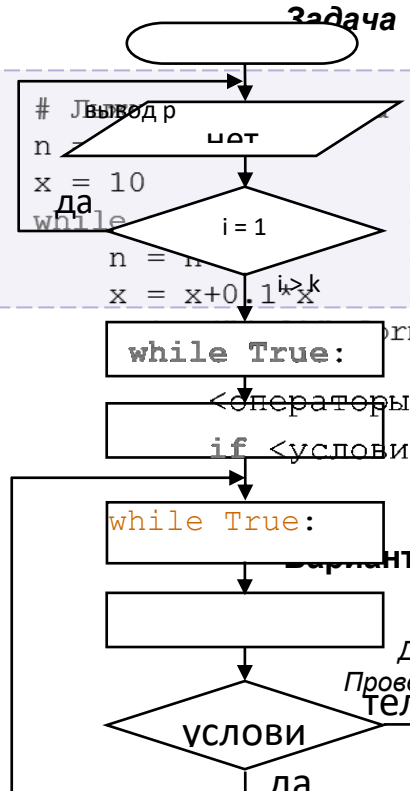


2. **Цикл while в Python** *Цикл с постусловием (с заданным условием окончания работы, цикл «ДО»)*

В языке Python нет оператора цикла с постусловием, но его можно организовать с помощью оператора while («пока») с условием True («истина»). Такой цикл будет выполняться бесконечно. Выход из цикла произойдет при истинности условия в операторе ветвления с помощью специального оператора break («прервать»).



Используется в тех случаях, когда требуется, чтобы тело цикла выполнилось хотя бы один раз.



15). **k! = 1·2·3· … ·k**

x = 10

```
# Л                                    
n =                  # день номер 1
x = 10               # р
while                # п
    n = n+1          # номер следующего дня
    x = x+0.1*x      # расстояние в следующий день
                     rmat(n), "{:6.1f}".format(x))
```

Вариант 0

Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

Решение:

Для начала введём целое число командой int(input(текст сообщения)).

Затем зададим переменной i значение 2. Переменная i выполняет роль счётчика. Если задать ей значение 1, то условие задачи не будет выполнено, а результатом всегда будет 1.

В цикле while в качестве логического выражения используется команда n%i сравниваемая с нулём. Таким образом, если остаток от деления введённого числа на текущее значение i не равно нулю, то счётчик увеличивается на 1, а если равно нулю цикл заканчивается и командой print() выводится сообщение и значение i.

Пример программы с циклом while

```
n = int(input('Введите целое число не меньшее 2\n'))
i = 2
while n%i != 0:
    i+=1
print('наименьший натуральный делитель:',i)
```

```
Введите целое число не меньшее 2
49
наименьший натуральный делитель: 7
```

***Задания для самостоятельного выполнения*:**

Вариант 1

Дана непустая последовательность целых чисел, оканчивающаяся нулем. Найти: а) сумму всех чисел последовательности; б) количество всех чисел последовательности

Решить задачу используя циклическую конструкцию while.

Вариант2

Дана последовательность отрицательных целых чисел, оканчивающаяся положительным числом. Найти среднее арифметическое всех чисел последовательности (без учета положительным числа).

Решить задачу используя циклическую конструкцию while.

Вариант 3

Дана последовательность из n целых чисел. Первое число в последовательности чётное. Найти сумму всех идущих подряд в начале последовательности чётных чисел. Условный оператор не использовать

Решить задачу используя циклическую конструкцию while.

Вариант 4

Дана последовательность из n вещественных чисел, начинающаяся с положительного числа. Определить, какое количество положительных чисел записано в начале последовательности. Условный оператор не использовать.

Решить задачу используя циклическую конструкцию while.

Вариант5

Дано целое число N (> 0), являющееся некоторой степенью числа 2: N = 2^K. Найти целое число K — показатель этой степени.

Решить задачу используя циклическую конструкцию while.

Вариант 6

Дано целое число N (> 1). Найти наименьшее целое число K, при котором выполняется неравенство 5^K > N.

Решить задачу используя циклическую конструкцию while.

Вариант 7

Дано целое число N (> 1). Найти наибольшее целое число K, при котором выполняется неравенство 2^K > N.

Решить задачу используя циклическую конструкцию while.

Вариант 8

Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр.

Решить задачу используя циклическую конструкцию while.

Вариант 9

Среди чисел 1, 5 10, 16, 23, ... найти первое число, большее n. Условный оператор не использовать.

Решить задачу используя циклическую конструкцию while.

Вариант 10

Дано число A (> 1). Вывести наибольшее из целых чисел K, для которых сумма 1 + 1/2 + … + 1/K будет меньше A, и саму эту сумму.

Решить задачу используя циклическую конструкцию while.

Вариант 11

Дано целое число N (> 0). Найти наибольшее целое число K, квадрат которого не превосходит N: K^2 ≤ N. Функцию извлечения квадратного корня не использовать.

Решить задачу используя циклическую конструкцию while.

Вариант 12

Выведите на экран для числа 2 его степени от 0 до 20

Решить задачу используя циклическую конструкцию while.

Вариант 13

Мой богатый дядюшка подарил мне один доллар в мой первый день рождения. В каждый день рождения он удваивал свой подарок и прибавлял к нему столько долларов, сколько лет мне исполнилось. Написать программу, указывающую, к какому дню рождения подарок превысит 100$.

Решить задачу используя циклическую конструкцию while.

Вариант 14

Вывести таблицу значений функции y = -0.5x + x. Значения аргумента (x) задаются минимумом, максимумом и шагом. Например, если минимум задан как 1, максимум равен 3, а шаг 0.5. То надо вывести на экран изменение x от 1 до 3 с шагом 0.5 (1, 1.5, 2, 2.5, 3) и значения функции (y) при каждом значении x.

Решить задачу используя циклическую конструкцию while.

Вариант 15

Найти сумму и произведение цифр, введенного целого числа. Например, если введено число 325, то сумма его цифр равна 10 (3+2+5), а произведение 30 (3*2*5).

Решить задачу используя циклическую конструкцию while.

Задача***Напишите программу-игру. Компьютер загадывает случайное число, пользователь пытается его угадать. Пользователь вводит число до тех пор, пока не угадает или не введет слово «Выход». Компьютер сравнивает число с введенным и сообщает пользователю больше оно или меньше загаданного.

Выводы и предложения о проделанной работе

Содержание отчета:

1.Наименование практического занятия;

2.Цель занятия;

3.Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

4.Список используемых источников;

5.Выводы и предложения;

6.Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1. Какие типы цикличных выражений существуют в Python?

2. Как организовать цикл с постусловием в Python?

3. Напишите синтаксис цикла while с предусловием .

Практическое занятие № 15 Функция range. Синтаксис цикла с параметром. Составление программ профессиональной направленности.

Цель занятия:

1. Познакомиться с циклической конструкцией- цикл с параметром;
2. Познакомиться с функцией range;
3. Уметь писать коды программ, содержащие циклические конструкции с for;
4. Формировать ОК5, ОК6, ЛР26, ЛР30 .

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

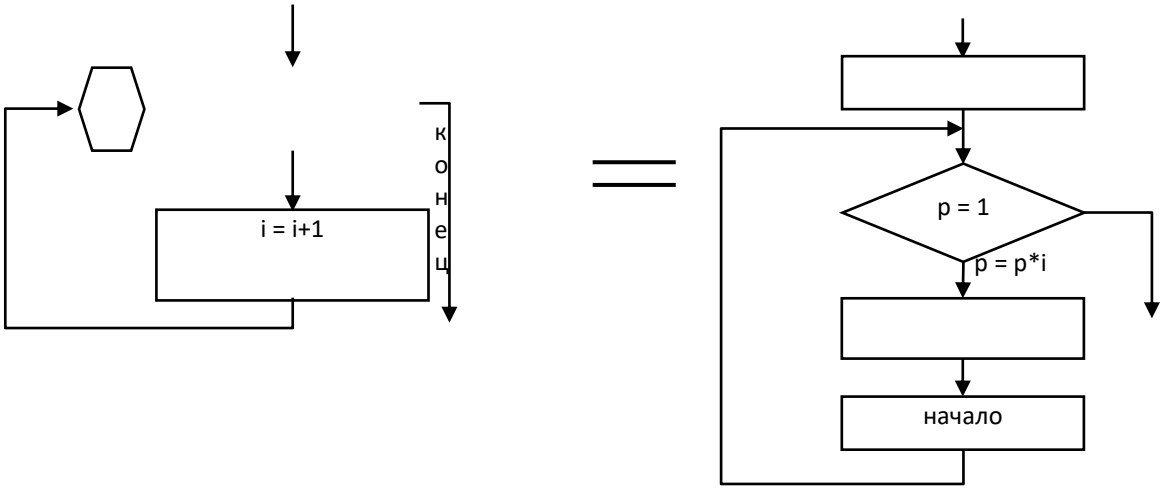
2. Выполнить задания.

Теоретическая часть

Цикл for в Python:

Цикл for в Python обладает способностью перебирать элементы любого комплексного типа данных (например, строки или списка).

Цикл с параметром (с заданным числом повторений, цикл «ДЛЯ»)



Синтаксис цикла for:

```
k < 13
```

Тело цикла повторяется *фиксированное число раз* **для** каждого значения параметра. Параметр – переменная целого типа. Функция **range** («диапазон») задаёт количество повторов тела цикла и содержит от одного до трёх чисел.

range(старт, стоп, шаг) - так выглядит стандартный вызов функции range() в Python. По умолчанию старт равняется нулю, шаг единице.

Одно число **(k)** – параметр цикла изменяется *от **0** до **k-1** с шагом **1***.

Два числа **(n, k)** – параметр цикла изменяется *от **n** до **k-1** с шагом **1***.

Три числа **(n, k, s)** – параметр цикла изменяется *от **n** до **k-1** с шагом **s***.

Возможно изменение параметра от большего значения к меньшему. В этом случае **n** должно быть больше **k**, а **s** – отрицательное.

