Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»
Колледж СКФУ в г. Ставрополе

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к практическим занятиям

по (учебной) дисциплине	ОД. 08 Информатика
Специальность	27.02.07 Управление качеством продукции, процессов и услуг (по отраслям)
Форма обучения	очная

Ставрополь

Методические указания к практическим занятиям по дисциплине «Информатика» составлены в соответствии с требованиями ФГОС СПО и предназначены для студентов, обучающихся по специальности: 27.02.07 Управление качеством продукции, процессов и услуг (по отраслям).

Методические указания для учебной дисциплины разработаны:
Масаловой Л.С., преподаватель колледжа СКФУ в г. Ставрополе

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данные методические указания предназначаются для оказания помощи студентам в выполнении практических работ по учебной дисциплине «ОД. 08 Информатика».

Раздел 1. Раздел 1. Информация

Тема 1.1 Понятие информации

Цель: Научиться эффективно искать, анализировать, оценивать и использовать информацию из разных источников. Пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью, приобрести навыки, которые могут быть применены в различных областях профессиональной деятельности.

Теоретический материал:

Информация - это концепция, которая играет важную роль в современном мире. В нашем информационном обществе информация является ценным ресурсом и олицетворяет знания, факты и данные, которые передаются, хранятся и обрабатываются с целью получения пользы.

Важные аспекты понятия информации:

Знание и данные: Информация представляет собой сведения или данные, которые передаются между людьми или системами. Знания могут быть описательными (например, факты) или контекстными (понимание, как использовать факты).

Передача и обработка: Информация передается через различные каналы связи, такие как речь, текст, изображения и сигналы. Она также подвергается обработке, чтобы стать более полезной и понятной. Процесс передачи и обработки информации может включать в себя кодирование, декодирование, сжатие, фильтрацию и т.д.

Ценность и релевантность: Информация ценна в том случае, если она предоставляет новые знания, решает проблемы или помогает в принятии решений. Важно отличать ценную информацию от массы ненужных данных.

Качество и достоверность: Качество информации связано с её точностью, актуальностью и надежностью. Неверные или неточные данные могут привести к неправильным выводам и решениям.

Сохранение и доступ: Информация может храниться в различных форматах, таких как книги, базы данных, файлы и т.д. Важно иметь возможность быстро находить и получать доступ к необходимой информации.

Защита и безопасность: В современном мире защита информации от несанкционированного доступа и вредоносных действий становится все более важной задачей. Кибербезопасность играет ключевую роль в обеспечении безопасности информации.

Информационная грамотность: Эффективное использование информации требует навыков анализа, критического мышления и оценки её достоверности. Информационная грамотность позволяет правильно выбирать и использовать информацию.

Информационные системы: Информация может быть организована и структурирована в информационных системах. Это могут быть базы данных, компьютерные программы, сайты и другие средства, предназначенные для хранения, обработки и передачи информации.

Коммуникация: Информация передается через коммуникационные каналы, такие как разговоры, письма, электронные сообщения и социальные сети. Коммуникация играет ключевую роль в обмене знаний и опыта между людьми.

Информационные технологии: Современные информационные технологии играют огромную роль в обработке, передаче и хранении информации. Компьютеры, сети, мобильные

устройства и программное обеспечение облегчают доступ к информации и усиливают её потенциал.

Информационные источники: Информацию можно получать из разных источников, таких как книги, журналы, интернет, аудио и видео материалы, учебники и научные статьи. Важно уметь выбирать надежные источники, чтобы обогатить свои знания.

Информационное общество: Современное общество считается информационным, так как информация стала основой для развития, прогресса и инноваций. Информационное общество ставит перед нами новые вызовы и возможности, связанные с умением эффективно управлять информацией.

Информация охватывает широкий спектр областей, включая науку, образование, бизнес, культуру и многие другие. Важно понимать, как работает информация и как использовать её в своей повседневной жизни и профессиональной деятельности.

Понимание концепции информации важна, так как она охватывает множество областей, начиная от технических и компьютерных аспектов до анализа данных, коммуникации и решения проблем. Осознание ценности информации поможет вам более эффективно использовать знания в учебе и будущей профессиональной деятельности, а также сделает вас более информационно грамотными и уверенными в быстро меняющемся мире.

Тема 1.2. Представление информации, языки, кодирование

Цель: Познакомится с простейшими приемами шифрования и дешифрования текстовой информации.

Теоретический материал:

Предоставление информации: Предоставление информации означает передачу знаний, данных или иных сведений от одного источника к другому. Это может быть коммуникация между людьми, обмен данными между компьютерами, передача информации через средства массовой информации и так далее. Передача информации может осуществляться различными способами, такими как речь, письменный текст, графика, аудио и видео записи, электронные сообщения и другие.

Языки: Язык - это средство коммуникации, которое позволяет передавать информацию между людьми или системами. Языки могут быть разными: говоренными, письменными, программирования и т.д. Они обеспечивают структурированный набор символов и правил, которые используются для передачи и интерпретации информации. Язык состоит из символов (например, буквы и слова) и грамматических правил, которые определяют, как символы могут быть объединены для передачи конкретных значений.

Кодирование: Кодирование - это процесс преобразования информации из одной формы в другую с целью её передачи или хранения. В современном мире, особенно в контексте компьютерных систем, кодирование является важным аспектом передачи данных. Примерами кодирования могут быть:

- **Текстовое кодирование:** Преобразование символов (букв, цифр, специальных знаков) в числовые значения, которые компьютеры могут обрабатывать. Примерами являются ASCII и Unicode.
- **Аудио и видео кодирование:** Преобразование аналоговых аудио или видео сигналов в цифровой формат с использованием различных алгоритмов сжатия, таких как MP3, JPEG, MPEG и другие.
- **Кодирование данных:** Преобразование различных типов данных (например, числа, текст, изображения) в битовую последовательность, которая может быть передана по сети или сохранена на диске.

Кодирование играет важную роль в передаче информации, так как оно позволяет сжимать данные, уменьшая объем передаваемой информации, и обеспечивать их правильную интерпретацию на стороне получателя.

Понимание этих понятий важно для тех, кто работает с информацией и технологиями. В мире, где обмен информацией играет столь важную роль, знание языков и принципов кодирования помогает эффективно общаться и работать с данными в различных ситуациях.

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 (рисунокунков, 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– аудиокомпакт-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 присуноквоение каждому элементу конкретного значения в форме кода.

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

Кодирование 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 присуноквается 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 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.

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

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.

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 2^8=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

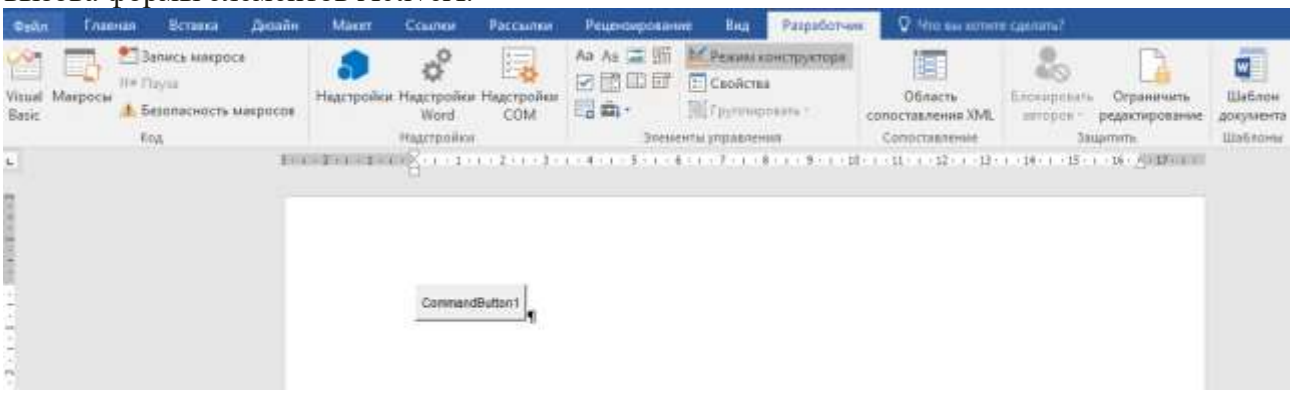
Векторное изображение представляет собой совокупность графических примитивов (точка, отрезок, эллипс…). Каждый примитив описывается математическими формулами. Кодирование зависит от прикладной среды.

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

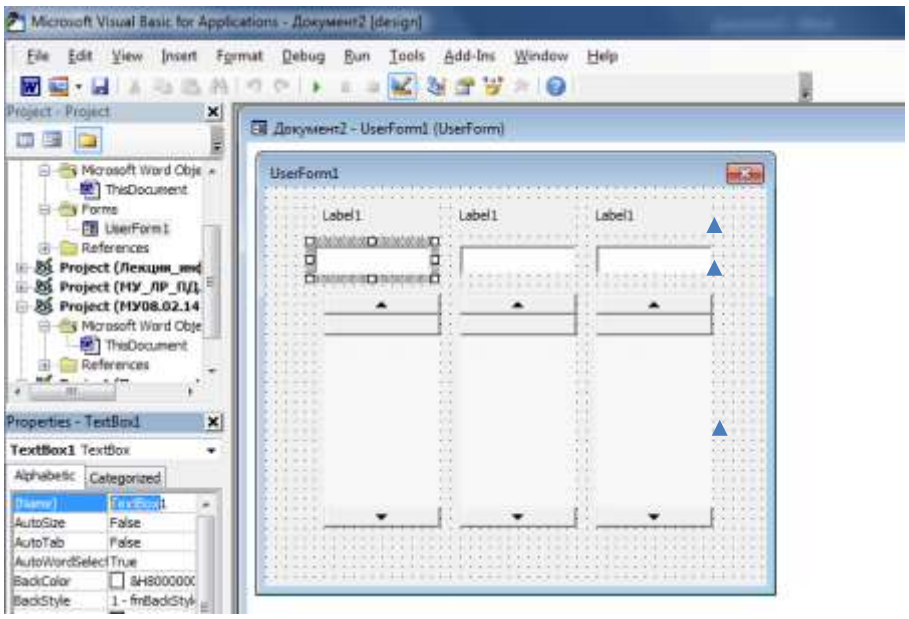
Важно также, что векторные графические изображения могут быть увеличены или уменьшены без потери качества.

Практический материал

Для создания и работы диалогового окна необходимо запустить текстовый или табличный процессор Microsoft word, excel перейти в режим разработки приложений, создать кнопку вызова формы элементов ActiveX.



Создать форму и разместить на ней обьекты Label 3 шт (несет информационную нагрузку), TextBox 3 шт (ввод и вывод текстовой и числовой информации), ScrollBar 3 шт. (ползунок).



Далее необходимо обработать каждый обьект.

```
Private Sub UserForm_Initialize() 'Инициализация формы
  TextRed = 255
  Label1.BackColor = RGB(255, 0, 0)
  TextGreen = 255
  Label2.BackColor = RGB(0, 255, 0)
  TextBlue = 255
  Label3.BackColor = RGB(0, 0, 255)
End Sub

Private Sub ScrollBarGreen_Change()
'Обработчик события, выполняемый при перемещении полосы прокрутки, отвечающей за зеленый цвет, пользователем
  UserFormColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
  TextGreen.Text = CStr(ScrollBarGreen)
  Sheets("Модель_rgb").CmdColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
End Sub

Private Sub ScrollBarRed_Change()
'Обработчик события, выполняемый при перемещении полосы прокрутки, отвечающей за красный цвет, пользователем
  UserFormColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
  TextRed.Text = CStr(ScrollBarRed)
  Sheets("Модель_rgb").CmdColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
End Sub

Private Sub ScrollBarBlue_Change()
'Обработчик события, выполняемый при перемещении полосы прокрутки, отвечающей за синий цвет, пользователем
  UserFormColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
  TextBlue.Text = CStr(ScrollBarBlue)
  Sheets("Модель_rgb").CmdColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
```

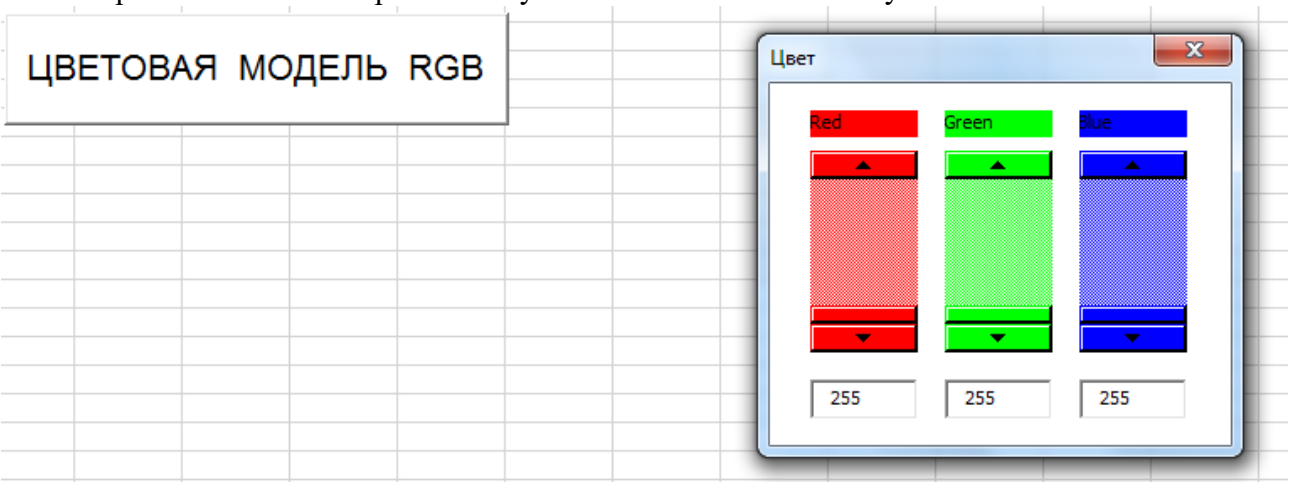
```
End Sub

Private Sub TextBlue_Change()
'Обработчик события, выполняемый при вводе значения в текстовое поле, отвечающего за синий цвет, пользователем
  Dim blue As Byte
  If Not IsNumeric(TextBlue) Then TextBlue = 0
  blue = TextBlue
  ScrollBarBlue.Value = blue
  UserFormColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
End Sub

Private Sub TextGreen_Change()
'Обработчик события, выполняемый при вводе значения в текстовое поле, отвечающего за зеленый цвет, пользователем
  Dim green As Byte
  If Not IsNumeric(TextGreen) Then TextGreen = 0
  green = TextGreen
  ScrollBarGreen.Value = green
  UserFormColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
End Sub

Private Sub TextRed_Change()
'Обработчик события, выполняемый при вводе значения в текстовое поле, отвечающего за красный цвет, пользователем
  Dim red As Byte
  If Not IsNumeric(TextRed) Then TextRed = 0
  red = TextRed
  ScrollBarRed.Value = red
  UserFormColor.BackColor = RGB(ScrollBarRed, ScrollBarGreen, ScrollBarBlue)
End Sub
```

Чтобы приложение начало работать нужно нажать F5 или кнопку RUN.