Функция range()

Достаточно часто при разработке программ необходимо получить последовательность (диапазон) целых чисел:

Для решения этой задачи в Python предусмотрена функция range(), создающая последовательность (диапазон) чисел. В качестве аргументов функция принимает: начальное значение диапазона (по умолчанию 0), конечное значение (не включительно) и шаг (по умолчанию 1). Если вызвать функцию, то результата мы не увидим:

```
>>> range(0,10,1)
range(0, 10)
>>> range(10)
range(0, 10)
```

Дело в том, что для создания диапазона чисел необходимо использовать цикл for:

```
>>> for i in range(0, 10, 1): print(i, end=' ')

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

```
>>> for i in range(10):
print(i, end=' ')
0 1 2 3 4 5 6 7 8 9
```

Синтаксис цикла for:

```
for int in range():
    команда 1
    команда 2
     ...
   команда
```

Переменной int присваивается значение первого элемента функции range(), после команда 2 чего выполняются команды. Затем переменной int присваивается следующее по порядку значение и так далее до тех пор, пока не будут перебраны все элементы команда функции range().

Функция range() является универсальной функцией Python для создания списков (list) содержащих арифметическую прогрессию. Чаще всего она используется в циклах for.

Вариант 0

1. Найти сумму n элементов следующего ряда чисел: 1 -0.5 0.25 -0.125 ... n. Количество элементов (n) вводится с клавиатуры. Вывести на экран каждый член ряда и его сумму. Решить задачу используя циклическую конструкцию for.

Решение:

В данном случае ряд чисел состоит из элементов, где каждый следующий меньше предыдущего в два раза по модулю и имеет обратный знак. Значит, чтобы получить следующий элемент, надо предыдущий разделить на -2.

Какой-либо переменной надо присвоить значение первого элемента ряда (в данном случае это 1). Далее в цикле добавлять ее значение к переменной, в которой накапливается сумма, после чего присваивать ей значение следующего элемента ряда, разделив текущее значение на -2. Цикл должен выполняться n раз.

```
n=int(input('Введите количество элементов последоваетльности: '))
x=1
s=0
print(x)
for i in range(n):
    s+=x
    x/=-2
    print(x)
print('Сумма ряда:',s)
```

Пример программы с циклом for

```
Введите количество элементов последоваетльности: 5
1
-0.5
0.25
-0.125
0.0625
-0.03125
Сумма ряда: 0.6875
```

Результат выполнения программы

Вариант 1

1. Дано вещественное число – цена 1 кг конфет. Вывести стоимость 1, 2, … 10 кг конфет. Решить задачу используя циклическую конструкцию for.

Вариант2

1. Даны два числа A и B (A<B). Найти сумму всех целых чисел от A до B включительно. Решить задачу используя циклическую конструкцию for.

Вариант 3

1. Даны два числа A и B (A<B). Найти сумму квадратов всех целых чисел от A до B включительно. Решить задачу используя циклическую конструкцию for.

Вариант 4

1. Найти среднее арифметическое всех целых чисел от a до 200 (значения a и b вводятся с клавиатуры; a≤ 200). Решить задачу используя циклическую конструкцию for.

Вариант 5

1. Найти сумму всех целых чисел от a до b (значения a и b вводятся с клавиатуры; b ≥a). Решить задачу используя циклическую конструкцию for.

Вариант 6

1. Найти сумму квадратов всех целых чисел от a до 50 (значение a вводится с клавиатуры; 0 ≤a≤50). Решить задачу используя циклическую конструкцию for.

Вариант 7

1. Дана непустая последовательность целых чисел, оканчивающаяся нулем.

Найти:

а) сумму всех чисел последовательности;

б) количество всех чисел последовательности.

Решить задачу используя циклическую конструкцию for.

Вариант 8

1. Дана последовательность из n вещественных чисел. Первое число в последовательности нечетное. Найти сумму всех идущих подряд в начале последовательности нечетных чисел. Условный оператор не использовать. Решить задачу используя циклическую конструкцию for.

Вариант 9

1. Среди чисел 1, 4, 9, 16, 25, ... найти первое число, большее n. Решить задачу используя циклическую конструкцию for.

Вариант 10

1. Известны оценки по физике каждого из 20 учеников класса. Определить среднюю оценку. Решить задачу используя циклическую конструкцию for.

Вариант 11

1. Известно сопротивление каждого из элементов электрической цепи. Все элементы соединены последовательно. Определить общее сопротивление цепи. Решить задачу используя циклическую конструкцию for.

Вариант 12

1. Известны оценки по физике каждого ученика двух классов. Определить среднюю оценку в каждом классе. Количество учащихся в каждом классе одинаковое. Решить задачу используя циклическую конструкцию for.

Вариант 13

1. В области 12 районов. Известны количество жителей (в тысячах человек) и площадь (в км2) каждого района. Определить среднюю плотность населения по области в целом. Решить задачу используя циклическую конструкцию for.

Вариант 14

1. Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, сколько клеток будет через 3, 6, 9, ..., 24 часа, если первоначально была одна амеба. Решить задачу используя циклическую конструкцию for.

Вариант 15

Банк принимает вклады под 10% годовых. На счет положена сумма 10000 рублей. Какая сумма будет на счету через 5 лет?

Решить задачу используя циклическую конструкцию for.

Задача*

да

Выводы и предложения о проделанной работе

Содержание отчета:

7.Наименование практического занятия;

8.Цель занятия;

9.Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

10. Список используемых источников;

11. Выводы и предложения;

12. Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1. Какие типы цикличных выражений существуют в Python?

2. Какой способностью обладает цикл for в Python?

3. Как работает функция range?

Практическое занятие № 16 Понятие списка в Python. Создание и считывание списков. Функции и методы списков. Работа со строками. Понятие картежа и словаря. Создание словарей и кортежей. Методы словарей.

Цель занятия:

1. Изучить методы работы со строками;
2. Овладеть умениями по созданию и считыванию списков;
3. Изучить понятие кортежей и словарей в Python;
4. Овладеть операциями и методами над кортежами и словарями;

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;
2. Выполнить задания.

Теоретическая часть 1

Строка — базовый тип, представляющий из себя неизменяемую последовательность символов; str от «string» — «строка».

Строковая константа (строка) – произвольная последовательность символов из таблицы Unicode, заключенная в одинарные или двойные кавычки (тип **str** – «string»).

Например:

'Это строка'

"Это тоже строка“

Вызовем функцию type() и передадим ей на вход целочисленный аргумент:

```
>>> type(0)
<class 'int'>
>>>
```

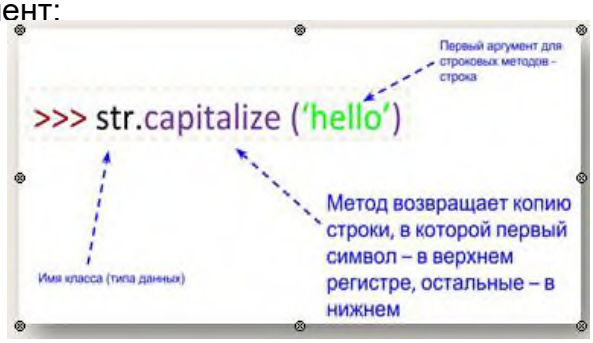
Функция сообщила нам, что объект 0 относится к классу 'int', т.е. тип данных является классом (тип данных и класс – синонимы).

Класс будем представлять, как некий аналог модуля, т.е. набор функций и переменных, содержащихся внутри класса. Функции, которые находятся внутри класса, называются методами.

Рассмотрим пример вызова строкового метода:

```
>>> str.capitalize('hello')
'Hello'
>>>
```

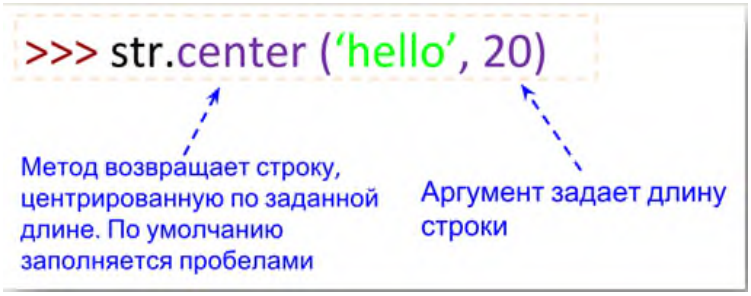
По аналогии с вызовом функции из модуля указываем имя класса – str, затем через точку пишем имя строкового метода capitalize(), который принимать один строковый аргумент:



Метод – это обычная функция, расположенная внутри класса. Вызовем еще один метод:

```
>>> str.center('hello',20)
'          hello           '
>>>
```

тот метод принимает два аргумента – строку и число:



Форма вызова метода через обращение к его классу через точку называется полной формой.

Постоянно писать имя класса перед вызовом каждого метода быстро надоест, поэтому чаще всего используют сокращенную форму вызова метода:

```
>>> 'hello'.capitalize() 'Hello'
>>>
```

В примере мы вынесли первый аргумент метода и поместили его вместо имени класса:

Вынесенный из метода первый строковый аргумент может быть выражением, возвращающим строку:

```
>>> ('TTA' + 'G'*3).count('T') 2
>>>
```

Не сложно догадаться, что делает метод count().
Python содержит интересный метод format()38:

```
>>> '{0} и {1}'.format('труд', 'май')
 'труд и май'
>>>
```

Вместо {0} и {1} подставятся аргументы методы format(). Поменяем их местами:

```
>>> '{1} и {0}'.format('труд', 'май') 'май и труд'
>>>
```

Формат вывода методы format() может варьироваться:

```
>>> n = 10
>>> '{:b}'.format(n)              # вывод в двоичной системе счисления
'1010'
>>> '{:c}'.format(n)              # вывод в формате Unicode
'\n'
>>> '{:d}'.format(n)              # по снованию 10
'10'
>>> '{:x}'.format(n)              # по основанию 16
'a'
>>>
```

В Python есть полезные строковые методы, которые возвращают (True) истину или (False) ложь:

```
>>>'spec'.startswith('a')
False
>>>
```

Метод startswith() проверяет, начинается ли строка с символа, переданного в качестве аргумента методу.
Метод swapcase() возвращает строку с противоположными регистрами символов:

```
>>>'Hello'.swapcase()
'hELLO'
>>>
```

Специальные строковые методы

Длина строки – количество символов в строке .

Пустая строка – строка с нулевой длиной (не содержит ни одного символа, обозначается **""**).

1. Присваивание значения строковой переменной

s = "Привет"

2. Ввод строки с клавиатуры

n = input("Введите имя: ")

Примечание: при вводе значения строки кавычки не вводятся.

3. Вывод строки на экран

print (n)

Примечание: при выводе строки кавычки не выводятся.

4. Объединение строк (конкатенация)

Соединяет несколько строк в одну строку. Обозначается знаком **+**.

Например:

"КОМ"+"ПЬЮ"+"ТЕР" # "КОМПЬЮТЕР"

"10"+"2" # "102"

Операции или методы работы со строковыми величинами вы можете посмотреть в таблице ниже.

Функции и методы работы со строками

Функция или метод	Назначение
S1 + S2	Конкатенация (сложение строк)
S1 * 3	Повторение строки
S[i]	Обращение по индексу
S[i:j:step]	Извлечение среза
len(S)	Длина строки
S.join(список)	Соединение строк из последовательности str через разделитель, заданный строкой
S1.count(S[, i, j])	количество вхождений подстроки s в строку s1. Результатом является число. Можно указать позицию начала поиска i и окончания поиска *j*
S.find(str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или -1
S.index(str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError
S.rindex(str, [start],[end])	Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError
S.replace(шаблон, замена)	Замена шаблона
S.split(символ)	Разбиение строки по разделителю

S.upper()	Преобразование строки к верхнему регистру
S.lower()	Преобразование строки к нижнему регистру
s.title()	Возвращает строку, первый символ которой в верхнем регистре
s.count('e', 1, 5)	Возвращает количество подстрок в интервале либо -1
s.isalpha()	Проверяет, состоит ли строка только из букв
s.isdigit()	Проверяет, состоит ли строка только из чисел
s.isupper()	Проверяет, написаны ли все символы в верхнем регистре
s.islower()	Проверяет, написаны ли все символы в нижнем регистре
s.istitle()	Проверяет, начинается ли строка с большой буквы
s.isspace()	Проверяет, состоит ли строка только из пробелов

Каждый из перечисленных ниже методов запустить в интерактивном режиме на примере различных строк.

Предположим, что переменная s содержит некоторую строку, тогда применим к ней методы:

При работе со строковыми величинами можно выполнять :

1.Выделение части строки (подстроки)

s[n:k]

Выделяет из строки **s** часть строки от позиции **n** до **k-1**.

s[:k]

Выделяет из строки **s** часть строки от начала до **k-1**.

s[n:]

Выделяет из строки **s** часть строки от позиции **n** до конца.

Примечание: символы нумеруются, начиная с 0.

Например:

```
s = "ИНФОРМАТИКА"
print(s[2:7])      # "ФОРМА"
```

2.Преобразование типов

При необходимости можно преобразовать число в строку или наоборот.

int(s) – преобразует строку в целое число.

float(s) – преобразует строку в вещественное число.

str(n) – преобразует целое или вещественное число в строку.

Примечание: если в строке содержатся символы, не допустимые для чисел, возникнет ошибка.

Например:

```
s1 = "123"          # строка цифр
i = int(s1)         # i=123
f = float(s1)       # f=123.0
s2 = str(f)         # s2="123.0"
print(i, f, len(s2)) # 123 123.0 5
```

3.*Операции с кодами символов*

ord(s) – возвращает код символа **s**.

chr(k) – возвращает символ с кодом **k**.

Примечание: коды из кодовой таблицы Unicode.

Например:

```
print (ord("И"))                    # 1048
print (chr(1048)+chr(1050)+chr(1058))   # ИКТ
```

4.*Сравнение строк*

Операции отношения: **==, !=, <, >, <=, >=.**

Сравнение строк производится слева направо до первого несовпадающего символа. Строка считается больше, если первый несовпадающий символ имеет больший код в кодовой таблице (пробел, цифры, латинские заглавные, латинские строчные, русские заглавные, русские строчные). Строки равны, если они совпадают по длине и содержат одни и те же символы.

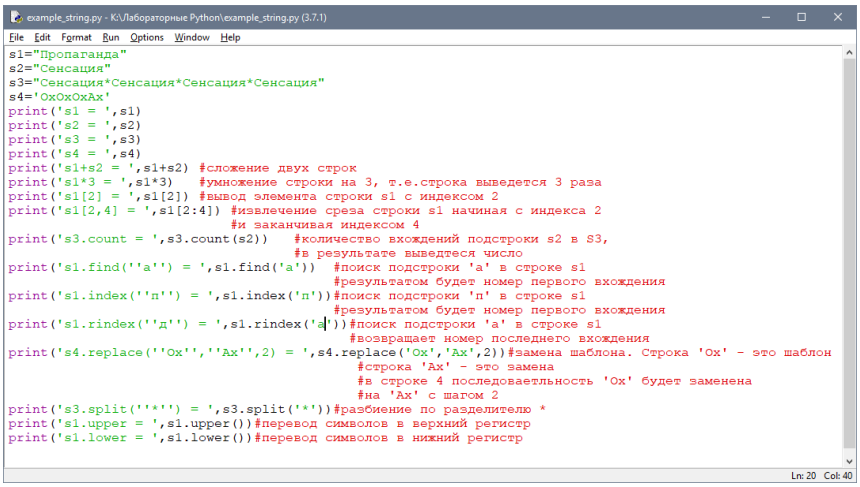
"1STR" < "STR" < "Str" < "str" < "ШАГ" < "Шаг" < "шаг"

5.*Перебор символов строки*

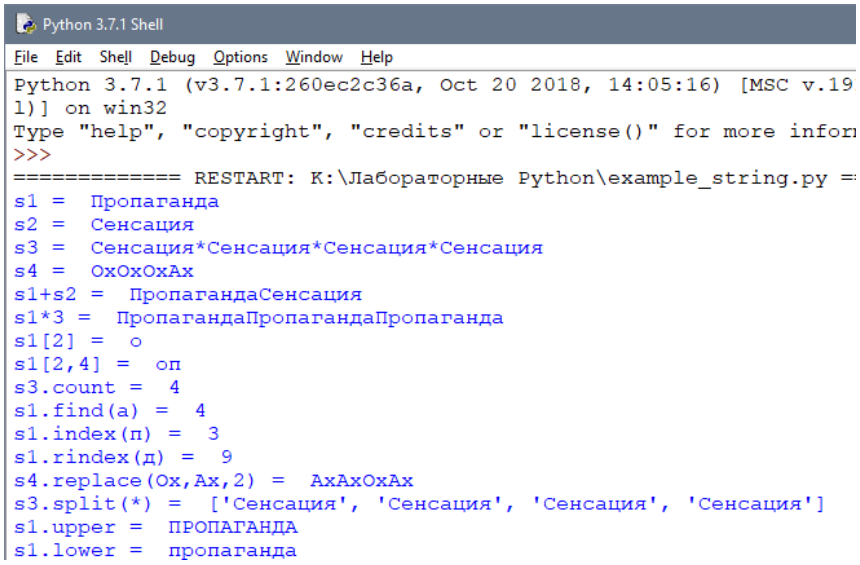
Часто в программах требуется перебирать по одному все символы строки для проведения с ними каких-либо действий. Для этого удобно использовать следующий цикл:

```
for C in S:      # перебор символов в строке S
    <операторы> # очередной символ в перем. C
```

Ниже приведена программа, демонстрирующая использование функций и методов работы со строками.



Пример программы на Python



Результат выполнения программы с использованием функций и методов работы со строками

Список (list) – это структура, состоящая из элементов, расположенных в определенном порядке. Каждому элементу соответствует номер (или индекс), по которому к нему можно обратиться. Для создания списка вквадратных скобках ([]) через запятую перечисляются все его элементы. Например, создадим список членов своей семьи:

```
>>>myfamily=['father','mother','sister','brother']
```

Программа на Python	Результат вывода на экран
myfamily = ['father', 'mother', 'sister', 'brother'] for item in myfamily: print('Hello', item)	Hello father Hello mother Hello sister Hello brother

В данном случае наш список будет храниться в переменной myfamily.

Когда список создан, можно написать программу для работы с этим списком. К примеру, напишем приветствие для каждого из членов семьи, используя цикл:

Каждый элемент списка имеет свой номер (индекс). Нумерация элементов начинается с нуля:

a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]

0	1	2	3	4
Андрей	Вера	Даша	Коля	Юра
a[0]	a[1]	a[2]	a[3]	a[4]

С каждым элементом списка можно работать отдельно:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
print(a[2])
```

Вывод: **Даша**

Список может содержать разные типы объектов. В один и тот же список одновременно можно включать строки, числа, объекты других типов данных:

objects=[1,2.6,'Hello',True]

Списки можно складывать, тогда новый список будет содержать элементы из обоих списков:

x = [1, 2, 3, 4]

y = [5, 6, 7, 8]

z = x + y print (z)

Результат: [1, 2, 3, 4, 5, 6, 7, 8]

Со списками можно делать много разных операций:

x in A	Проверить, содержится ли элемент **x** в списке **A.** Возвращает True или False	`A = [1, 2, 3, 4, 5, 6, 7, 8]`
		`print (2 in A)`
		`Результат:`
		`True`

min(A)	Найти наименьший элемент в списке **A.**	`A = [1, 2, 3, 4, 5, 6, 7, 8]`
		`print (min(A))`
		`Результат:`
		`1`

max(A)	Найти наибольший элемент в списке **A.**	`A = [1, 2, 3, 4, 5, 6, 7, 8]`
		`print (max(A))`
		`Результат:`
		`8`