Тема 1.3. Измерение информации. Алфавитный подход. Содержательный подход

Цель: Закрепить знания о способах измерения информации при использовании содержательного и алфавитного подхода.

Теоретический материал

Измерение информации и алфавитный подход являются ключевыми понятиями в теории информации, разработанной Клодом Шенноном. Давайте рассмотрим их подробнее.

Измерение информации: В теории информации, измерение информации означает количественное определение степени неожиданности или неопределенности события. Информация можно рассматривать как меру того, насколько конкретное событие «изменяет» наше понимание текущей ситуации.

Алфавитный подход: Алфавитный подход - это метод оценки количества информации, связанного с определенными символами или событиями в заданном алфавите. Алфавитом может быть набор символов, букв, цифр или любых других дискретных элементов. Классическим примером алфавита является алфавит букв английского языка.

В рамках алфавитного подхода, количество информации, связанной с конкретным событием или символом, определяется следующим образом:

$$\mathbf{I(x) = -\log_2(P(x))}$$

Где:

- **I(x)** - количество информации, связанное с событием (символом) x.
- **P(x)** - вероятность наступления события x.

Чем менее вероятно наступление события, тем более информации оно несет. Например, если какое-то событие очень вероятно (P(x) близко к 1), то количество информации будет близко к нулю, так как это событие мало что добавляет к нашему пониманию ситуации. С другой стороны, если событие маловероятно (P(x) близко к 0), то количество информации будет высоким, так как это событие неожиданное и значимое.

Применение алфавитного подхода позволяет количественно измерить информацию и определить, насколько новое или удивительное событие. Этот подход широко используется в теории компьютерных наук, телекоммуникаций, статистике и других областях, связанных с обработкой информации.

Содержательный подход к измерению информации, также известный как «содержательная теория информации», является альтернативой алфавитному подходу Клода Шеннона. Этот подход был разработан Андреем Колмогоровым и Алексеем Чайшиным в конце 1960-х годов.

Содержательный подход не оценивает информацию в терминах вероятности и алфавита символов, а скорее сосредотачивается на структуре и организации данных. Суть этого подхода заключается в том, что информация связана с тем, какие знания и понимание о мире мы получаем благодаря данным. То есть, информация заключается в том, как новые данные изменяют или обогащают нашу модель мира.

В содержательной теории информации информация измеряется через концепцию алгоритмической сложности. Суть этой концепции заключается в следующем: количество информации, которое содержится в некоторых данных, можно оценить через длину наименьшего программного алгоритма, который может создать эти данные. Следовательно, информация здесь связана с уровнем сжатия данных, то есть с тем, насколько коротким может быть алгоритм для их создания.

Содержательный подход позволяет более точно оценивать информацию в случаях, когда вероятности событий неизвестны или когда данные имеют сложную структуру. Этот подход широко используется в областях, связанных с компрессией данных, алгоритмами сжатия и биоинформатикой.

Важно заметить, что и алфавитный подход Шеннона, и содержательный подход Колмогорова-Чайшина имеют свои применения и преимущества в разных контекстах. Оба подхода представляют разные способы подхода к измерению информации и нахождения её значимости в различных ситуациях.

Практический материал

Создать с помощью табличного процессора таблицу для автоматического перевода чисел из битов в байты, килобайты, мегабайты, гигабайты, терабайты.

Перевести во все предложенные единицы измерения 1000 битов, 8192 бита, 20500 битов, 16394 бита, 100200 битов.

Тема 1.4. Представление чисел в компьютере. Представление текста, изображения и звука в компьютере

Цель: изучить способы представления текстовой, графической, звуковой информации и видеоинформации, научиться записывать числа в различных системах счисления.

Теоретический материал:

Представление чисел в компьютере основано на использовании битовых последовательностей, которые представляют значения чисел в двоичной системе счисления. Давайте рассмотрим основные методы представления целых чисел, дробных чисел и отрицательных чисел в компьютерных системах.

Представление целых чисел:

Целые числа в компьютере обычно представляются в двоичной системе счисления. Каждая позиция в числе представляет определенную степень двойки. Например, число 1011 в двоичной системе равно 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 8 + 0 + 2 + 1 = 11.

Для представления отрицательных целых чисел обычно используется дополнительный код или обратный код. Дополнительный код получается инверсией всех битов в числе и добавлением 1. Этот метод позволяет сохранить арифметические операции (сложение, вычитание) для положительных и отрицательных чисел.

Представление дробных чисел:

Дробные числа могут быть представлены с использованием плавающей запятой. Этот метод представления позволяет работать с большим диапазоном значений, включая числа с очень большими или очень маленькими порядками.

Представление в плавающей запятой включает три компонента: знак, мантиссу и порядок. Мантисса представляет дробную часть числа, а порядок определяет местоположение запятой (степень двойки). Этот метод предоставляет возможность представления как очень больших, так и очень маленьких чисел, но также может вызвать проблемы точности из-за ограниченной длины мантиссы.

Представление отрицательных чисел:

Отрицательные числа часто представляются с использованием дополнительного кода. В дополнительном коде отрицательные числа представлены как инвертированные биты положительных чисел, плюс единица. Этот метод обеспечивает удобство при выполнении арифметических операций.

Представление целых чисел разной длины:

Целые числа могут быть представлены на разных битовых длинах, что влияет на диапазон представимых значений. Например, 8-битное целое число может представлять значения от -128 до 127 (включая ноль). 16-битное число имеет более широкий диапазон и так далее.

Представление символов и текста:

Символы и текст часто представляются с использованием кодировок, таких как ASCII, Unicode и UTF-8. В этих кодировках каждому символу сопоставляется числовое значение, что позволяет компьютеру работать с текстовой информацией.

Важно понимать, как числа представляются в компьютере, чтобы эффективно использовать и обрабатывать данные в программировании и других областях информационных технологий.

Представление текста, изображения и звука в компьютере основывается на специфических форматах и кодировках, которые позволяют компьютерам эффективно хранить, передавать и обрабатывать разнообразные типы данных. Давайте рассмотрим каждый тип данных по отдельности:

Представление текста:

Текст обычно представляется с использованием кодировок символов. Вот несколько распространенных кодировок:

- **ASCII (American Standard Code for Information Interchange):** Это одна из старейших кодировок, где каждому символу английского алфавита сопоставляется 7-битный код (от 0 до 127).
- **Unicode:** Это стандарт, который позволяет представлять символы практически всех языков мира, а также различные символы и знаки препинания. В Unicode используются различные форматы кодирования, такие как UTF-8, UTF-16 и UTF-32.
- **UTF-8 (Unicode Transformation Format - 8-bit):** Это переменной длины кодировка Unicode, где символы могут быть представлены разным количеством байт. Она широко используется в веб-разработке и текстовых файлах.

Представление изображений:

Изображения обычно представляются в растровом (пиксельном) или векторном форматах.

- **Растровые изображения:** В растровых изображениях каждый пиксель имеет свой цвет и яркость. Изображение разбивается на сетку пикселей, и каждый пиксель хранит информацию о своем цвете. Форматы JPEG, PNG и BMP - это примеры растровых форматов.
- **Векторные изображения:** Векторные изображения описываются математическими объектами, такими как линии, кривые и формы. Они могут быть масштабированы без потери качества. Форматы SVG (Scalable Vector Graphics) и AI (Adobe Illustrator) - это примеры векторных форматов.

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:

BMP
GIF
JPEG
TIFF
PNG

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 рисунокунков (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.

Представление звука:

Звук может быть представлен как аналоговый сигнал, так и цифровым аудио. Аудио обычно представляется в виде цифровых сигналов, где звук разбивается на небольшие временные интервалы, называемые «звуковыми выборками» или «семплами». Примерами форматов для цифрового аудио являются WAV, MP3, AAC и FLAC.

Для каждого типа данных существует множество форматов и методов сжатия, которые оптимизируют размер и качество данных в зависимости от потребностей. Понимание этих форматов поможет вам эффективно работать с разнообразными типами информации в компьютерных системах.

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.

Каждому уровню громкости присуноквaивается 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 $N = 2^{16} = 65536$.

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.

Практический материал:

Задание 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
200	194	192	205	206	194		192	208	210	197	204

Выполнение задания 1

Задание 2.

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.

Выполнение задания 2

Задание 3. Ответить на вопросы:

Что такое информация?

Перечислить свойства информации.

Какие виды информации Вы знаете?

Приведите примеры аналогового представления графической информации.

Что такое пиксель?

Что такое система счисления?

Напишите правило перевода десятичных чисел в двоичный код.

Перечислите единицы измерения информации.

Задание 4. Сделать вывод о проделанной лабораторной работе:

Раздел 2. Информационные процессы

Тема 2.1. Хранение информации. Передача информации

Цель: Научиться обрабатывать данные с целью получения нужной информации, включая в себя вычисления, анализ, сортировку, фильтрацию и другие операции.

Теоретический материал

Информационные процессы служат основой для работы компьютеров, программ и систем, обеспечивая обработку данных, преобразование информации и удовлетворение потребностей пользователей.

Информационные процессы представляют собой последовательность действий и операций, связанных с сбором, обработкой, передачей и использованием информации. Эти процессы играют ключевую роль в современном обществе, так как обеспечивают обмен знаний, принятие решений, коммуникацию и многое другое. Давайте рассмотрим основные этапы информационных процессов:

Сбор информации: Этот этап включает в себя получение данных из различных источников. Информация может быть собрана с помощью наблюдений, исследований, опросов, датчиков и других методов. Важно выбрать правильные источники и методы сбора, чтобы обеспечить точность и полноту данных.

Обработка информации: На этом этапе данные анализируются, структурируются и преобразуются в понятный формат. Обработка может включать в себя сортировку, фильтрацию, агрегацию, анализ и другие операции. Цель обработки - получить из данных полезные знания и выводы.

Передача информации: Этот этап связан с передачей обработанных данных от одного источника к другому. Коммуникационные каналы могут включать в себя интернет, сети, радиоволны, кабели и другие средства передачи информации. Важно обеспечить надежность, конфиденциальность и целостность передаваемых данных.

Хранение информации: После сбора и обработки информации её можно сохранить для будущего использования. Хранение может осуществляться в базах данных, файловых системах, архивах и других хранилищах. Важно обеспечить доступность и защиту хранимых данных.

Использование информации: Завершающий этап - это использование полученных данных для принятия решений, решения задач, получения знаний и обмена информацией. Информация может быть использована в научных исследованиях, бизнесе, образовании, медиа и других областях.

Информационные процессы охватывают широкий спектр деятельности в современном обществе. Отправные точки для этих процессов могут варьироваться в зависимости от контекста: от получения данных от датчиков в промышленности до обмена сообщениями на социальных сетях. Понимание этих процессов позволяет эффективно управлять информацией и использовать её в разных аспектах жизни и работы.

Хранение информации

Хранение информации - это процесс сохранения данных таким образом, чтобы они могли быть доступны в будущем. Это важный аспект в современной информационной технологии, так как организации и люди собирают и создают огромные объемы данных, которые требуется эффективно управлять и сохранять.

Хранилища данных: Для хранения информации используются различные типы хранилищ, такие как жесткие диски, SSD (Solid State Drives), серверы, облака данных, флеш-

накопители и другие устройства. Каждый тип хранилища имеет свои особенности, включая емкость, скорость доступа, надежность и стоимость.

Организация данных: Данные могут быть организованы в различных форматах и структурах в зависимости от потребностей. Это может включать в себя создание баз данных, файловых систем, таблиц, документов и других структур для хранения данных.

Резервное копирование и восстановление: Для обеспечения надежности данных важно регулярно создавать резервные копии. Это позволяет восстановить информацию в случае потери данных из-за технических сбоев, ошибок или вредоносных атак.

Обеспечение доступности: Информация должна быть доступной для тех, кто имеет к ней права доступа. Обеспечение безопасности и контроля доступа к данным помогает предотвратить несанкционированный доступ и утечки информации.

Масштабируемость: С ростом объема данных необходимо обеспечивать масштабируемость хранилищ. Это может быть реализовано с помощью кластеризации, облачных хранилищ и других методов.

Сжатие и оптимизация: Для экономии места и ускорения доступа к данным можно использовать методы сжатия данных и оптимизации структур хранения.

Долгосрочное хранение: В некоторых случаях данные требуется хранить на долгий срок в соответствии с законодательством или требованиями организации. Долгосрочное хранение включает в себя сохранение данных в архивах или специальных системах.

Утилизация и удаление данных: Устаревшие или неактуальные данные должны быть утилизированы или удалены в соответствии с политиками безопасности и законодательством.

Эффективное хранение информации включает в себя баланс между доступностью, надежностью, безопасностью и экономичностью. В зависимости от конкретных потребностей, методы хранения и организации данных могут различаться.

Передача информации

Передача информации - это процесс перемещения данных из одного места или источника в другое. Этот процесс играет ключевую роль в современном мире, обеспечивая коммуникацию между людьми, устройствами и системами. Передача информации может происходить по различным каналам и с использованием разнообразных технологий.

Каналы связи: Канал связи - это физический или логический путь, по которому данные передаются. Каналы могут быть проводными (например, кабели) или беспроводными (например, радиоволны). Примеры каналов связи включают интернет, сети мобильной связи, локальные сети и другие.

Средства передачи: Средства передачи - это устройства и технологии, которые обеспечивают передачу данных по каналам связи. Это могут быть модемы, маршрутизаторы, сотовые телефоны, спутники и многое другое.

Протоколы связи: Протоколы связи определяют правила и форматы передачи данных между устройствами. Протоколы обеспечивают структуру сообщений, проверку ошибок, управление потоком данных и другие аспекты коммуникации.