Функция нахождения длины списка len(a):

```
a = ["Яблоко", "Банан", "Груша"]
x = len(a)
print(x)
```

Результат:3

Сортировка списка. Функция sorted(a):

По возрастанию:

```
animals = ["кот", "еж", "собака", "барсук"]
animals = sorted(animals)
print(animals)
```

Вывод:

['барсук', 'еж', 'кот', 'собака']

Поубыванию:

```
a = [5, 65, 14, 700, 8]
a = sorted(a, reverse = True)
print(a)
```

Вывод:

[700, 65, 14, 8, 5]

Способы вывода списков

1)*С помощью функции*print():

b = [17, 409, 88]

s=[‘f’,’g’,’r’,’a’]

print(b)

print(s)

Вывод:

[**17, 409, 88**]

[f,g,r,a]

2) *Вывод каждого элемента списка по отдельности*:

a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]

fori in range(5):

print(a[i])

Вывод:

Андрей

Вера

Даша

Коля

Юра

3)Вывод каждого элемента списка поотдельности в одной строке:

a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]

fori in range(5):

print(a[i], end = " ")

Вывод

Андрей Вера Даша Коля Юра

4) Вывод элементов списка без обращения к индексам элементов:

fruits = ["Яблоко", "Банан", "Груша"]

for x in fruits:

print(x, end = " ")

Вывод

Яблоко Банан Груша

Способы вывода списков

1)С помощью функции print():

```
b = [17, 409, 88]
s=[‘f’,’g’,’r’,’a’]
print(b)
print(s)
```

Вывод:

[17, 409, 88]
[f,g,r,a]

2) Вывод каждого элемента списка по-отдельности:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
    print(a[i])
```

Вывод:

Андрей
Вера
Даша
Коля
Юра

3)Вывод каждого элемента списка по-отдельности в одной строке:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
    print(a[i], end = " ")
```

Вывод

Андрей Вера Даша Коля Юра

4) Вывод элементов списка без обращения к индексам элементов:

```
fruits = ["Яблоко", "Банан", "Груша"]
for x in fruits:
    print(x, end = " ")
```

Вывод

Яблоко Банан Груша

Пример

Вариант 0

Проверить, будет ли строка читаться одинаково справа налево и слева направо (т. е. является ли она палиндромом).

Решение

Сначала введём строку командой: s=input('Введите строку ').

Затем определим логическую переменную flag и присвоим ей значение 1: flag=1.

Для начала в введённой строке нужно удалить пробелы. Для этого воспользуемся циклической конструкцией for, которая выполнится столько раз, какую имеет длину строка. Длину строки определим функцией len(s).

В теле цикла будем проверять следующее условие: s[i]!=' '. Данное логическое выражение будет истинно в том случае, если i-ый элемент строки не будет равен пробелу, тогда выполнится команда следующая после двоеточия: string+=s[i].

К строке string, которая была объявлена в начале программы, будет добавляться посимвольно строка s, но уже без пробелов.

Для проверки строки на "палиндром" воспользуемся циклической конструкцией for.

Длина половины строкинаходится делением нацело на 2.Если количество символов нечетно,то стоящий в середине не учитывается,т.к. его сравниваемая пара - он сам.

Количество повторов цикла равно длине половины строки. Длину строки определим функцией len(s), где аргумент введённая нами строка s. Зная длину строки, можно вычислить количество повторов цикла. Для этого целочисленно разделим длину строки на 2: len(s)//2.

Для задания диапазона для цикла используем функцию range(), в которой аргументом будет являться половина длины строки: range(len(s//2)).

for i in range(len(s//2)).

Если символ с индексом i не равен "симметричному" символу с конца строки (который находится путем индексации с конца)

if s[i] != s[-1-i],

то переменной flag присваивается значение 0 и происходит выход из цикла командой break.

Далее, при помощи условной конструкции if-else в зависимости от значения flag либо - 0, либо -1 выводится сообщение, что строка палиндром, либо нет.

```
s=input('Введите строку \n')
flag=1
string=''
for i in range(len(s)):
    if s[i]!=' ':
        string+=s[i]
print(string)
for i in range(len(s)//2):
    if string[i]!=string[-i-1]:
        flag=0
        break
if flag: print('Палиндром')
else: print('не палиндром')
```

Пример программы на Python

```
Введите строку
а роза упала на лапу азора
арозаупаланалапуазора
Палиндром
```

Результат выполнения программы

Задания для самостоятельной работы (по вариантам)

Вариант 1

1.Дана строка, содержащая русскоязычный текст. Найти количество слов, начинающихся с буквы "е".

2.Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Вариант 2

1.В строке заменить все двоеточия (:) знаком процента (%). Подсчитать количество замен.

2.Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'
Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.
Подсказка: используйте метод choice() модуля random.

Вариант 3

1.В строке удалить символ точку (.) и подсчитать количество удаленных символов.

2.Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Вариант 4

1.В строке заменить букву(а) буквой (о). Подсчитать количество замен.Подсчитать, сколько символов в строке.

2.Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'
Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.
Подсказка: используйте метод choice() модуля random.

Вариант 5

1.В строке заменить все заглавные буквы строчными.

2.Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе

добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Вариант 6

1.В строке удалить все буквы "а" и подсчитать количество удаленных символов.

2.Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'
Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.

Подсказка: используйте метод choice() модуля random.

Вариант 7

1.Дана строка. Преобразовать ее, заменив звездочками все буквы "п", встречающиеся среди первых n/2 символов. Здесь n - длина строки.

2. Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Вариант 8

1.Дана строка, заканчивающаяся точкой. Подсчитать, сколько слов в строке.

2.Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'

Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.
Подсказка: используйте метод choice() модуля random.

Вариант 9

1.Определить, сколько раз в тексте встречается заданное слово.

2.Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Вариант 10

1.Дана строка-предложение на английском языке. Преобразовать строку так, чтобы каждое слово начиналось с заглавной буквы.

2.Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'
Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.
Подсказка: используйте метод choice() модуля random.

Вариант 11

1.Дана строка. Подсчитать самую длинную последовательность подряд идущих букв «н». Преобразовать ее, заменив точками все восклицательные знаки.

2.Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Вариант 12

1.Дана строка. Вывести все слова, оканчивающиеся на букву "я".

2.Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'
Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.
Подсказка: используйте метод choice() модуля random.

Вариант 13

1.Дана строка символов, среди которых есть одна открывающаяся и одна закрывающаяся скобки. Вывести на экран все символы, расположенные внутри этих скобок.

2.Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Вариант 14

1.Дана строка. Вывести все слова, начинающиеся на букву "а" и слова оканчивающиеся на букву "я".

2.Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'
Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.
Подсказка: используйте метод choice() модуля random.

Вариант 15

1.Дана строка текста. Подсчитать количество букв «т» в строке

2.Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Теоретическая часть 2

Часто нам нужно держать много однообразных данных в одном файле, например, список учеников колледжа или номера телефонов в справочнике. В Python такие наборы данных можно организовывать в списки, кортежи и словари.

Кортеж (**tuple**), как и список, представляет собой последовательность элементов. Однако хранящиеся в нем элементы нельзя изменять, добавлять или удалять. Для создания кортежа используются круглые скобки, в которые помещаются его значения, разделенные запятыми:

```
user=('Timur',23,1.10.1998)
print(user)
```

В кортежах удобно хранить свойства объектов, например, имя, возраст, дату рождения. Если вдруг кортеж состоит из одного элемента, то после единственного элемента кортежа необходимо поставить запятую:

```
user=('Tom',)
```

Некоторые операции над кортежами:

```
>>> ()          #  создание пустого кортежа
()
>>> (4)         # это не кортеж, а целочисленный объект! 4
>>> (4,)        # а вот это – кортеж, состоящий из одного элемента! (4,)
>>> b=('1',2,'4')              # создаем кортеж
>>> b
('1', 2, '4')
>>> len(b)  # определяем длину кортежа
3
>>> t=tuple(range(10)) # создание кортежа с помощью функции range()
>>> t+b       # слияние кортежей
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, '1', 2, '4')
>>> r=tuple([1,5,6,7,8,'1'])                  # кортеж из списка
>>> r
```

```
(1, 5, 6, 7, 8, '1')
>>>
```

С помощью кортежей можно присваивать значения одновременно двум переменным:

```
>>> (x,y)=(10,5)
>>
>x
10
>>
>y
5
>>> x,y = 1,3 # если убрать круглые скобки, то результат не изменится
>>
>x
1
>>
>y
3
>>>
```

Поменять местами содержимое двух переменных:

```
>>> x, y = y, x
>>
>x
3
>>
>y
1
>>>
```

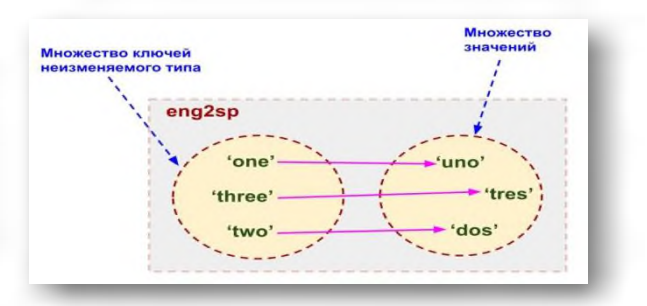
Кортеж нельзя изменить, но можно изменить, например, список, входящий в кортеж:

```
>>> t = (1,[1,3],'3')
>>>t[1]
[1, 3]
>>> t[1][0] = '1'
>>> t
(1, ['1', 3], '3')
>>>
```

Словари (**dictionary**) – это структура данных, в которой каждый элемент вместо индекса имеет уникальный ключ. Элементы словаря можно изменять. Для создания словаря используются фигурные скобки ({}):

1. Словари создаются с помощью фигурных скобок.
2. Пары из ключа и значения разделяются запятыми.
3. Ключи и значения разделяются между собой двоеточием
4. Ключи в словаре могут быть только строками, целыми числами или числами с плавающей точкой. А вот значения могут быть любого типа
5. Важно не забывать использовать кавычки для строки-ключа

Пример создания словаря, который каждому слову на английском языка будет ставить в соответствие слово на испанском языке.



Фактически, словарь – это отображение двух множеств: множества ключей и множества значений.

Некоторые из методов и функций словарей:

- .keys() — используется для вывода ключей словаря.
- .items() — используется для создания кортежей с ключами и значениями.
- .get() — метод для получения значения по ключу.
- .clear() — очистить словарь.
- .copy() — скопировать весь словарь.
- len() — получить длину словаря.
- type() — узнать тип.

К словарям применим оператор in:

```
>>> eng2sp
```

```
{'three': 'tres', 'one': 'uno', 'two': 'dos'}
>>> 'one' in eng2sp           # поиск по множеству КЛЮЧЕЙ
True
>>>
```

Часто словари используются, если требуется найти частоту встречаемости элементов в последовательности (списке, строке, кортеже).

Функция, которая возвращает словарь, содержащий статистику встречаемости элементов в последовательности:

```
def histogram(s):
    d = dict()
    for c in s:
        if c not in d:
            d[c]=1
        else:
            d[c]=d[c]+1 # или d[c] += 1
    return d
```

Результат вызова функции histogram() для списка, строки, кортежа соответственно:

```
>>> histogram([2,5,6,5,4,4,4,4,3,2,2,2,2])
{2: 5, 3: 1, 4: 4, 5: 2, 6: 1}
>>> histogram("ywte3475eryt3478e477477474")
{'4': 6, '8': 1, 'e': 3, '3': 2, '7': 7, '5': 1, 'r': 1, 'y': 2, 'w':
1, 't': 2}
>>> histogram((5,5,5,6,5,'r',5))
{5: 5, 6: 1, 'r': 1}
>>>
```

Начиная с версии Python 3.9, в языке появились новые операторы, которые облегчают процесс слияния словарей.

1. Merge (|): этот оператор позволяет объединять два словаря с помощью одного символа |.

2. Update (|=): с помощью такого оператора можно обновить первый словарь значением второго (с типом dict)

Вот основные отличия этих двух операторов:

- «|» создает новый словарь, объединяя два, а «|=» обновляет первый словарь.
- Оператор merge (|) упрощает процесс объединения словарей и работы с их значениями.
- Оператор update (|=) используется для обновления словарей.

dictionary={ключ1:значение1,ключ2:значение2,...}

Создадим словарь под именем mycollege:

```
mycollege = {'4kurs':'Анна, Кирилл, Pavel',
'3kurs':'Игорь,Олег,Ольга'}
```

В этом словаре в качестве ключей используются названия классов, а в качестве значений – имена тех, кто учится в этих классах.

В словарь можно добавить значение, пометив его новым ключом:

```
mycollege['2kurs']='Elena,Алиса,Дима’
print(mycollege)
```

Результат:

```
{'4kurs': 'Анна, Кирилл, Pavel', '3kurs':
'Игорь,Олег,Ольга','2kurs':'Elena,Алиса,Дима}
```

Чтобы изменить значение элемента, нужно придать его ключу новое значение:

```
mycollege = {'4kurs':'Анна, Кирилл, Pavel',
'3kurs':'Игорь,Олег,Ольга'}
mycollege['3kurs']='Matvei,Таня,Света'
print(mycollege)
```

Результат:

```
{'4kurs':Анна,Кирилл,Pavel','3kurs':'Matvei,Таня,Света'}
```

Используя циклfor,можно вывести на экран только ключи словаря:

```
foriinmycollege:
    print(i)
```

Результат:

```
4 kurs
3 kurs
```

Или вывести только значения словаря:

```
foriinmycollege:
    print(mycollege[i])
```

Результат:

```
Анна,Кирилл,Pavel
Matvei, Таня, Света
```

Задания для самостоятельного выполнения:

Задание №1

Создать список своих любимых фильмов. Вывести список тремя способами: а) в строчку; б) в столбик; в) в строчку через запятую.

Задание №2

Ввести с клавиатуры список фамилий 10 -ти писателей. Отсортировать список в алфавитном порядке и вывести его.

Задание №3

Создать список своих любимых песен. Отсортировать список в алфавитном порядке и вывести его. Найти длину списка и вывести результат на экран.

Задание №4

Создайте словарь «Рыбы», а его элементы разделите на 3 вида:

«речные», «озерные» и «морские рыбы». Выведите на экран сначала только ключи, а потом элементы словаря.

Задание №5

Добавить в данный словарь «Рыбы» ключ «океанические», заполнить его и вывести на экран обновленный словарь.

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия

3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»

4. Список используемых источников

5. Выводы и предложения

6. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки

1. Что такое список, кортеж, словарь в Питоне?
2. Как создаются списки?
3. Какие операции можно производить со списками?
4. Какие существуют способы вывода списка?
5. Как записывается функция определения длины списка?
6. Как отсортировать список?
7. Что называется строкой в Python
8. Перечислите функции и методы работы со строками.
9. Для чего служит функция S.split**?**
10. Для чего используется функция range()?

Практическое занятие № 17 Система координат в графике Python, управление цветом. Примитивы в Python Использование процедур в графике.

Цель занятия:

1. Изучить понятие графики Python;
2. Овладеть операциями и методами над процедурами в графике ;
3. Изучить понятие процедур в графике в Python;

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

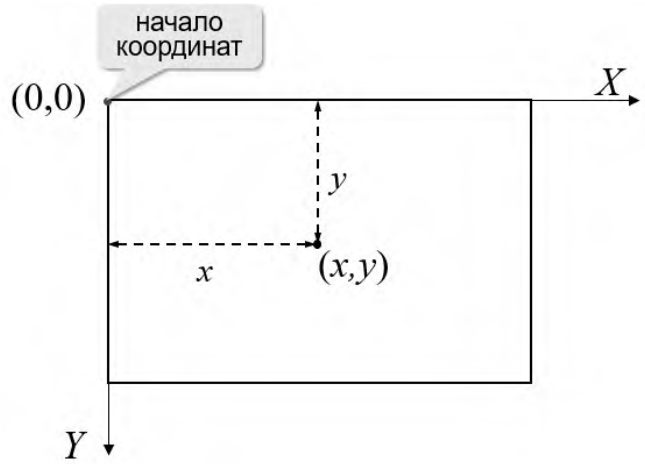
Изучить теоретическую часть

Выполнить задания

Подпрограммы прежде всего необходимы в ситуации, когда в разных частях программы необходимо выполнять одни и те же действия несколько раз. В таком случае повторяемые операторы оформляются в виде функции или процедуры, к которой можно обращаться и вызывать ее выполнение из разных частей программы.

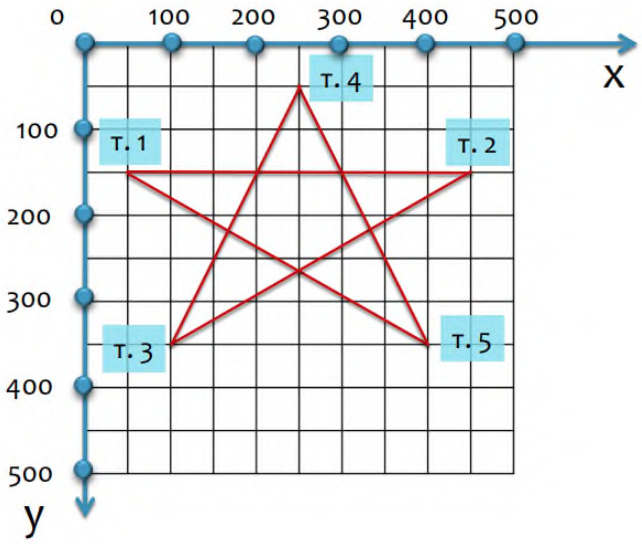
В python существует два вида реализации подпрограмм: функции в роли процедуры и функции в классическом понимании.

Функция в роли процедуры призвана не возвратить значение в основную программу, а вывести его, либо выполнить какие-либо действия с глобальными переменными, при этом не возвращая полученные значения основной программе (не используя ключевое слово return).



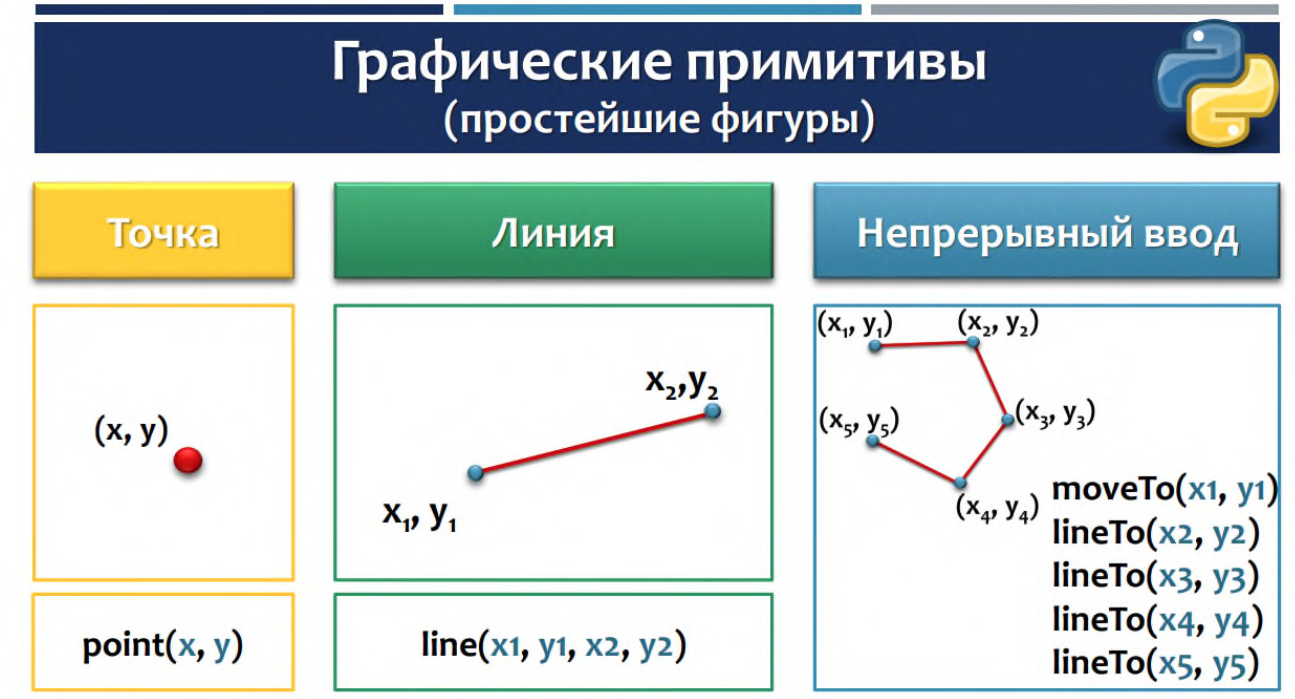
Верхняя левая точка экрана имеет координату (0,0), ось OX направлена вправо, ось OY – вниз, т.е. чем ниже на экране расположена точка, тем больше ее координата по оси OY.