Безопасность передачи: Обеспечение безопасности данных во время передачи - важный аспект. Шифрование, аутентификация и другие методы используются для защиты данных от несанкционированного доступа и атак.

Скорость передачи: Скорость передачи данных может различаться в зависимости от технологии и канала связи. Быстрая передача данных важна для обеспечения эффективной коммуникации и обработки информации.

Типы данных: Информация может быть передана в различных форматах: текст, аудио, видео, изображения и другие. Каждый тип данных может требовать своих методов передачи и обработки.

Протоколы интернета: Протоколы интернета, такие как TCP/IP (Transmission Control Protocol / Internet Protocol), обеспечивают структуру и организацию передачи данных в сетях интернета.

Беспроводная связь: Технологии беспроводной связи, такие как Wi-Fi, Bluetooth и мобильные сети, позволяют передавать данные без использования физических кабелей.

Передача информации стала неотъемлемой частью нашей повседневной жизни и бизнеса. Технологии коммуникации позволяют нам обмениваться данными, взаимодействовать и получать информацию из разных источников, делая мир более связанным и информационно насыщенным.

Тема 2.2. Обработка информации и алгоритмы

Цель: Закрепить навыки программного управления учебными исполнителями алгоритмов, полученных при изучении курса информатики.

Обработка информации и алгоритмы играют важную роль в области информатики и компьютерных наук. Эти понятия связаны с методами и процессами преобразования данных из исходного состояния в целевое состояние с использованием определенных инструкций и логики. Давайте рассмотрим каждый из этих аспектов подробнее:

Обработка информации:

Обработка информации - это процесс изменения, анализа и преобразования данных, чтобы извлечь из них знания, выводы или другую ценную информацию. В компьютерной науке обработка информации включает в себя множество действий, таких как сортировка, фильтрация, агрегация, вычисления и многое другое. Важно понимать, что обработка информации может применяться как к текстовым данным, так и к аудио, видео, изображениям и другим типам данных.

Алгоритмы:

Алгоритм - это последовательность шагов или инструкций, предназначенных для выполнения определенной задачи или решения конкретной проблемы. Алгоритмы обычно описываются на различных языках программирования или в виде псевдокода. Они используются для формализации и автоматизации процессов обработки информации.

Важные аспекты алгоритмов:

- **Корректность:** Алгоритм должен решать задачу верно и давать правильные результаты в соответствии с поставленной задачей.
- **Эффективность:** Алгоритм должен быть эффективным в смысле времени и ресурсов, не занимать слишком много времени и не требовать слишком большой памяти.
- **Понятность:** Алгоритм должен быть понятным и легко интерпретируемым для того, кто будет его реализовывать.
- **Универсальность:** Алгоритмы могут применяться в разных сферах, поэтому они должны быть абстрагированы от конкретных контекстов.

Примеры алгоритмов включают в себя сортировку массивов, поиск данных, вычисление чисел Фибоначчи, алгоритмы пути в графах и многое другое. Алгоритмы являются фундаментальным элементом программирования и компьютерных наук, помогая автоматизировать сложные задачи и решать проблемы эффективно.

Тема 2.3. Автоматическая обработка информации

Цель: Ознакомиться с основами теории алгоритмов на примерах решения задач на программное управление алгоритмической машиной Поста.

Автоматическая обработка информации - это процесс, при котором данные обрабатываются компьютерами или программами без необходимости вмешательства человека. Это понятие тесно связано с автоматизацией и использованием компьютерных систем для выполнения различных операций обработки данных.

1. Автоматизация процессов: Автоматическая обработка информации позволяет автоматизировать рутинные, повторяющиеся или сложные задачи, которые ранее могли быть выполнены только вручную. Это увеличивает эффективность и точность выполнения задач.

2. Скорость и эффективность: Компьютеры могут обрабатывать большие объемы данных намного быстрее, чем человек. Это позволяет получить результаты и выводы в кратчайшие сроки.

3. Автоматические решения: На основе заранее заданных алгоритмов и правил, компьютеры могут принимать решения и выполнять действия без непосредственного участия человека. Примером может быть автоматическое управление производственными процессами или автоматическая классификация данных.

4. Минимизация ошибок: Автоматическая обработка информации помогает минимизировать человеческие ошибки, которые могут возникнуть при ручной обработке данных.

5. Обработка больших данных: В условиях современного информационного общества объемы данных становятся все больше. Автоматическая обработка позволяет эффективно управлять и анализировать такие большие объемы информации.

6. Повторяемость: Автоматические процессы обработки информации могут быть легко повторены многократно с одними и теми же результатами. Это особенно полезно в задачах тестирования программ или обработки данных.

Примеры автоматической обработки информации включают в себя обработку заказов в электронной коммерции, автоматизированное управление инфраструктурой в области умных городов, анализ больших данных с использованием машинного обучения и многое другое. Автоматическая обработка информации является важным инструментом для оптимизации и улучшения различных процессов в современном мире.

Практический материал

С помощью виртуальной алгоритмической машины Поста используя команды программного продукта играть в крести нолики.

Команда	Действие
n▲ m	Сдвиг каретки на шаг влево и переход к выполнению команды с номером m
n▼ m	Сдвиг каретки на шаг вправо и переход к выполнению команды с номером m
n V m	Установка метки в текущую пустую клетку
n ↕ m	Стирание метки в текущей клетке
n !	Остановка выполнения программы
n ? m, k	Переход по содержимому текущей клетки: если пустая, то выполняется команда m, если метка стоит то выполняется команда k

Тема 2.4. Информационные процессы в компьютере

Цель: Научиться обрабатывать информацию. Изучить архитектуру персонального компьютера, способы представления и хранения информации в нем, системы счисления, используемые в компьютере.

Теоретический материал

Информационные процессы в компьютере представляют собой последовательность действий и операций, которые выполняются внутри компьютерной системы для обработки данных и получения нужной информации. Эти процессы являются основой работы компьютера и включают в себя множество этапов, от ввода данных до получения результата.

1. Ввод данных: Информационный процесс начинается с ввода данных в компьютер. Это может быть ввод с клавиатуры, мыши, сканера, микрофона, камеры и других устройств ввода. В этот момент данные переходят из внешнего мира внутрь компьютерной системы.

2. Обработка данных: После ввода данных компьютер начинает их обработку. Обработка включает в себя применение алгоритмов, операций и функций к входным данным. Это может включать в себя вычисления, сортировку, фильтрацию, агрегацию и другие операции в зависимости от цели обработки.

3. Хранение данных: В процессе обработки данных часто требуется их временное хранение в памяти компьютера. Память компьютера может быть оперативной (RAM), где данные хранятся во время работы, или постоянной (например, жесткий диск), где данные сохраняются даже после выключения компьютера.

4. Вывод данных: После обработки и, возможно, хранения данных, результаты выводятся на устройства вывода, такие как монитор, принтер, аудиоустройства и другие. Это переводит информацию обратно в визуальный, аудио или другой формат для восприятия человеком.

5. Взаимодействие с пользователем: В процессе информационных процессов важное значение имеет взаимодействие с пользователем. Это может быть в виде запросов на ввод данных, отображения результатов, обработки команд пользователя и других действий, связанных с интерфейсом.

6. Управление и управляемость: Компьютерные системы управляют информационными процессами с помощью операционных систем и программного обеспечения. Управление процессами включает в себя распределение ресурсов, планирование задач, обеспечение безопасности и другие аспекты.

7. Цикличность: Информационные процессы могут повторяться в цикле, в зависимости от потребностей. Это позволяет компьютерам выполнять однотипные задачи многократно, автоматизируя рутинные операции.

Информационные процессы в компьютере обеспечивают обработку данных, преобразование информации и решение разнообразных задач. Понимание этой структуры помогает разработчикам и пользователям эффективно работать с компьютерными системами.

Архитектурой компьютера называется ее логическая организация, структура и ресурсы, которые может использовать программист.

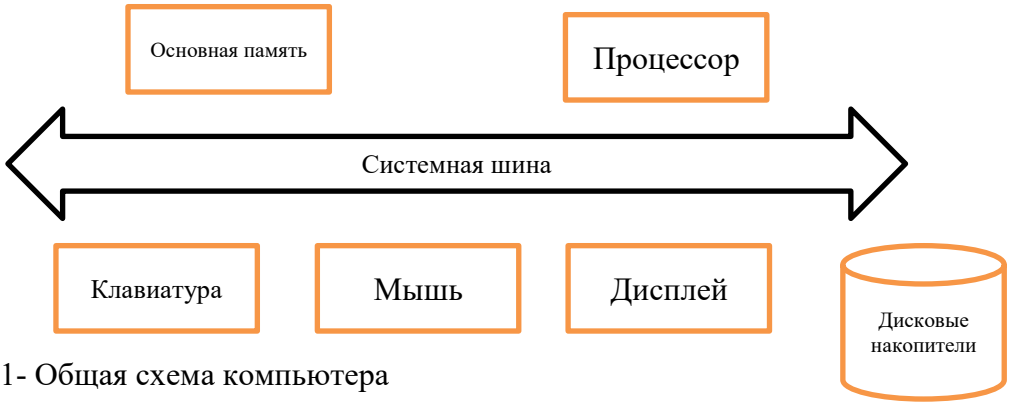


Рисунок 1- Общая схема компьютера

Персональный компьютер состоит из следующих блоков (рисунок 1):

- процессор;
- основная память;
- дисковые накопители;
- периферийные устройства.

Все блоки связаны между собой системной шиной (магистралью).

Основу компьютеров образует аппаратура (HardWare), построенная с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (SoftWare).

Компьютерная программа представляет собой последовательность отдельных команд. Команда - это описание операции, которую должен выполнить компьютер. Как правило, у

команды есть свой *код* (условное обозначение), *исходные данные* (операнды) и *результат*. Например, у команды «*сложить два числа*» операндами являются слагаемые, а результатом — их сумма. Совокупность команд, выполняемых данным компьютером, называется *системой команд* этого компьютера.

Функции памяти:

- приём информации из других устройств;
- запоминание информации;
- выдача информации по запросу в другие устройства машины.
- Функции процессора:
- обработка данных по заданной программе путем выполнения арифметических и логических операций;
- программное управление работой устройств компьютера.

Та часть процессора, которая выполняет команды, называется арифметико-логическим устройством (АЛУ), а другая его часть, выполняющая функции управления устройствами, называется устройством управления (УУ). Обычно эти два устройства выделяются чисто условно, конструктивно они не разделены.

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых *регистрами***.** Регистр выполняет функцию кратковременного хранения числа или команды. Над содержимым некоторых регистров специальные электронные схемы могут выполнять некоторые манипуляции. Основным элементом регистра является электронная схема, называемая триггером, которая способна хранить одну двоичную цифру (разряд двоичного кода). *Регистр* представляет собой совокупность триггеров, связанных друг с другом определённым образом общей системой управления.

Основная память (ОП). Она предназначена для хранения и оперативного обмена информацией с прочими блоками машины. ОП содержит два вида запоминающих устройств: постоянное запоминающее устройство (ПЗУ) и оперативное запоминающее устройство (ОЗУ).

ПЗУ служит для хранения неизменяемой (постоянной) программной и справочной информации, позволяет оперативно только считывать хранящуюся в нем информацию (изменить информацию в ПЗУ нельзя).

ОЗУ предназначено для оперативной записи, хранения и считывания информации (программ и данных), непосредственно участвующей в информационно-вычислительном процессе, выполняемом ПК в текущий период времени. Главными достоинствами оперативной памяти являются ее высокое быстродействие и возможность обращения к каждой ячейке памяти отдельно (прямой адресный доступ к ячейке). В качестве недостатка ОЗУ следует отметить невозможность сохранения информации в ней после выключения питания машины.

Внешняя память. Она относится к внешним устройствам ПК и используется для долговременного хранения любой информации, которая может когда-либо потребоваться для решения задач. Внешняя память содержит разнообразные виды запоминающих устройств, но наиболее распространенными, имеющимися практически на любом компьютере, являются накопители на жестких (НЖМД) и гибких (НГМД) магнитных дисках. В качестве устройств внешней памяти используются также запоминающие устройства на кассетной магнитной ленте (стриммеры), накопители на оптических дисках CD-R, CD-RW, DVD-R, DVR-RW.

Все многообразие данных, которое обрабатывает ПК, складывается из элементарных «кирпичиков» - цифр 0 и 1 (битов). Далее, в порядке укрупнения, возникает цепочка: ***«бит – байт – файл – каталог – логический диск»***.

Компьютер может обрабатывать информацию, представленную только в числовом виде. Вся другая информация (звуки, изображения, показания приборов) должна быть преобразована в числовую форму, закодирована. Все буквы и цифры закодированы восьмизначным кодом. Например: буква А имеет код 00110111, а когда на устройство поступает этот код, на мониторе высвечивается буква А.

Единицей информации в компьютере является *один бит*, т.е. двоичный разряд, который принимает значение 0 или 1. Однако, компьютер редко работает с конкретными битами в отдельности. Поэтому, слегка упрощая, будем утверждать, что наименьшая единица информации, с которой работает компьютер – это совокупность из восьми битов, воспринимаемая компьютером как единое целое. Эта комбинация битов называется ***байтом*.** Байт можно трактовать иначе: как символ, как ячейку памяти, как единицу измерения оперативной и внешней памяти.

Каждый байт имеет свой номер, который называют адресом (нумерация с нуля). **Биты** в байте нумеруются справа-налево от 0 до 7. Биты с меньшими номерами называются младшими, с большими номерами - старшими. Минимальная адресуемая ячейка - байт.

Количество значений, которые могут быть записаны в байт равно 2^8=256 (количество значений равно 2 в степени количества разрядов).

Слово - 2 байта. В слове помещается 2^{16}=65536 значений.

Двойное слово - 4 байта 32 разряда 2^{32}=8589934592 значений.

В вычислительной технике существуют разные способы представления чисел:.

- Двоичные (binary) числа
- Шестнадцатеричные (hexadecimal)
- Десятичные (decimal) числа
- Восьмеричные (octal) числа

В таблице 1 представлены десятичные числа от 0 до 15 и их эквиваленты в двоичной системе счисления.