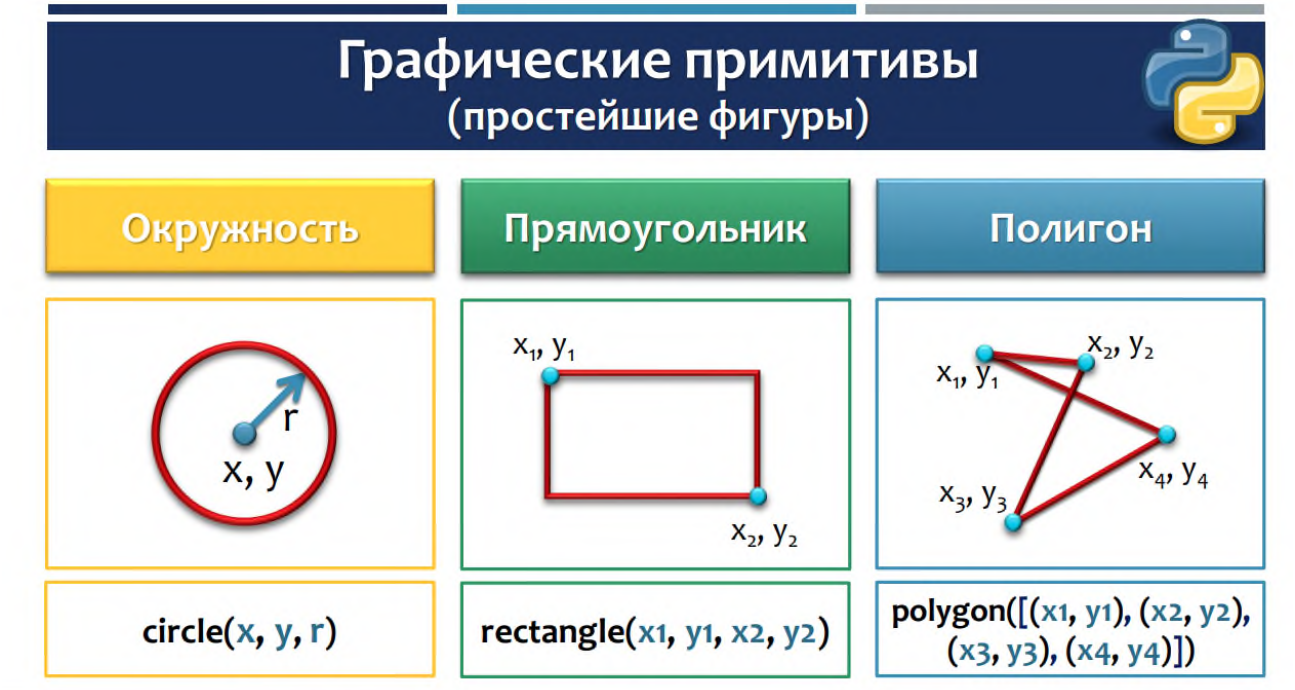
Задание №1 Определить координаты точек, изображённых на рисунке:



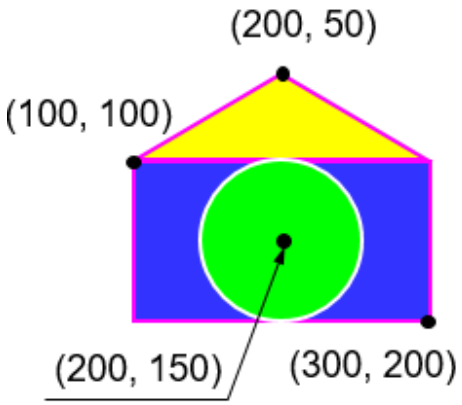
При построении изображения и его оформлении возможно использование нескольких цветов.

Цвет линий:	**penColor("red")**
Толщина линий:	**penSize(2)**
Цвет заливки:	**brushColor("green")**

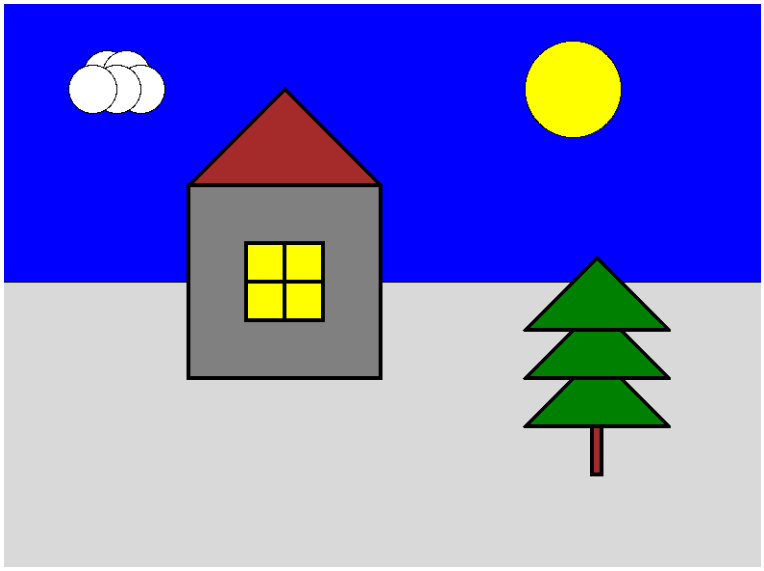




Пример программы и результат её работы.

from graph import * penColor("magenta") brushColor("blue") rectangle(100,100,300,200) brushColor("yellow") polygon([(100,100), (200,50),(300,100), (100,100)]) penColor("white") brushColor("green") circle(200, 150, 50) run()	

Задание №2 Используя полученные знания, нарисуйте любую статическую сцену, которая содержит не менее 3 различных объектов, состоящих из пяти и более примитивов.. Примером сцены может являться следующая картинка:



Вспомогательный алгоритм – это алгоритм решения какой-либо подзадачи, который может вызываться из основного алгоритма.

В программировании вспомогательные алгоритмы называют **подпрограммами**. В языке Python существуют два вида подпрограмм: процедуры и функции.

Процедура – это подпрограмма, которая выполняет некоторые действия после вызова её из основной программы или другой процедуры. Каждая процедура имеет уникальное ***имя,*** может иметь произвольное количество входных ***параметров***. При вызове процедуры указываются ***фактические значения параметров***.

Локальные переменные – это переменные, определённые в процедуре, они доступны только внутри процедуры.

Глобальные переменные – это переменные, определённые в основной программе. Они доступны внутри процедуры только для чтения, а для изменения требуется объявить их в процедуре после служебного слова **global**.

Процедура начинается служебным словом **def** (define – «определить»).

Формальные параметры процедуры перечисляются через запятую.

Операторы, входящие в тело процедуры, записываются с отступом. Процедура должна быть определена до первого её вызова.

def <имя>(<параметры>):

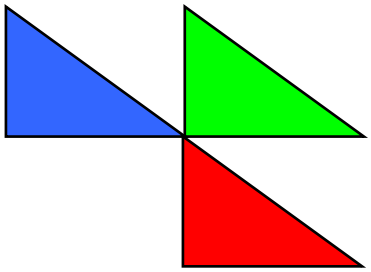
<операторы>

Вызов процедуры осуществляется по её имени с указанием фактических параметров (аргументов).

<имя>(<аргументы>)

Примечание: Между **формальными** и **фактическими** параметрами должно быть соответствие по количеству, порядку следования и типу.

Задача: Построить фигуру:



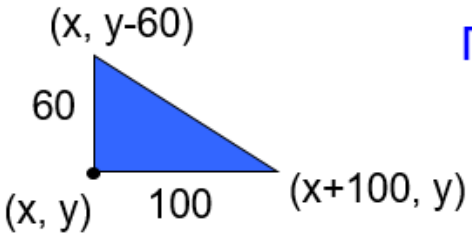
Особенность: Три похожие фигуры.

Общее: размеры, угол поворота

Отличия: координаты, цвет

Отличия обозначаем как переменные,

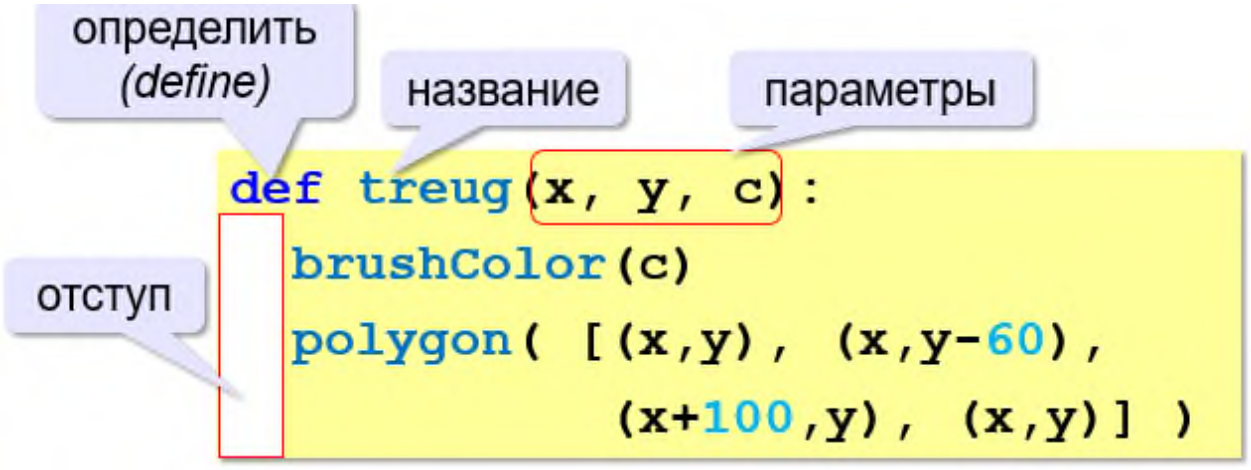
они будут параметрами процедуры.