Таблица 1. Представление десятичных чисел в двоичном коде

10-й код	**2-й код**	**10-й код**	**2-й код**	**10-й код**	**2-й код**
0	0000	5	0101	10	1010
1	0001	6	0110	11	1011
2	0010	7	0111	12	1100
3	0011	8	1000	13	1101
4	0100	9	1001	14	1110
				15	1111

Из таблицы видно, что для представления десятичных чисел от 0 до 15 требуется четырехразрядные двоичные коды. Для представления числа 16 необходимо добавить пятый разряд. Двоичные числа, представленные в таблице 1, характеризуют прямой код. Существуют и другие коды, с помощью которых упрощаются арифметические операции. К ним относятся обратный и дополнительный коды.

Двоичное число в обратном коде получается из числа в прямом коде путем инвертирования каждой цифры. Например, имеем прямой код числа 1101. Обратный код имеет вид 0010. Дополнительный код получают из обратного путем прибавления 1. Например, дополнительный код числа 1101 равен 0011.

Для стандартного **представления нечисловых данных** используется код ASCII (American National Standard Code for Information Interchange — Американский национальный стандартный код для обмена информацией), представленный на рисунокунке 2.

Рисунок 2 - таблица ASCII кода

На каждый символ отводится 1 байт. Первые 128 кодов отведены под управляющие символы, большие и малые буквы английского алфавита, цифры, знаки препинания и специальные символы. Остальные коды отводятся для национального алфавита (для русских букв коды от 128 до 255). Для представления символов также используется код Unicode, в котором для кодировки каждого символа используется 2 байта. Коды 0-128 используются как ASCII-коды, далее идут коды национальных письменностей.

Пример: ASCII-код символа A (лат)= 65_{10} =41_{16}= 01000111_2

Продолжая разбирать цепочку: ***«бит – байт – файл – каталог – логический диск»,*** можно сказать, что информация, представленная наборами байтов, называется файлом.

Файл - это именованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и др.

Файловая система — это средство для организации хранения файлов на каком-либо носителе. Файлы физически реализуются как участки памяти на внешних носителях — магнитных дисках или CD-ROM. Каждый файл занимает некоторое количество блоков дисковой памяти. Обычная длина блока — 512 байт. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска в специальных таблицах размещения файлов (FAT-таблицах (File Allocation Table)).

Архивный файл — это специальным образом организованный файл, содержащий в себе один или несколько файлов в сжатом или несжатом виде и служебную информацию об именах файлов, дате и времени их создания или модификации, размерах и т.п.

Степень сжатия файлов характеризуется коэффициентом Кс, определяемым как отношение объема сжатого файла Vc к объему исходного файла Vo, выраженное в процентах: Кс=(Vc/ V0)*100%

Архивация (упаковка) — помещение (загрузка) исходных файлов в архивный файл в сжатом или несжатом виде.

Разархивация (распаковка) — процесс восстановления файлов из архива точно в таком виде, какой они имели до загрузки в архив. Программы, осуществляющие упаковку и распаковку файлов, называются программами-архиваторами.

Практический материал:

Задание 1.

Выполнить работы по переводу чисел из одной системы счисления в другие, по кодированию и дешифрованию текста с использованием таблицы ASCII-кодов.

Задание 2.

- Составить краткое описание файловой системы и структуры жестких дисков ПК.
- Занести в таблицу сведения о клавиатуре, мыши, процессоре, оперативной памяти, дисплее своего компьютера.
- Открыть окно *Мой компьютер* и Создать папку Lab3_Inform на диске D в папке Basket.
- Создать файлы с расширением bmp, jpg, txt, doc, xls.
- Занести в созданную папку файлы с расширением bmp, jpg, txt, doc, xls.
- Выполнить операцию архивации созданных файлов.
- Сравнить степень сжатия файлов и занести в таблицу.

Выполнение заданий

1. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.Сделать проверку путем обратных преобразований.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 0011010111010110

- Закодировать слово ИНФОРМАТИЗАЦИЯ .

- Дешифровать данный текст: 8A AE AC AF EC EE E2 A5 E0

2. Перевести число 357 из 8-й системы счисления в 10-ю, 2-ю, 16-ю системы счисления. Сделать проверку путем обратных преобразований.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 0010101011101011

- Закодировать МИКРОПРОЦЕССОР .

- Дешифровать данный текст: 50 72 6F 67 72 61 6D

3. Перевести число 2BE из 16-й системы счисления в 10-ю, 8-ю, 2-ю системы счисления. Сделать проверку путем обратных преобразований.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 1001010101110101

- Закодировать слово ЗАДАЧА

-Дешифровать данный текст: 50 72 6F 63 65 64 75 72 65

4. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12. Сделать проверку путем обратных преобразований.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 01000110111101

-Закодировать слово COMPUTER .

-Дешифровать данный текст: 88 AD E4 AE E0 AC A0 E2 A8 AA A0

5. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12. Сделать проверку путем обратных преобразований.

-Перевести число 421 из 8-й системы счисления в 10-ю, 2-ю, 16-ю системы счисления.

-Дешифровать данный текст: 50 72 6F 67 72 61 6D

6. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Перевести число 3CC из 16-й системы счисления в 10-ю, 8-ю, 2-ю системы счисления.

-Дешифровать текст:43 6F 6D 70 75 74 65 72 20 49 42 4D 20 50 43

7. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 010001101111

-Закодировать слово ИНФОРМАЦИЯ .

-Дешифровать данный текст: 50 52 49 4E 54

8. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 11011110111

-Закодировать слово ПРИНТЕР .

-Дешифровать данный текст: 4C 61 73 65 72

9. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 00011011110

-Закодировать слово СКАНЕР .

-Дешифровать данный текст: 88 AD E4 AE E0 AC A0 E2 A8 AA A0

10. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 010010111

-Закодировать слово IBM PC .

-Дешифровать данный текст: 42 69 6E 61 72 79

11. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 1001001001

-Закодировать слово PENTIUM .

-Дешифровать данный текст: 49 6E 66 6F 72 6D 61 74 69 6F 6E

12. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 11011110111

-Закодировать слово COMPUTER .

-Дешифровать данный текст:.

91 A8 E1 E2 A5 AC A0 20 E1 E7 A8 E1 AB A5 AD A8 EF

13. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 01000110

-Закодировать слово TOTAL COMMANDER .

-Дешифровать данный текст: AC AE A4 A5 AB A8 E0 AE A2 A0 A2 A0 AD A8 A5

14. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 010011110111

-Закодировать слово СИСТЕМА .

-Дешифровать данный текст: 42 69 6E 61 72 79

15. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 111011110111

-Закодировать слово MICROSOFT .

-Дешифровать текст: 43 6F 6D 70 75 74 65 72 20 49 42 4D 20 50 43

16. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 010011110111

-Закодировать слово INTERNET .

-Дешифровать данный текст: 8A AE AC AF EC EE E2 A5 E0

17. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 010011110111

-Закодировать слово ВИНЧЕСТЕР.

-Дешифровать данный текст: 50 72 6F 67 72 61 6D

18. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 010011110111

-Закодировать слово АЛГОРИТМ .

-Дешифровать данный текст: 50 72 6F 63 65 64 75 72 65.

19. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 010011110111

-Закодировать слово ИНФОРМАТИКА.

-Дешифровать данный текст: 84 88 91 8A 8E 82 8E 84

20. Перевести десятичное число, полученное из дня рождения плюс месяц рождения, в 2-ю, 8-ю, 16-ю системы счисления. Если исходное число меньше 10, прибавить 12.

-Записать в десятичной системе счисления целое число, если дан его дополнительный код: 01000011111

-Закодировать слово МОНИТОР.

-Дешифровать данный текст: 42 69 6E 61 72 79

Раздел 3. Программирование обработки информации

Программирование обработки информации - это процесс создания программ и алгоритмов, которые позволяют компьютеру обрабатывать данные, преобразовывать их и получать нужную информацию. Программирование играет ключевую роль в автоматизации информационных процессов и реализации различных задач обработки данных.

1. Выбор языка программирования: Первый шаг в программировании обработки информации - это выбор подходящего языка программирования. Разные языки предоставляют разные инструменты и возможности для работы с данными и алгоритмами.

2. Разработка алгоритмов: Прежде чем начать писать код, необходимо разработать алгоритмы, которые описывают последовательность шагов для обработки данных. Это важный этап, так как алгоритмы определяют логику работы программы.

3. Структурирование кода: Структурирование кода - это организация программы на более мелкие блоки, функции или модули. Это делается для повышения читаемости кода, улучшения его поддерживаемости и переиспользуемости.

4. Ввод и вывод данных: В программировании обработки информации важно реализовать механизмы ввода данных (через консоль, файлы, сеть и т.д.) и вывода результатов (на экран, в файлы и другие устройства).

5. Работа с данными: Программы обработки информации часто работают с различными типами данных, такими как числа, строки, структуры данных и т.д. Важно правильно выбирать и обрабатывать данные в соответствии с поставленными задачами.

6. Применение алгоритмов обработки: Реализация алгоритмов обработки данных, таких как сортировка, поиск, фильтрация и другие, является ключевым аспектом программирования обработки информации.

7. Обработка ошибок: Программы должны быть устойчивы к ошибкам и некорректным данным. Реализация обработки и проверки ошибок помогает предотвратить сбои в работе программы.

8. Оптимизация и производительность: В программировании обработки информации важно учитывать производительность программы, особенно при работе с большими объемами данных. Оптимизация кода может улучшить скорость работы программы.

9. Тестирование: После написания программы необходимо провести тестирование, чтобы убедиться, что она работает корректно и выдает правильные результаты в различных сценариях.

Программирование обработки информации является центральной частью компьютерных наук и информатики. Оно позволяет автоматизировать процессы, решать сложные задачи и создавать программы, которые обеспечивают эффективную обработку данных для достижения конкретных целей.

Тема 3.1. Алгоритмы и величины

Цель: Научиться создавать блок схемы алгоритма используя геометрические фигуры.

Вычисления и алгоритмы: Алгоритмы часто используются для выполнения различных вычислительных задач, которые включают в себя работу с числами и другими

величинами. Примерами могут быть математические операции, вычисление статистических показателей, решение уравнений и другие.

Сортировка и поиск: Алгоритмы сортировки и поиска являются важными для работы с коллекциями числовых значений. Они позволяют упорядочивать данные или находить конкретные элементы в них, что может быть полезным для анализа данных.

Алгоритмы и временная сложность: Алгоритмы имеют разную временную сложность, что означает, сколько времени требуется для выполнения алгоритма в зависимости от размера входных данных. Это может быть связано с величинами, такими как количество элементов в массиве или длина строки.

Математические операции: Алгоритмы часто используют математические операции для обработки чисел. Это может включать в себя арифметические операции, операции сравнения, возведение в степень и другие.

Алгоритмы в научных расчетах: Величины могут быть связаны с научными расчетами, например, в физических или инженерных задачах. Алгоритмы могут использоваться для моделирования различных физических процессов и проведения сложных вычислений.

Алгоритмы и обработка изображений/звука: Алгоритмы также могут быть применены для обработки изображений, звука и других мультимедийных данных. Они позволяют преобразовывать и анализировать данные разного типа.

Оптимизация и оптимизационные алгоритмы: Оптимизационные алгоритмы используются для поиска оптимальных решений в задачах, связанных с величинами и ограничениями. Эти алгоритмы могут применяться, например, для оптимизации расписания или распределения ресурсов.

Алгоритмы и статистика: Алгоритмы могут использоваться для анализа статистических данных, вычисления средних, дисперсий, корреляций и других статистических показателей.

Решение задачи на компьютере обычно включает в себя ряд этапов, которые следует последовательно выполнить для достижения требуемого результата.

Понимание задачи: Первый этап - полное понимание поставленной задачи. Это включает в себя анализ постановки задачи, выявление требований и ожидаемых результатов.

Проектирование: На этом этапе вы разрабатываете общий план решения задачи. Определяются алгоритмы, структуры данных и подходы, которые будут использованы.

Разработка алгоритма: На основе проектирования разрабатывается алгоритм, который описывает последовательность шагов для решения задачи. Это может включать в себя вычисления, операции обработки данных и преобразования.

Кодирование: На этом этапе алгоритм переводится в код, который может быть исполнен компьютером. Вы пишете программный код на выбранном языке программирования, используя структуры языка, операции и функции.

Тестирование: После написания кода следует тестирование программы. Вы проверяете, как она работает в разных сценариях и с разными данными. Важно обнаруживать и исправлять ошибки и недоработки.

Оптимизация: Если необходимо, вы можете провести оптимизацию кода и алгоритмов для повышения производительности и эффективности программы.

Документирование: Важно создать документацию к вашей программе, объясняющую ее работу, использование и особенности. Это поможет другим разработчикам понять ваш код.

Развертывание: После успешного тестирования и документирования программы вы можете развернуть ее для использования. Это может включать в себя установку на сервере, распространение среди пользователей и другие этапы.

Обслуживание и поддержка: После развертывания важно следить за работой программы, обнаруживать возможные проблемы и обеспечивать ее поддержку и обновления.

Эти этапы могут незначительно меняться в зависимости от конкретной задачи, но общая структура остается примерно такой же. Планирование и последовательность этапов важны для эффективного и успешного решения задачи на компьютере.

Тема 3.2. Структура алгоритмов

Цель: Понимать структуры алгоритмов для решения задач и программирования действий.

Структура алгоритмов обычно включает в себя несколько основных компонентов, которые определяют логику и последовательность действий для решения конкретной задачи.

1. Ввод данных: Первый этап алгоритма - ввод начальных данных или параметров, с которыми будет работать алгоритм. Входные данные могут быть получены от пользователя, из файла, базы данных или других источников.

2. Обработка данных: Этот этап включает в себя выполнение основных вычислений и операций над входными данными. Здесь определена центральная логика алгоритма, которая приводит к получению желаемого результата.

3. Условия и ветвления: Алгоритмы часто включают в себя проверку условий и ветвления, чтобы выполнение разных частей кода зависело от определенных условий. Для этого используются операторы условных ветвлений, такие как «if», «else», «switch» и другие.

4. Циклы: Циклы позволяют выполнять определенные действия многократно. Они позволяют повторять части алгоритма до тех пор, пока выполняется определенное условие. Примеры циклов - «for», «while», «do-while» и др.

5. Вывод данных: На этом этапе алгоритм производит вывод результатов обработки или вычислений. Результаты могут быть выведены на экран, записаны в файл, переданы другой программе и т.д.

6. Обработка ошибок: Алгоритмы также могут включать обработку ошибок и исключений. Это позволяет предусмотреть ситуации, когда выполнение алгоритма может привести к непредвиденным результатам.