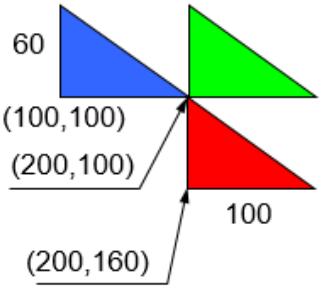
Параметры:

x, y – координаты угла

c – цвет заливки



Программа с процедурой



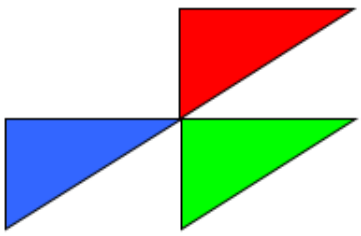

```
from graph import *
def treug(x, y, c):
  brushColor(c)
  polygon([(x,y),(x,y-60),
           (x+100,y),(x,y)] )
penColor ( "black" )
treug ( 100, 100, "blue" )
treug ( 200, 100, "green" )
treug ( 200, 160, "red" )
run()
```

вызовы процедуры

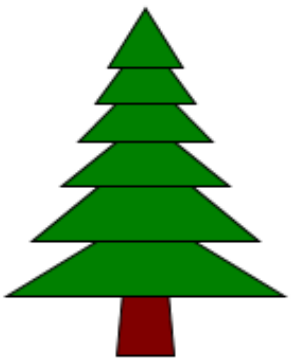
аргументы (значения параметров)

Задание №3 Используя одну процедуру, построить предложенные фигуры:

Оценка «3»

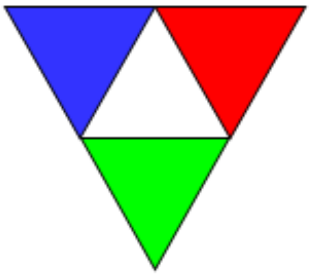


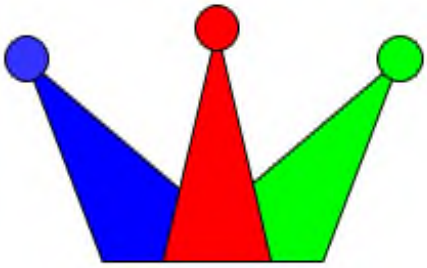
Оценка «5»



Оценка «4»

Оценка «5»





Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
5. Список используемых источников
6. Выводы и предложения
7. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что используют для написания программ с графическими возможностями?
2. Какой оператор позволяет нарисовать точку на экране?
3. Какой оператор позволяет нарисовать линию и прямоугольник?
4. Какие операторы позволяют нарисовать закрашенный прямоугольник?
5. Какой оператор позволяет нарисовать окружность?
6. Что называют подпрограммами?
7. Какие виды подпрограмм известны?
8. Когда целесообразно использовать подпрограммы в графике?

Практическое занятие № 18 Использование цикла в графике и закрашивание областей

Цель занятия:

1. Изучить понятие цикла в графике в Python;
2. Овладеть операциями и методами работы с циклами в графике;

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

Изучить теоретическую часть

Выполнить задания

Алгоритмы описываются с помощью трёх основных структур:

1) следования;

2) ветвления;

3) повторения.

Алгоритм повторения - описание повторяющихся действий. Алгоритм повторения может быть зацикленным и иметь принудительное прерывание и может быть с фиксированным числом повторений с естественным окончанием повторения или цикла.

В языке Python обычно используют два вида циклов:

– «цикл WHILE» или «цикл с предусловием» и попробуем написать первые игры.

WHILE – «пока» в переводе с английского

Например:

Пока <выполняется условие>: делать какие-то действия.

```
n=0
while n<3:
    n=n+1
```

«Пока n меньше 3, прибавлять к n единицу»

Цикл повторяется, пока условие истинно, если же нет, цикл заканчивается

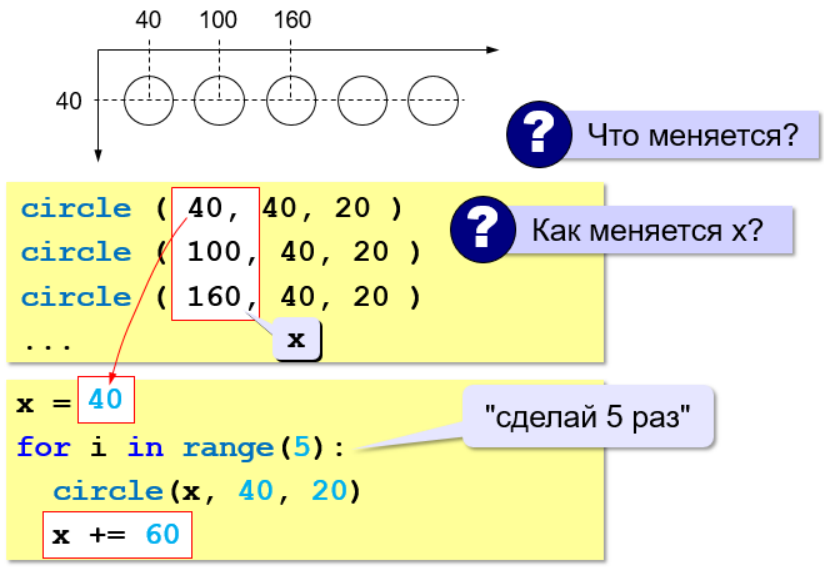
– цикл с параметром FOR.

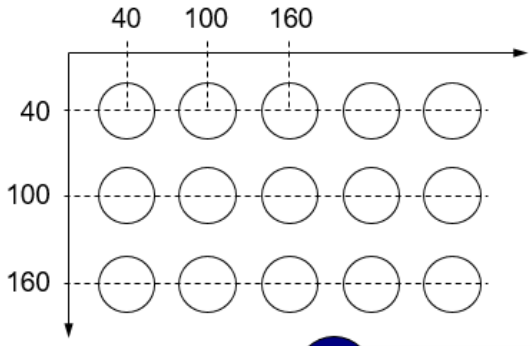
```
for i in range(5):
print("Привет!")
```

В результате работы программы слово «Привет!» будет напечатано 5 раз. Range – диапазон в переводе с английского. При этом переменная i по мере выполнения цикла будет принимать значения 0, 1, 2, 3, 4.

```
for i in range(3, 20):
    print(i)
```

При этом переменная i будет принимать значения **от 3 до 19**





1-й ряд:

? Что меняется для 2-го ряда?

```
x = 40
for i in range(5):
  circle(x, 40, 20)
  x += 60
```

y

! Можно сделать это процедурой с параметром y!

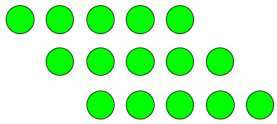
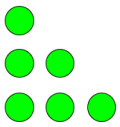
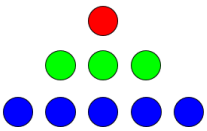
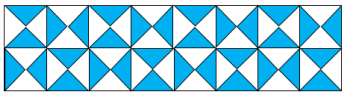
```
from graph import *
def row ( y ):
  x = 40
  for i in range(5):
    circle(x, y, 20)
    x += 60
y = 40
for k in range(3):
  row ( y )
  y += 60
run()
```

процедура

вызов процедуры

вниз на 60

Задание №1 Используя циклы и процедуры, нарисуйте узор. Число повторений рисунка N введите с клавиатуры.

	I вариант	II вариант
Оценка «3»	**«3»:** Ввести с клавиатуры число N и нарисовать N рядов по 5 кругов. **Пример** (N = 3): 	**«3»:** Ввести с клавиатуры число N и нарисовать N вертикальных рядов по 5 ромбиков. **Пример** (N = 2):
Оценка «4»	**«4»:** Ввести с клавиатуры число N и нарисовать из кругов прямоугольный размером N на N. **Пример** (N = 3): 	**«4»:** Используя циклы и процедуры, нарисуйте узор. Число повторений рисунка N введите с клавиатуры. **Пример** (N = 3):
Оценка «5»	**«5»:** Ввести с клавиатуры число N и нарисовать из кругов равнобедренный треугольник с высотой N. Каждый ряд должен быть покрашен в свой цвет. **Пример** (N = 3): 	**«5»:** Используя циклы и процедуры, нарисуйте узор. 

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
5. Список используемых источников
6. Выводы и предложения
7. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что представляет собой алгоритм повторения?
2. Какие виды циклов известны?
3. Когда целесообразно использовать циклы в графике?

Практическое занятие № 19. Построение графиков математических функций в Python

Цель занятия:

1. Овладеть операциями и методами построения графиков математических функций Python;

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

Изучить теоретическую часть

Выполнить задания

Задача: построить график функции $y = x^2$ на отрезке от -2 до 2.

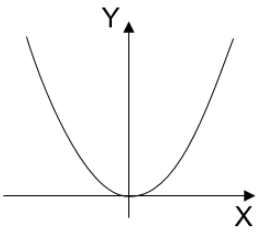
Анализ:

максимальное значение

$y_{max} = 4$ при $x = \pm 2$

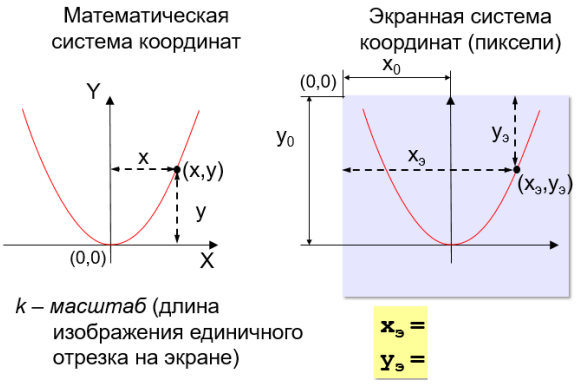
минимальное значение

$y_{min} = 0$ при $x = 0$



Проблема: функция задана в математической системе координат, строить надо на экране, указывая координаты в пикселях.

Преобразование координат

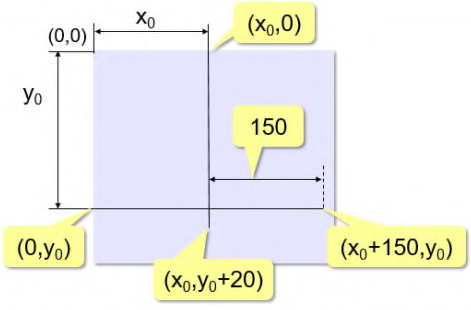


k – масштаб (длина изображения единичного отрезка на экране)

$x_э =$

$y_э =$

Оси координат




```
line(0, y0, x0+150, y0)
line(x0, 0, x0, y0+20)
```

Рисуем оси координат

```
from graph import *
x0 = 150 # начало координат
y0 = 250
k = 50   # масштаб
xmin = -2; xmax = 2 # пределы по x
line(0, y0, x0+150, y0)
line(x0, 0, x0, y0+20)
...
```

Строим по точкам

```
...
x = xmin  # начальное значение x
h = 0.02  # шаг изменения x
penColor("red")
while x <= xmax:
  y = x*x  # функция
  xe = x0 + k*x
  ye = y0 - k*y
  point(xe, ye) # точка на экране
  x += h   # к следующей точке
run()
```

экранные координаты (в пикселях)

Соединяем точки линиями

Идея: сначала создаём в памяти массив точек, затем соединяем точки линиями (polygon)

```
points = []  # пустой массив
while x <= xmax:
  y = x*x
  xe = x0 + k*x
  ye = y0 - k*y
  points.append( (xe, ye) )
  x += h
```

добавляем точку в массив

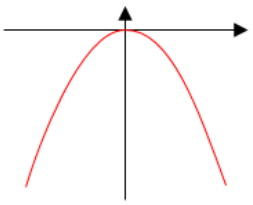
```
penColor("red")
polyline(points) # рисуем линию!
```

Задание №1 Построить графики функций:

Оценка «3»: Построить график функции

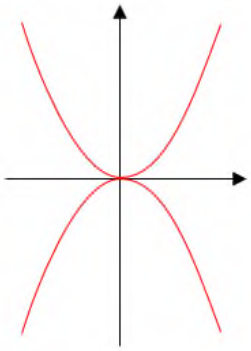
$$y = -x^2$$

на отрезке [-2,2].



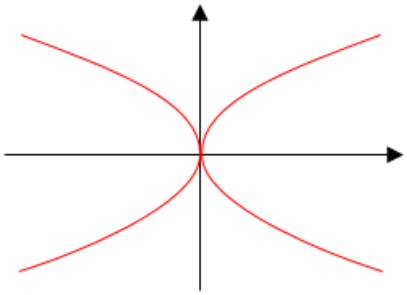
Оценка «4»

Построить графики функций $y = x^2$ и $y = -x^2$ на отрезке [-2,2].



Оценка «5»

«5»: Построить графики функций $x = y^2$ и $x = -y^2$ на отрезке [-2,2].



Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
5. Список используемых источников
6. Выводы и предложения
7. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Как построить экранную систему координат?

2. Как построить точки графика?
3. Как соединить построенные точки?

Практическое занятие № 20 Анимация в Python

Цель занятия:

1. Овладеть операциями и методами работы с анимацией в графике Python;

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

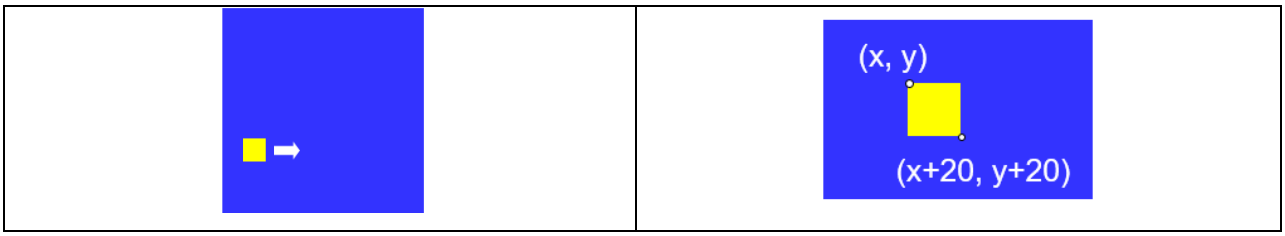
Изучить теоретическую часть

Выполнить задания

Анимация (англ. *animation*) – оживление изображения на экране.

Задача: внутри синего квадрата 400 на 400 пикселей слева направо двигается желтый квадрат 20 на 20 пикселей. Программа останавливается, если нажата клавиша *Esc* или квадрат дошел до границы синей области.

Привязка: состояние объекта задается координатами (x,y)

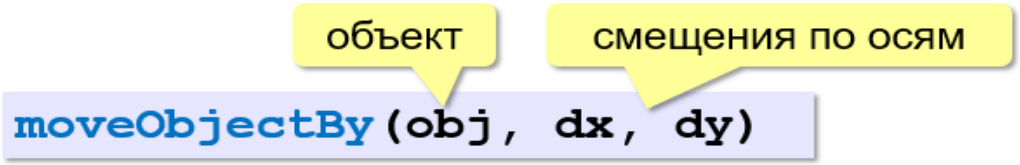


Принцип анимации:

1. рисуем объект в точке (x,y)
2. задержка на несколько миллисекунд

3. стираем объект
4. изменяем координаты (x,y)
5. переходим к шагу 1

В Python все фигуры, из которых состоит рисунок, – **объекты** (умеют перерисовывать себя сами)!



Начальная картинка

```
from graph import *
brushColor("blue")
rectangle(0, 0, 400, 400)
x = 100
y = 100
penColor("yellow")
brushColor("yellow")
obj = rectangle(x, y, x+20, y+20)
run()
```

синий квадрат; начальные координаты; жёлтый квадрат

Движение

```
def update():
  moveObjectBy(obj, 5, 0)
  if xCoord(obj) >= 380: # если вышел
    close()              # за границу
onTimer(update, 50)
```

x-координата; вызывать update каждые 50 мс

Выход по Escape

Событие (англ. *event*) – изменение состояния какого-то объекта в программе (нажатие на клавишу, щелчок мышью, перемещение или изменение размеров окна и т.п.).

```
def keyPressed(event):
  if event.keycode == VK_ESCAPE:
    close()  # закрыть окно
onKey(keyPressed)
```

обработчик события; код клавиши Esc = 27; вызывать при нажатии любой клавиши; установка обработчика события

Полная программа

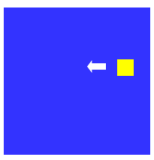
```
from graph import *
def update():
  ...
def keyPressed(event):
  ...
brushColor("blue")
rectangle(0, 0, 400, 400)
x = 100
y = 100
penColor("yellow")
brushColor("yellow")
obj = rectangle(x, y, x+20, y+20)
onKey(keyPressed)
onTimer(update, 50)
run()
```

процедуры; обработка событий

Задание №1:

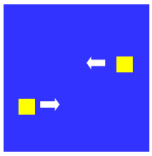
Оценка «3»

Квадрат двигается справа налево:



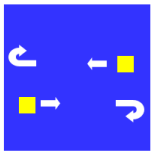
Оценка «4»

Два квадрата двигаются в противоположных направлениях:



Оценка «5»

«5»: Два квадрата двигаются в противоположных направлениях и отталкиваются от стенок синего квадрата:



Управление клавишами

Задача: жёлтый квадрат внутри синего квадрата управляется клавишами-стрелками. Коды клавиш:

влево – 37 вверх – 38 Esc – 27

вправо – 39 вниз – 40

VK_ESCAPE

Проблема: как изменять направление движения?

Обработчик события:

=37

=39

```
def keyPressed(event):
  if event.keycode == VK_LEFT:
    moveObjectBy(obj, -5, 0)
  elif event.keycode == VK_RIGHT:
    moveObjectBy(obj, 5, 0)
  ...  # дальше – сами...
onKey(keyPressed) # установить обработчик
```

VK_UP = 38
VK_DOWN = 40

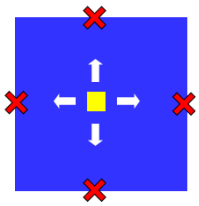
Задание №2:

Оценка «3»

«3»: Квадрат в самом начале стоит в правом нижнем углу, и двигается при нажатии стрелок только вверх или влево:

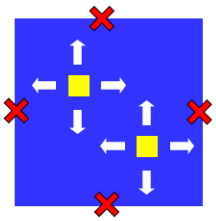
Оценка «4»

«4»: Квадрат двигается при нажатии стрелок, однако не может выйти за границы синего квадрата:



Оценка «5»

«5»: Два квадрата, один управляется стрелками, второй – любыми другими клавишами. Оба не могут выйти за границы синего поля.



Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
5. Список используемых источников
6. Выводы и предложения
7. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что называется анимацией?
2. Как можно имитировать движение объектов?
3. Как можно управлять движением с помощью клавиш?