7. Другие элементы: В зависимости от задачи и языка программирования, алгоритмы могут содержать дополнительные элементы, такие как функции и процедуры, обработку строк, массивы и структуры данных, математические операции и многое другое.

Важно разрабатывать структуру алгоритмов таким образом, чтобы они были логичными, понятными и эффективными. Грамотное построение структуры алгоритма обеспечивает читаемость кода, упрощает отладку и поддержку программы.

Базовые структуры

В программировании существует несколько базовых структур, которые служат основой для организации кода и управления потоком выполнения. Эти структуры помогают разработчикам создавать логичные и эффективные алгоритмы. Вот некоторые из базовых структур:

Последовательная структура (Sequential Structure): Последовательная структура представляет собой последовательное выполнение операций или инструкций в порядке их записи. Каждая инструкция выполняется поочередно, без каких-либо переходов. Пример:

```
python
ввод a
ввод b
сумма = a + b
вывод сумма
```

Ветвление (Conditional Structure): Ветвление позволяет программе выбирать между несколькими альтернативами на основе условий. В зависимости от выполнения условия, выполняется определенный блок кода. Пример:

```
python
```

если x > 0 то
  вывести «Положительное число»
иначе
  вывести «Отрицательное или нуль»

Циклы (Loop Structures): Циклы позволяют повторять определенные операции несколько раз. Они выполняют блок кода до тех пор, пока выполняется условие. Существуют разные типы циклов, такие как цикл «for», «while», «do-while». Пример цикла «for»:

python
для i от 1 до 5 с шагом 1 выполнять
  вывести i
конец цикла

Подпрограммы (Subroutines): Подпрограммы (или функции) позволяют разделить код на более мелкие и логические блоки. Они могут быть вызваны из разных мест программы для выполнения определенных задач. Пример:

python
функция вычислить_сумму(a, b)
  вернуть a + b
конец функции

Эти базовые структуры могут комбинироваться и вкладываться друг в друга для создания более сложных алгоритмов. Важно использовать их с умом для создания понятного и эффективного кода.

Тема 3.3. Языки структурного программирования. Элементы языка, типы данных. Операции, функции, выражения

Цель: Познакомиться с элементами структурного программирования, научиться создавать простые приложения.

Теоретический материал

Паскаль - это язык программирования, который разработан с учетом концепций структурного программирования. Он был создан Никлаусом Виртом в начале 1970-х годов и впоследствии стал одним из популярных языков программирования для обучения и разработки.

Одним из ключевых аспектов, который отличает Паскаль как язык структурного программирования, является его поддержка базовых структур, таких как последовательность, ветвление и циклы, которые я рассказал в предыдущем ответе. Это позволяет программистам организовывать код логичным образом, упрощая его чтение, понимание и отладку.

Паскаль также поддерживает использование подпрограмм (или процедур) для разделения кода на более мелкие блоки, а также поддерживает типизированные данные и структуры данных.

Изначально Паскаль создавался как язык для обучения программированию, но в последующие годы он также использовался для разработки реальных программ и систем. Хотя его популярность уменьшилась с появлением более современных языков программирования, Паскаль все еще остается важным языком в истории компьютерных наук и обучения программированию.

Практический материал

Элементы языка Паскаль

Язык программирования Паскаль имеет свой набор элементов, которые разработчики используют для создания программ.

1. Переменные и типы данных: Паскаль поддерживает различные типы данных, такие как целые числа (integer), вещественные числа (real), логические значения (boolean), символы

(char) и строки (string). Вы можете объявлять переменные с использованием ключевого слова var и указывать типы данных.

```pascal
var
  age: integer;
  price: real;
  isAvailable: boolean;
```

2. Операторы: Паскаль поддерживает различные операторы для выполнения арифметических, логических и других операций. Например: +, -, *, / для арифметики, and, or, not для логических операций.

```pascal
sum := a + b;
isTrue := (x > 0) and (y < 10);
```

3. Условные операторы: Для ветвления и выполнения кода в зависимости от условий, Паскаль использует операторы if, else и case.

```pascal
if age >= 18 then
  writeln('Вы совершеннолетний.')
else
  writeln('Вы несовершеннолетний.');
```

4. Циклы: Паскаль поддерживает циклы for, while и repeat-until для повторения блоков кода.

```pascal
for i := 1 to 10 do
  writeln(i);

while x > 0 do
begin
  writeln(x);
  x := x - 1;
end;
```

5. Процедуры и функции: Вы можете создавать собственные процедуры и функции для организации кода и повторного использования.

```pascal
procedure PrintMessage;
begin
  writeln('Hello, World!');
end;

function CalculateSum(a, b: integer): integer;
begin
  CalculateSum := a + b;
end;
```

6. Массивы и записи: Паскаль поддерживает массивы и записи для организации и хранения данных.

```pascal
type
  Student = record
    name: string;
    age: integer;
  end;
```

```
var
  numbers: array[1..5] of integer;
  students: array[1..3] of Student;
```

Типы данных

В языке программирования Паскаль существует несколько базовых типов данных, которые позволяют разработчикам определять, какие виды значений могут храниться в переменных или структурах данных.

1. Целочисленные типы данных:

- integer: Целые числа, как положительные, так и отрицательные.
- shortint: Короткие целые числа, ограниченные диапазоном от -128 до 127.
- byte: Байт, ограниченный диапазоном от 0 до 255.
- longint: Длинные целые числа, с большим диапазоном значений.

2. Вещественные типы данных:

- real: Вещественные числа (числа с плавающей запятой), обычно использующие 4 байта.
- double: Вещественные числа с увеличенной точностью (8 байт).

3. Символьный тип данных:

- char: Один символ.
- string: Строка символов.

4. Логический тип данных:

- boolean: Логические значения true или false.

5. Указатели:

- ^: Указатель на другой тип данных. Позволяет работать с динамической памятью и структурами данных.

6. Перечислимые типы данных:

- enum: Список именованных значений.

7. Массивы:

- Массивы позволяют объединить несколько элементов одного типа в одной переменной.

8. Записи (Records):

- Записи позволяют создавать пользовательские типы данных, содержащие несколько полей разных типов.

9. Множества:

- set: Множества позволяют хранить упорядоченный набор значений одного типа.

10. Файлы:

- file: Тип данных, используемый для работы с файлами.

Кроме этих базовых типов данных, в Паскале также можно определять пользовательские типы данных с помощью ключевых слов type, record, array и других. Это позволяет разработчикам создавать более сложные и абстрактные структуры данных, соответствующие конкретным потребностям.

Операции

В языке программирования Паскаль существует ряд операций, которые позволяют выполнять различные действия с данными, переменными и значениями.

1. Арифметические операции:

- +: Сложение.
- -: Вычитание.
- *: Умножение.
- /: Деление (вещественное деление).
- div: Целочисленное деление.
- mod: Остаток от деления.

Примеры:

```pascal
a := 10 + 5;
b := 20 - 8;
c := 3 * 4;
d := 15 / 2;
e := 20 div 3;
f := 20 mod 7;
```

2. Операции сравнения:

- =: Равно.
- <>: Не равно.
- <: Меньше.
- >: Больше.
- <=: Меньше или равно.
- >=: Больше или равно.

Примеры:

```pascal
x := 5;
y := 10;
isEqual := x = y;
isNotEqual := x <> y;
isLessThan := x < y;
isGreaterThan := x > y;
isLessThanOrEqual := x <= y;
isGreaterThanOrEqual := x >= y;
```

3. Логические операции:

- and: Логическое «и».
- or: Логическое «или».
- not: Логическое «не».

Примеры:

```pascal
a := true;
b := false;
result1 := a and b; // false
result2 := a or b; // true
result3 := not a; // false
```

4. Операции присваивания:

- :=: Присваивание значения переменной.

Пример:

```pascal
x := 10;
y := x + 5;
```

5. Операции инкремента и декремента:

- inc: Увеличение значения переменной на 1.
- dec: Уменьшение значения переменной на 1.

Пример:

```pascal
count := 0;
inc(count); // count становится равным 1
dec(count); // count становится равным 0
```

Это лишь некоторые из основных операций в языке Паскаль. В зависимости от конкретных задач и типов данных, вы можете использовать другие операции для выполнения нужных действий.

Функции

В языке программирования Паскаль функции позволяют создавать блоки кода, которые могут выполнять определенные вычисления и возвращать результат. Функции полезны для структурирования кода, изолирования логики и обеспечения повторного использования определенных операций.

```
pascal
function ИмяФункции(Параметры: ТипыПараметров): ВозвращаемыйТип;
begin
  // Тело функции
  // Выполнение вычислений
  // Возвращение результата
end;
```

- ИмяФункции: Имя функции, которое вы выбираете.
- Параметры: Параметры функции, которые принимаются как аргументы. Параметры могут быть опущены, если функция не принимает входных данных.
- ТипыПараметров: Типы данных параметров.
- ВозвращаемыйТип: Тип данных, который функция будет возвращать.

Примеры:

1. Функция без параметров:

```
pascal
function GetPi: real;
begin
  GetPi := 3.14159;
end;

var
  piValue: real;
begin
  piValue := GetPi;
  writeln('Значение Pi: ', piValue);
end.
```

2. Функция с параметрами:

```
pascal
function CalculateSum(a, b: integer): integer;
begin
  CalculateSum := a + b;
end;

var
  x, y, sum: integer;
begin
  x := 5;
  y := 10;
  sum := CalculateSum(x, y);
  writeln('Сумма: ', sum);
end.
```

Важно, чтобы тип данных, указанный в возвращаемом типе, соответствовал реальному типу значения, возвращаемого функцией. Также учтите, что область видимости переменных, объявленных внутри функции, ограничивается её телом, и они недоступны за пределами функции.

Использование функций позволяет разбить программу на более мелкие блоки, что способствует повышению читаемости кода и упрощению его сопровождения.

Выражения

В языке программирования Паскаль выражения представляют собой комбинации операторов, переменных, констант и функций, которые выполняются для вычисления значения. Выражения могут быть использованы в различных контекстах, таких как присваивание значений переменным, передача аргументов в функции, условные проверки и многое другое.

Арифметические выражения:

```
pascal
x := a + b * c;
total := price * (1 + taxRate);
```

Логические выражения:

```
pascal
isTrue := (x > 0) and (y < 10);
isEven := num mod 2 = 0;
```

Выражения сравнения:

```
pascal
isEqual := a = b;
isGreater := x > y;
isNotEqual := x <> y;
```

Вызовы функций в выражениях:

```
pascal
area := CalculateCircleArea(radius);
result := Max(a, b);
```

Использование выражений в условных операторах:

```
pascal
if age >= 18 then
  writeln('Вы совершеннолетний.');
```

Использование выражений в циклах:

```
pascal
for i := 1 to 10 do
  writeln(i);
```

Присваивание значения выражения переменной:

```
pascal
total := price * quantity;
```

Выражения помогают программистам манипулировать данными, выполнять вычисления и принимать решения внутри программы. Они составляют основу для работы с данными и управления потоком выполнения в языке Паскаль.

Оператор присваивания, ввод и вывод данных

Оператор присваивания, ввод и вывод данных - это важные части программирования, позволяющие взаимодействовать с данными в программе. Давайте рассмотрим каждый из них в контексте языка программирования Паскаль.

1. Оператор присваивания: Оператор присваивания := используется для присвоения значения выражения переменной. Синтаксис следующий:

```
pascal
переменная := выражение;
```

Пример:

```
pascal
x := 10;
y := x + 5;
```

2. Ввод данных: Для ввода данных с клавиатуры используется функция readln. Синтаксис следующий:

```
pascal
```

readln(переменная1, переменная2, ...);

Пример:

```
pascal
var
  name: string;
  age: integer;
begin
  writeln('Введите ваше имя:');
  readln(name);
  writeln('Введите ваш возраст:');
  readln(age);
  writeln('Привет, ', name, '! Вам ', age, ' лет.');
end.
```

3. Вывод данных: Для вывода данных на экран используется функция writeln. Синтаксис следующий:

```
pascal
writeln(выражение1, выражение2, ...);
```

Пример:

```
pascal
var
  x, y: integer;
begin
  x := 5;
  y := 10;
  writeln('Значение x:', x);
  writeln('Значение y:', y);
end.
```

Обратите внимание, что операторы readln и writeln работают с одной строкой ввода или вывода данных. Однако вы можете комбинировать их для работы с разными переменными и выражениями.

Взаимодействие с данными с помощью операторов присваивания, ввода и вывода является неотъемлемой частью разработки программ, так как оно позволяет программам обмениваться информацией с пользователем и обрабатывать данные для получения желаемых результатов.

Логические величины, операции, выражения

Логические величины, операции и выражения в программировании используются для работы с логическими значениями (истина или ложь) и для выполнения логических операций. В языке программирования Паскаль существует поддержка логических данных и операций. Вот основные аспекты:

Логические величины: В языке Паскаль логические величины представлены типом данных boolean. Этот тип может принимать два значения: true (истина) или false (ложь). Логические величины часто используются для создания условий и выполнения ветвления в коде.

```
pascal
var
  isTrue: boolean;
  hasPermission: boolean;
begin
  isTrue := true;
  hasPermission := false;
end.
```

Логические операции: Паскаль поддерживает стандартные логические операции: and, or, not. Они используются для комбинирования и обработки логических значений.

```
pascal
var
  a, b: boolean;
  result1, result2: boolean;
begin
  a := true;
  b := false;

  result1 := a and b;  // false
  result2 := a or b;   // true
end.
```

Логические выражения: Логические выражения представляют собой комбинации логических операций и логических значений. Они используются в условных операторах (if, while, for) и других местах, где нужно проверять истинность или ложность выражения.

```
pascal
var
  age: integer;
  isAdult: boolean;
begin
  writeln('Введите ваш возраст:');
  readln(age);

  isAdult := age >= 18;

  if isAdult then
    writeln('Вы совершеннолетний.')
  else
    writeln('Вы несовершеннолетний.');
end.
```

Сравнение логических выражений: Вы можете сравнивать логические выражения с помощью операций сравнения (=, <>, <, >, <=, >=) и логических операторов (and, or, not). Это позволяет вам создавать более сложные условия.

```
pascal
var
  x, y: integer;
  result: boolean;
begin
  x := 5;
  y := 10;

  result := (x > 0) and (y < 20); // true
end.
```

Логические величины, операции и выражения позволяют программам принимать решения на основе условий и работать с булевыми значениями, что является ключевой частью логики программирования.

Раздел 4. Информационные системы и базы данных

Цель: Научиться создавать реляционные базы данных, создавать и настраивать таблицы, запросы, связи и формы.

Теоретический материал

Информационные системы (ИС) и базы данных (БД) играют важную роль в современном мире, обеспечивая эффективное хранение, управление и обработку больших объемов информации.

Информационные системы (ИС): Информационная система - это комплекс взаимосвязанных компонентов, включающих в себя людей, процессы, технологии и данные, которые совместно выполняют задачи по сбору, хранению, обработке и передаче информации с целью поддержки принятия решений и достижения целей организации или бизнеса. Информационные системы могут быть разнообразными, включая системы управления предприятием (ERP), системы управления отношениями с клиентами (CRM), системы управления проектами и другие.

Базы данных (БД): База данных - это организованная коллекция данных, которая позволяет эффективно хранить, управлять и извлекать информацию. Базы данных используются для хранения структурированных данных различных типов, таких как текст, числа, изображения и другие, с целью обеспечения доступа к этим данным и выполнения запросов. Базы данных позволяют улучшить организацию и анализ данных, обеспечивая централизованное хранение и управление.

Связь между ИС и БД: Информационные системы часто используют базы данных для хранения и управления данными, которые они обрабатывают. Например, в банковских информационных системах хранятся данные о клиентах и их транзакциях в базе данных. Это обеспечивает возможность быстрого доступа к данным и выполнения операций, таких как поиск, сортировка и анализ.

Преимущества использования ИС и БД:

- Улучшение организации данных: ИС и БД позволяют эффективно организовать и структурировать большие объемы информации.
- Улучшение принятия решений: ИС предоставляют аналитические инструменты для анализа данных, что помогает в принятии обоснованных решений.
- Автоматизация бизнес-процессов: ИС автоматизируют выполнение задач, что повышает эффективность работы организации.
- Обеспечение целостности и безопасности: БД обеспечивают контроль целостности данных и меры безопасности, чтобы защитить информацию от несанкционированного доступа.

Примеры использования:

- Онлайн-магазин использует ИС для управления заказами и клиентскими данными, храня информацию в базе данных.
- Банковская система использует БД для хранения информации о клиентах, истории транзакций и балансах.
- Медицинская клиника может использовать ИС для учета пациентов и медицинских записей, храня данные в специализированных базах данных.

Информационные системы и базы данных играют решающую роль в управлении информацией, обеспечивая эффективность, надежность и безопасность обработки данных.

Практический материал

Проектирование базы данных:

Задание: Спроектируйте базу данных для онлайн-магазина, включая таблицы для товаров, заказов, клиентов и т.д. Определите связи между таблицами и атрибуты каждой таблицы.

Создание таблиц и заполнение данными:

Задание: Создайте таблицу для хранения информации о студентах, включая поля для имени, фамилии, возраста и номера студенческого билета. Заполните таблицу несколькими записями.

Написание запросов:

Задание: Напишите SQL-запрос, который выберет всех студентов старше 20 лет из таблицы студентов.

Нормализация данных:

Задание: Проведите нормализацию данных для базы данных о библиотеке, включая таблицы для книг, авторов и жанров. Обоснуйте свои решения.

Обновление данных:

Задание: Напишите SQL-запрос, который обновит цены на все товары в таблице товаров, увеличив их на 10%.

Создание отчетов:

Задание: Напишите SQL-запрос, который выведет список заказов с указанием имени клиента и названия заказанных товаров.

Использование внешних ключей:

Задание: Создайте таблицу для хранения информации о заказах и свяжите ее с таблицами клиентов и товаров с использованием внешних ключей.

Оптимизация запросов:

Задание: Оптимизируйте SQL-запрос для выбора всех студентов, у которых фамилия начинается на «А».

Работа с индексами:

Задание: Создайте индекс для поля с номерами студенческих билетов в таблице студентов и сравните скорость выполнения запросов до и после создания индекса.

Интеграция с программой:

Задание: Напишите программу, которая будет взаимодействовать с базой данных, позволяя добавлять, изменять и удалять данные, а также выполнять различные запросы.

Создание простой базы данных для учета контактов:

Запустите Microsoft Access: Откройте Microsoft Access на вашем компьютере.

Выберите «Пустая база данных»: При запуске Access выберите «Пустая база данных» и укажите имя и расположение для новой базы данных.

Определите структуру таблицы:

Создайте таблицу, кликнув на «Таблица». Введите поля (например, «Имя», «Фамилия», «Эл. почта» и «Телефон») и их типы данных (текст, число и т.д.).

Нажмите «Далее» и дайте таблице имя.

Заполните таблицу данными:

После создания таблицы, заполните ее данными. Добавьте несколько записей с контактами.

Сохраните базу данных:

Сохраните базу данных, выбрав опцию «Сохранить как» и указав имя и место сохранения.

Создание базы данных для учета задач:

Запустите Microsoft Access: Откройте Microsoft Access на вашем компьютере.

Выберите «Пустая база данных»: При запуске Access выберите «Пустая база данных» и укажите имя и расположение для новой базы данных.

Определите структуру таблицы:

Создайте таблицу для учета задач. Введите поля (например, «Название задачи», «Описание», «Дата начала», «Дата завершения» и «Статус») и их типы данных (текст, дата/время и т.д.).

Нажмите «Далее» и дайте таблице имя.

Заполните таблицу данными:

После создания таблицы, заполните ее данными. Добавьте несколько задач с разными характеристиками.

Сохраните базу данных:

Сохраните базу данных, выбрав опцию «Сохранить как» и указав имя и место сохранения.

Создание базы данных для библиотеки:

Запустите Microsoft Access: Откройте Microsoft Access на вашем компьютере.

Выберите «Пустая база данных»: При запуске Access выберите «Пустая база данных» и укажите имя и расположение для новой базы данных.

Определите структуру таблиц:

Создайте таблицу для книг. Введите поля (например, «Название книги», «Автор», «Год издания», «Жанр» и «Статус») и их типы данных (текст, число и т.д.).

Создайте таблицу для читателей. Введите поля (например, «Имя», «Фамилия», «Дата рождения», «Адрес» и «Телефон») и их типы данных.

Нажмите «Далее» и дайте таблицам имена.

Создайте связь между таблицами:

Определите связь между таблицами книг и читателей. Например, создайте поле «ID читателя» в таблице книг, которое будет ссылаться на поле «ID» в таблице читателей.

Заполните таблицы данными:

После создания таблиц, заполните их данными. Добавьте несколько книг и читателей.

Создайте запрос для анализа данных:

Напишите SQL-запрос, который выведет список книг, доступных для чтения (со статусом «доступна») и их авторов.

Создайте форму для ввода данных:

Создайте форму для добавления новых книг и читателей, чтобы упростить процесс ввода данных.

Сохраните базу данных:

Сохраните базу данных, выбрав опцию «Сохранить как» и указав имя и место сохранения.

Создание базы данных для учета сотрудников компании:

Запустите Microsoft Access: Откройте Microsoft Access на вашем компьютере.

Выберите «Пустая база данных»: При запуске Access выберите «Пустая база данных» и укажите имя и расположение для новой базы данных.

Определите структуру таблиц:

Создайте таблицу для сотрудников. Введите поля (например, «Фамилия», «Имя», «Должность», «Отдел», «Дата приема на работу» и «Зарплата») и их типы данных (текст, дата/время, число и т.д.).

Создайте таблицу для отделов. Введите поля (например, «Название отдела» и «Местоположение») и их типы данных.

Нажмите «Далее» и дайте таблицам имена.

Создайте связь между таблицами:

Определите связь между таблицами сотрудников и отделов. Например, создайте поле «ID отдела» в таблице сотрудников, которое будет ссылаться на поле «ID» в таблице отделов.

Заполните таблицы данными:

После создания таблиц, заполните их данными. Добавьте несколько сотрудников разных отделов.

Создайте запрос для анализа данных:

Напишите запрос, который выведет список сотрудников определенного отдела и их должностей.

Создайте отчет о зарплатах:

Создайте отчет, который покажет среднюю зарплату по отделам.

Сохраните базу данных:

Сохраните базу данных, выбрав опцию «Сохранить как» и указав имя и место сохранения.

Создание базы данных для учета клиентов:

Запустите Microsoft Excel: Откройте Microsoft Excel на вашем компьютере.

Создайте новую электронную таблицу:

Создайте новый документ Excel.

Введите заголовки для колонок, такие как «Имя», «Фамилия», «Эл. почта», «Телефон» и т.д.

Вводите данные:

Заполните строки таблицы данными о клиентах.

Фильтрация и сортировка данных:

Используйте функции фильтрации и сортировки для удобного доступа к данным.

Создание базы данных для учета продуктов:

Запустите Microsoft Excel: Откройте Microsoft Excel на вашем компьютере.

Создайте новую электронную таблицу:

Создайте новый документ Excel.

Введите заголовки для колонок, такие как «Наименование продукта», «Цена», «Количество» и т.д.

Вводите данные:

Заполните строки таблицы данными о продуктах.

Использование формул:

Используйте формулы для расчета общей стоимости продуктов (цена * количество) и других вычислений.

Создание диаграмм:

Создайте диаграммы для визуализации данных, например, график цен на продукты.

Создание базы данных для учета задач:

Запустите Microsoft Excel: Откройте Microsoft Excel на вашем компьютере.

Создайте новую электронную таблицу:

Создайте новый документ Excel.

Введите заголовки для колонок, такие как «Название задачи», «Описание», «Дата начала», «Дата завершения» и т.д.

Вводите данные:

Заполните строки таблицы данными о задачах.

Установка условного форматирования:

Примените условное форматирование для выделения сроков выполнения задач.

Использование фильтров:

Используйте фильтры для отображения задач по определенным критериям, например, по статусу «выполнена» или «в процессе».

Создание базы данных для учета расходов:

Запустите Microsoft Excel: Откройте Microsoft Excel на вашем компьютере.

Создайте новую электронную таблицу:

Создайте новый документ Excel.

Введите заголовки для колонок, такие как «Дата», «Категория расходов», «Сумма» и т.д.

Вводите данные:

Заполните строки таблицы данными о ваших расходах.

Суммирование данных:

Используйте функции Excel для автоматического суммирования расходов по категориям и датам.

Создание сводных таблиц:

Создайте сводные таблицы для анализа расходов по категориям и периодам.

Создание базы данных для учета инвентаря:

Запустите Microsoft Excel: Откройте Microsoft Excel на вашем компьютере.

Создайте новую электронную таблицу:

Создайте новый документ Excel.

Введите заголовки для колонок, такие как «Наименование», «Количество», «Стоимость» и т.д.

Вводите данные:

Заполните строки таблицы данными об инвентаре.

Использование фильтров и сортировки:

Примените фильтры и сортировку для удобного поиска и организации данных.

Создание графиков:

Создайте графики для визуализации распределения инвентаря по категориям или стоимости.

Создание базы данных для учета клиентских заказов:

Запустите Microsoft Excel: Откройте Microsoft Excel на вашем компьютере.

Создайте новую электронную таблицу:

Создайте новый документ Excel.

Введите заголовки для колонок, такие как «Номер заказа», «Дата заказа», «Клиент», «Сумма» и т.д.

Вводите данные:

Заполните строки таблицы данными о заказах от клиентов.

Создание сводных таблиц:

Создайте сводные таблицы для анализа суммарных заказов по клиентам и периодам.

Использование формул:

Используйте формулы для автоматического расчета суммы заказа и других статистических данных.

Тема 4.1. Что такое система. Модели систем

Система - это совокупность взаимосвязанных и взаимодействующих элементов, объединенных для достижения определенной цели или выполнения определенных функций. В широком смысле системой можно назвать любой организованный комплекс элементов, которые взаимодействуют между собой и с окружающей средой.

Основные характеристики системы:

Элементы: Система состоит из отдельных компонентов, которые называются элементами. Элементы могут быть материальными (например, детали в механической системе) или абстрактными (например, функции в программной системе).

Взаимодействие: Элементы системы взаимодействуют друг с другом. Их взаимодействие может проявляться в передаче информации, энергии, сигналов и других ресурсов.

Организация: Элементы системы организованы определенным образом, чтобы выполнять определенные функции. Организация может включать в себя структуру, иерархию, связи и порядок работы элементов.

Цель: Система имеет определенную цель или задачу, которую она выполняет. Цель системы может быть разнообразной, например, производство товаров, обработка данных или управление проектами.

Граничная область: Система имеет границу, которая разделяет ее от окружающей среды. Граничная область определяет, какие элементы и воздействия входят в систему, а какие остаются за ее пределами.

Вывод: Система производит результаты или выводы в соответствии с ее целью. Эти результаты могут быть видимыми продуктами, информацией или действиями.

Обратная связь: Обратная связь представляет собой механизм передачи информации об исполнении системы. Она позволяет системе корректировать свои действия и адаптироваться к изменениям в окружающей среде.

Примеры систем могут включать в себя механические устройства, программные приложения, экологические системы, бизнес-процессы, социальные структуры и многие другие. Системный подход широко применяется для анализа, проектирования и оптимизации различных областей деятельности, позволяя лучше понимать сложные взаимодействия и зависимости между элементами.

Модели систем

Модели систем представляют собой абстрактные конструкции или представления реальных систем с целью анализа, понимания и прогнозирования их поведения. Модели помогают упростить сложные системы, выделяя основные характеристики и отношения между элементами. В зависимости от целей и природы системы, существует несколько видов моделей:

1. **Физические модели:** Это модели, которые создаются в физическом пространстве и имитируют поведение реальных объектов и процессов. Примеры включают физические макеты, макеты зданий и механизмов.
2. **Графические модели:** Это модели, представленные в виде графиков, схем, диаграмм и других визуальных образов. Они используются для визуализации структуры и взаимодействия элементов системы. Примеры включают блок-схемы, сетевые диаграммы, организационные структуры.
3. **Математические модели:** Это формальные математические выражения, которые описывают отношения и зависимости между элементами системы. Математические модели позволяют проводить количественный анализ и симуляции поведения системы. Примеры включают дифференциальные уравнения, статистические модели, линейные программы.
4. **Концептуальные модели:** Это абстрактные модели, которые описывают концептуальную структуру системы и ее ключевые компоненты. Они обычно используются на начальных этапах проектирования для определения общей концепции системы.
5. **Компьютерные модели:** Это модели, созданные с использованием компьютерных программ и специальных программных средств. Компьютерные модели позволяют проводить численные симуляции и анализ сложных систем.
6. **Символьные модели:** Это модели, представляющие систему в виде символов и символьных выражений. Они широко используются в информационных системах, например, для описания логики и алгоритмов.

Модели систем могут быть статическими или динамическими, а также дискретными или непрерывными в зависимости от того, какие аспекты системы они учитывают. Выбор подходящей модели зависит от целей анализа, сложности системы и доступных ресурсов.

Примеры структурной модели предметной области

Структурные модели предметной области представляют собой абстрактные представления о структуре и организации элементов и отношений в данной области. Вот несколько примеров структурных моделей для различных предметных областей:

1. **Организационная структура предприятия:** В данной модели элементами могут быть отделы, должности, сотрудники, руководители и их взаимосвязи. Отношения могут быть иерархическими, функциональными, информационными и т.д.
2. **Структура базы данных:** Для базы данных структурная модель может включать в себя таблицы, поля, ключи, связи между таблицами и другие компоненты. Эта модель описывает организацию данных и их взаимосвязи.

3. **Функциональная структура программного приложения:** В этой модели элементами могут быть модули, функции, классы и методы программного кода. Отношения определяют, как модули взаимодействуют друг с другом для выполнения задачи.

4. **Структура сети:** Для сетевой инфраструктуры элементами модели могут быть узлы, маршрутизаторы, коммутаторы и соединения между ними. Эта модель описывает, как устройства связаны для обмена данными.

5. **Структура молекулы:** В химии структурная модель может представлять молекулу в виде атомов и связей между ними. Эта модель помогает понять химический состав и структуру вещества.

6. **Иерархия классов в объектно-ориентированном программировании:** Для программирования элементами этой модели будут классы, наследование, интерфейсы и отношения между классами. Она помогает организовать код и определить отношения между объектами.

Эти примеры демонстрируют, как структурные модели могут быть применены в различных областях для анализа, проектирования и понимания структуры и взаимосвязей элементов.

Тема 4.2. Пример структурной модели предметной области. Что такое информационная система

Информационная система (ИС) - это комплексно организованная система, которая включает в себя совокупность взаимосвязанных элементов, включая людей, процессы, технологии и данные, с целью сбора, обработки, хранения, передачи и использования информации для поддержки принятия решений, выполнения задач и достижения целей организации или бизнеса.

Основные характеристики информационных систем:

1. **Элементы:** Элементами информационной системы могут быть люди, аппаратное и программное обеспечение, данные, процессы и методы.

2. **Взаимодействие:** Элементы информационной системы взаимодействуют друг с другом, передавая информацию, ресурсы и управляя данными и процессами.

3. **Цель:** Информационная система имеет определенную цель или набор задач, которые она выполняет. Цели могут варьироваться от автоматизации бизнес-процессов до предоставления аналитической информации.

4. **Данные:** Данные являются ключевым аспектом информационных систем. Они собираются, хранятся, обрабатываются и используются для получения информации.

5. **Процессы:** Процессы описывают последовательность операций и действий, которые выполняются в рамках системы для достижения целей.

6. **Технологии:** Технологии включают в себя аппаратное и программное обеспечение, которые поддерживают функционирование системы.

7. **Обратная связь:** Обратная связь позволяет системе адаптироваться к изменениям в окружающей среде и корректировать свои действия.

Примеры информационных систем могут быть разнообразными, включая системы управления предприятием (ERP), системы управления отношениями с клиентами (CRM), системы управления базами данных (БД), системы онлайн-торговли, банковские информационные системы и многие другие. Они могут быть ориентированы на автоматизацию процессов, обработку данных, аналитику, связь с клиентами и другие функции в соответствии с потребностями организации.

Тема 4.3. База данных – основа информационной системы. Проектирование много табличной БД

Цель: Научиться создавать базы данных, таблицы, связи, запросы, формы.

База данных (БД) играет ключевую роль в информационной системе (ИС) и считается ее основой. База данных представляет собой организованную коллекцию данных, которая хранит информацию, необходимую для функционирования и достижения целей информационной системы.

Вот почему база данных считается основой информационной системы:

1. **Хранение данных:** База данных предоставляет структурированный способ хранения данных. Она позволяет сохранять информацию о клиентах, продуктах, заказах, транзакциях и других аспектах, необходимых для работы ИС.
2. **Обеспечение доступа:** БД обеспечивает доступ к данным. Она позволяет пользователям извлекать, обновлять и анализировать информацию в соответствии с их потребностями.
3. **Централизация:** Использование единой базы данных позволяет централизованно управлять данными. Это уменьшает дублирование данных и обеспечивает единую точку доступа.
4. **Целостность данных:** БД поддерживает целостность данных, что означает, что она предотвращает некорректные или несогласованные данные. Это важно для поддержания точности информации.
5. **Безопасность данных:** Базы данных позволяют устанавливать уровни доступа и права пользователей. Это обеспечивает защиту конфиденциальных данных и контроль над доступом.
6. **Управление данными:** Системы управления базами данных (СУБД) предоставляют инструменты для управления данными, включая добавление, изменение, удаление и поиск информации.
7. **Поддержка аналитики:** Данные в базе данных могут быть анализированы и использованы для принятия решений. Это может помочь выявить тенденции, паттерны и прогнозировать будущее.
8. **Интеграция:** Информационные системы часто взаимодействуют между собой, и база данных может служить мостом между разными системами, обеспечивая обмен данными.

Следовательно, правильное проектирование, организация и управление базой данных имеет огромное значение для эффективной работы информационной системы.

Проектирование многотабличной БД

Проектирование многотабличной базы данных (БД) требует внимательного планирования, чтобы обеспечить эффективное хранение, организацию и доступ к данным. Вот шаги, которые следует учесть при проектировании многотабличной БД:

1. **Анализ требований:** Первым шагом является анализ требований системы. Определите, какие данные должны быть хранены, какие связи между данными существуют и какие операции будут выполняться над данными.
2. **Идентификация сущностей:** Определите сущности (объекты или понятия), которые будут представлены в БД. Например, для интернет-магазина это могут быть клиенты, продукты, заказы и т.д.
3. **Определение отношений:** Определите, какие отношения существуют между сущностями. Эти отношения могут быть однозначными (например, каждый заказ связан с одним клиентом) или многозначными (например, каждый клиент может иметь несколько заказов).
4. **Нормализация:** Примените процесс нормализации, который помогает организовать данные для устранения избыточности и обеспечения эффективности. Обычно, это включает в себя разделение данных на таблицы, чтобы избежать повторения информации.
5. **Определение атрибутов:** Для каждой сущности определите атрибуты (поля), которые будут храниться в таблицах. Например, для сущности «клиент» атрибутами могут быть имя, адрес, номер телефона и т.д.

6. **Определение первичных ключей:** Для каждой таблицы определите первичный ключ, который уникально идентифицирует записи в таблице. Первичные ключи обеспечивают уникальность и быстрый доступ к данным.

7. **Определение внешних ключей:** Если существуют отношения между таблицами, определите внешние ключи. Внешние ключи связывают записи в одной таблице с записями в другой, обеспечивая целостность данных.

8. **Проектирование структуры таблиц:** Создайте таблицы в соответствии с определенными атрибутами, первичными и внешними ключами. Учтите типы данных, индексы и ограничения для каждой таблицы.

9. **Определение запросов и операций:** Рассмотрите, какие запросы и операции будут часто выполняться над данными, и убедитесь, что структура БД поддерживает эти операции.

10. **Оптимизация:** Проведите оптимизацию БД для обеспечения быстрого доступа к данным. Это может включать в себя создание индексов, кластеризацию таблиц и другие меры.

11. **Тестирование:** Перед тем как внедрять систему, протестируйте ее на соответствие требованиям и выполнение операций.

Проектирование многотабличной БД - это сложный процесс, который требует понимания требований, анализа структуры данных и учета эффективности. Корректно спроектированная БД обеспечит надежное и эффективное хранение и обработку данных.

Тема 4.4. Создание базы данных. Запросы как приложение ИС. Логические условия выбора данных

Цель: Изучить рабочее пространство приложения MS Access, научиться формировать структуру таблиц, создавать формы и отчеты, познакомиться с возможностями фильтрации данных.

Теоретический материал:

Создание базы данных (БД) включает в себя ряд этапов, начиная от планирования структуры данных и заканчивая фактическим созданием таблиц и отношений. Вот шаги, которые вы можете выполнить для создания базы данных:

Определение требований: Первым шагом является понимание требований к базе данных. Определите, какие данные будут храниться, какие операции будут выполняться, и какие отношения между данными существуют.

Выбор системы управления базами данных (СУБД): Выберите подходящую СУБД для вашей системы. Например, это может быть MySQL, PostgreSQL, Microsoft SQL Server, Oracle и др. Выбор зависит от ваших потребностей, бюджета и экосистемы.

Проектирование структуры данных: На основе анализа требований определите сущности, атрибуты и отношения между ними. Рассмотрите, какие таблицы будут необходимы и какие поля будут содержаться в каждой таблице.

Создание таблиц: Используйте SQL (Structured Query Language) для создания таблиц в СУБД. Определите их структуру, включая названия полей, типы данных, ограничения, первичные и внешние ключи.

Определение связей: Если данные имеют отношения, определите внешние ключи, чтобы связать записи в разных таблицах. Это обеспечит целостность данных и возможность выполнять запросы с использованием связей.

Создание индексов: Создайте индексы для полей, которые часто используются для поиска или сортировки данных. Индексы ускоряют выполнение запросов.

Заполнение данными: Заполните таблицы данными, используя операторы INSERT. Введите начальные данные, которые будут служить основой для работы с БД.

Тестирование: Протестируйте функциональность и целостность базы данных. Выполните разнообразные запросы и убедитесь, что данные хранятся и извлекаются корректно.

Оптимизация: Оптимизируйте запросы и структуру данных для обеспечения быстрого доступа и эффективной работы с БД.

Безопасность: Установите соответствующие права доступа к данным для пользователей и ролей. Обеспечьте защиту данных от несанкционированного доступа.

Резервное копирование: Разработайте стратегию регулярного резервного копирования данных. Это поможет восстановить данные в случае сбоев или потери информации.

Внедрение: Разверните созданную базу данных на сервере и начните использование. Обучите пользователей работе с ней и убедитесь, что всё функционирует как задумано.

Создание базы данных - это итеративный процесс, который может потребовать корректировок и оптимизаций по мере развития проекта. Хорошо спроектированная и организованная БД обеспечит надежное хранение данных и эффективную работу вашей информационной системы.

Запросы как приложения ИС

Запросы в базе данных играют важную роль в функционировании информационных систем. Они позволяют получать, изменять, фильтровать и агрегировать данные с целью выполнения различных операций и анализа. Запросы можно рассматривать как приложения в информационной системе, так как они выполняют определенные функции над данными.

Извлечение данных: Запросы позволяют извлекать информацию из базы данных в соответствии с заданными критериями. Например, в ИС для учета продаж можно создать запрос для получения списка всех заказов за определенный период времени.

Фильтрация и сортировка данных: Запросы позволяют фильтровать и сортировать данные по определенным условиям. Например, в ИС для управления инвентарем можно создать запрос для поиска всех товаров с нулевым остатком на складе.

Агрегация данных: Запросы позволяют агрегировать данные для получения обобщенных результатов. Например, в ИС для управления проектами можно создать запрос для расчета суммарной стоимости и затрат на каждый проект.

Обновление данных: Запросы позволяют вносить изменения в данные. Например, в ИС для учета сотрудников можно создать запрос для изменения информации о зарплате определенного сотрудника.

Создание отчетов: Запросы могут быть использованы для создания отчетов, которые предоставляют визуализированную информацию. Например, в ИС для учета финансов можно создать запрос для получения ежемесячного отчета о доходах и расходах.

Создание динамических представлений: Запросы позволяют создавать временные представления данных для конкретных целей. Например, в ИС для управления клиентами можно создать запрос для формирования списка клиентов, у которых истек срок действия контракта.

Автоматизация рутинных задач: Запросы могут быть автоматизированы и выполняться по расписанию. Например, в ИС для учета складских запасов можно создать запрос, который будет автоматически обновлять остатки товаров каждую неделю.

Таким образом, запросы в базе данных могут служить важными компонентами информационных систем, выполняя разнообразные функции и обеспечивая доступ к актуальной и необходимой информации.

Логические условия выбора данных

Логические условия выбора данных являются основой для создания запросов в базе данных. Они позволяют фильтровать данные в соответствии с определенными критериями и выбирать только те записи, которые соответствуют заданным условиям. Логические условия состоят из логических операторов и сравнений.

Оператор равенства (=): Выбирает записи, которые имеют значение атрибута точно равное заданному значению. Например: age = 30.

Оператор неравенства (!= или <>): Выбирает записи, которые имеют значение атрибута, отличное от заданного значения. Например: status != 'завершен'.

Оператор больше (>), меньше (<): Выбирает записи, где атрибут больше (или меньше) заданного значения. Например: price > 100.

Оператор больше или равно (>=), меньше или равно (<=): Выбирает записи, где атрибут больше или равен (или меньше или равен) заданному значению. Например: quantity >= 10.

Оператор BETWEEN: Выбирает записи, где атрибут находится в заданном диапазоне значений. Например: year BETWEEN 2000 AND 2020.

Оператор IN: Выбирает записи, где атрибут имеет одно из заданных значений. Например: category IN ('фрукты', 'овощи').

Оператор NOT: Инвертирует логическое условие, выбирая записи, которые не соответствуют заданному условию. Например: NOT status = 'отменен'.

Логический оператор AND: Выбирает записи, которые соответствуют обоим заданным условиям. Например: age >= 18 AND gender = 'женский'.

Логический оператор OR: Выбирает записи, которые соответствуют хотя бы одному из заданных условий. Например: category = 'электроника' OR price > 500.

Логический оператор NOT: Инвертирует логическое условие, выбирая записи, которые не соответствуют заданному условию. Например: NOT age < 18.

Логические условия позволяют точно определить, какие данные будут выбраны при выполнении запроса. Они позволяют создавать более гибкие и управляемые запросы для работы с базой данных.

Microsoft Access - это функционально полная реляционная СУБД. В ней предусмотрены все необходимые средства для определения и обработки данных, а также для управления ими при работе с большими объемами информации. СУБД Microsoft Access предоставляет возможность контролировать задание структуры и описание своих данных, работу с ними и организацию коллективного пользования этой информацией. Она также существенно увеличивает возможности и облегчает каталогизацию и ведение больших объемов хранящейся в многочисленных таблицах информации. Microsoft Access включает в себя три основных типа функций: определение (задание структуры и описание) данных, обработка данных и управление данными. MS Access — это набор инструментальных средств для создания и эксплуатации информационных систем.

Средствами Access можно проводить следующие операции.

1.Проектирование базовых объектов ИС — двумерных таблиц, с разными типами данных, включая поля объектов OLE. Прежде чем заполнять данными таблицу, надо создать ее макет.

2.Установление связей между таблицами, с поддержкой целостности данных, каскадного обновления полей и каскадного удаления записей.

3.Ввод, хранение, просмотр, сортировка, модификация и выборка данных из таблиц с использованием различных средств контроля информации, индексирования таблиц и аппарата алгебры логики (для фильтрации данных).

4.Создание, модификация и использование производных объектов ИС (форм, запросов и отчетов).

Объектом обработки MS Access является файл базы данных, имеющий произвольное имя и расширение .MDB. В этот файл входят основные объекты MS Access: таблицы, формы, запросы, отчеты, макросы и модули.

Информация в базе данных Access представляется в виде отдельных таблиц. При этом каждый столбец таблицы соответствует полю данных, а каждая строка - записи данных. Запись данных состоит из нескольких полей. При этом действует следующее правило: запись данных представляет собой группу взаимосвязанных полей, рассматриваемых как единое целое.

Каждая запись данных в таблице содержит одинаковое число полей. Каждое поле содержит один и тот же тип информации. Тип данных для конкретного поля данных выбирается в зависимости от того, какая информация будет располагаться в этом поле. Размер поля данных определяется в зависимости от выбранного для него типа.

Кроме таблиц СУБД Access работает со следующими объектами: формами; запросами; отчетами; макросами; модулями.

Форма помогает вводить, просматривать и модифицировать информацию в таблице или запросе. Запросы и отчеты выполняют самостоятельные функции: выбирают, группируют, представляют, печатают информацию.

Каждый объект MS Access имеет имя. Длина имени любого объекта MS Access (таблицы, формы и т. д.) — не более 64 произвольных символов (за исключением точки и некоторых служебных знаков). В имя могут входить пробелы и русские буквы. С каждым объектом работают в отдельном окне, причем предусмотрено два режима работы:

1)режим конструктора, — когда создается или изменяется макет, структура объекта (например, структура таблицы);

2)оперативный режим, — когда в окне задачи ИС просматривается, изменяется, выбирается информация.

Кроме того, в файл базы данных входит еще один документ, имеющий собственное окно: схема данных. В этом окне создаются, просматриваются, изменяются и разрываются связи между таблицами. Эти связи помогают контролировать данные, создавать запросы и отчеты.

Запрос — это производная таблица, в которую входят данные из других таблиц и над ними проводятся различные операции. В частности, в запросе могут появиться вычисляемые поля, т. е. поля, значения которых являются функциями значений других полей (возможно, из разных таблиц). Кроме того, запросы позволяют проводить групповые операции, т.е. операции над группой записей, объединенных каким-то общим признаком (например, можно просуммировать количество для записей с одним и тем же кодом). Наконец, запросы позволяют составлять выборки из таблиц по какому-то условию. В таких случаях применяется алгебра логики.

Отчет — это фактически тот же запрос, но оформленный так, чтобы его можно было напечатать на бумаге и представить начальству (с красивыми заголовками, промежуточными итогами и т. п.).

Создание любых объектов возможно двумя способами: с помощью мастера и в режиме конструктора.

Мастера сами выполняют нужные действия. При вызове Мастера открывается соответствующая последовательность диалоговых окон. Получив ответы на заданные вопросы, Мастер выполняет все необходимые действия. Предположим, вы хотите сформировать отчет. Мастер отчетов спросит вас, как должен выглядеть отчет и какие данные он должен содержать. После получения ответов отчет будет полностью готов. С помощью других Мастеров можно формировать таблицы баз данных, запросы и формы. Работа с Мастерами проста, надежна и эффективна, и позволяет использовать все богатство возможностей Microsoft Access.

Конструктор (Builder) - это инструмент Access, который облегчает выполнение конкретного задания. Крайне полезным является Конструктор выражений - он позволяет быстро сформировать сложное выражение. С помощью Конструктора запросов легко формируются запросы, которые используются для получения выборок данных для формы или отчета. Помимо перечисленных, в Access имеются и другие конструкторы. Это - макро-конструктор, с помощью которого формируются различные макросы, а также конструкторы меню, полей, цветов, кодов и другие.

Практический материал:

Задание 1.

Создайте новую базу данных СЕССИЯ.

Технология работы

1. Откройте программу Microsoft Access 2010: Пуск – Программы – Microsoft Access - Microsoft Access 2010

2. В меню Файл выберите команду Создать – Новая база данных (рисунок 1).

3. Через строку Имя файла войдите в Свою папку (если своя папка не создана создайте ее), в строке Имя файла назовите будущую базу данных СЕССИЯ, нажмите кнопку Создать.

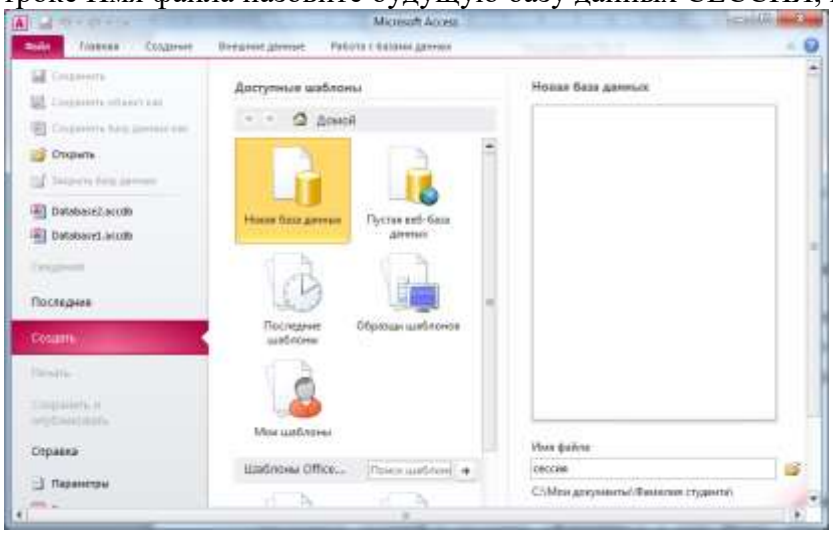


Рисунок 1. Диалоговое окно создания новой базы данных

После выпоненных дейсвий откроется окно базы данных с активным объектом Таблица 1 (рисунок 2).

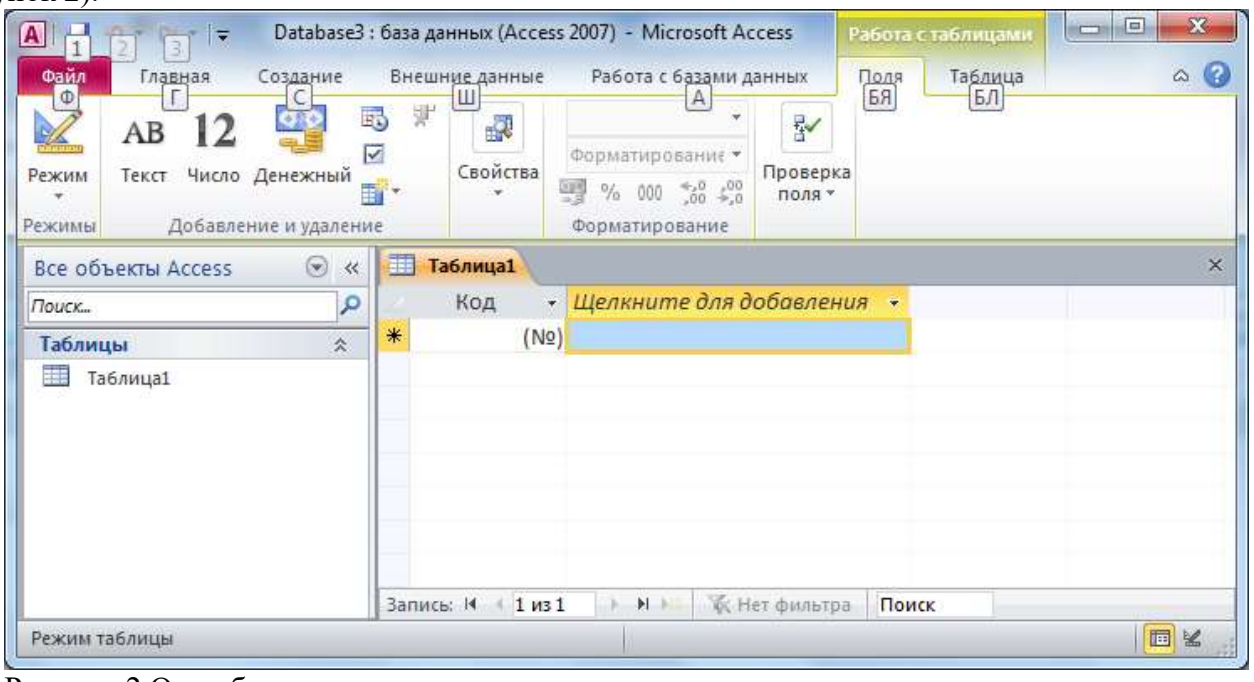


Рисунок 2 Окно базы данных

Необходимые для работы объекты можно выбрать на Панели объектов в левой части экрана (рисунок 3).

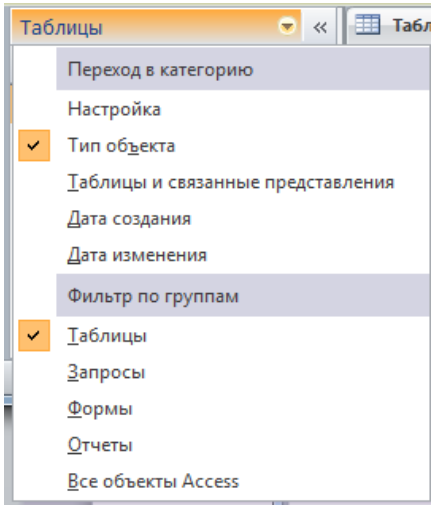


Рисунок 3. Панель объектов

Задание 2.

Сформируйте структуру таблицы СТУДЕНТ для хранения в ней справочных сведений о студентах, обучающихся в вузе. Имена, типы и размеры полей таблицы приведены на таблице 1.

Таблица 1. Структура таблицы СТУДЕНТ

Поле	Тип поля	Размер поля
Номер	Текстовое	5
Фамилия	Текстовое	15
Имя	Текстовое	10
Отчество	Текстовое	15
Пол	Текстовое	1
Дата рождения	Дата/время	Краткий формат даты
Группа	Текстовое	3

Технология работы

1. В левой части вкладки Файл выберите режим Конструктор.

2. В открывшемся диалоговом окне Сохранение введите имя таблицы СТУДЕНТ. Нажмите ОК. В результате проделанных операций открывается окно таблицы в режиме конструктора, в котором следует определить поля таблицы.

3. Определите поля таблицы в соответствии с таблицей 1 и рисунок 4.

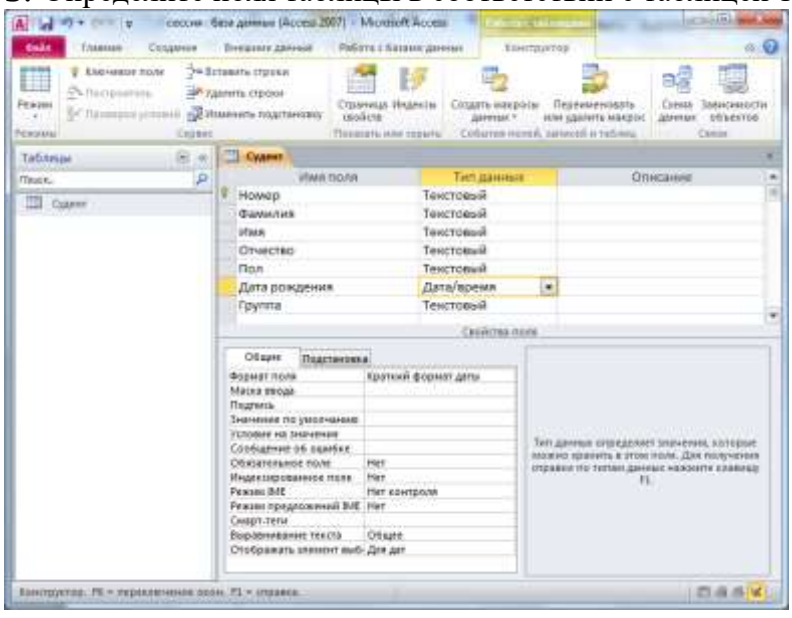


Рисунок 4. Таблица Студент в режиме конструктора

4. Для определения первого поля выполните следующие действия:

● введите в ячейку столбца Поле имя первого поля Номер, обратите внимание, что первое поле таблицы по умолчанию определено как ключевое;

● в ячейке столбца Тип данных оставьте выводящееся по умолчанию значение Текстовый;

● переключитесь на панель Свойства поля (нижняя часть таблицы), откорректируйте Размер поля (введите 5).

Примечание. Заполнение ячеек столбца Описание является необязательным и включает сведения о содержащихся в поле данных.

5. Для определения всех остальных полей таблицы базы данных СЕССИЯ в соответствии с рисунком 4 выполните действия, аналогичные указанным в п.4. Если значение типа Текстовый не подходит, то нажмите кнопку раскрытия списка и выберите нужный тип данных.

6. Сохраните таблицу, нажав на кнопку Сохранить:

Задание 3.

Введите данные о студентах в таблицу СТУДЕНТ.

Технология работы

1. Откройте таблицу СТУДЕНТ в режиме таблицы. Для этого в левой части вкладки Файл выберите Режим таблицы.

2. Введите данные об учебной группе в таблицу СТУДЕНТ,

3. Если в таблице определено ключевое поле, то это поле должно быть обязательно заполнено, причем уникальными записями. Заполните поле Номер номерами зачетных книжек, как на рисунке 5.1

Номер
16493
16793
16693
16893
16593
17193
16993
17093
*

Рисунок 5.1. Поле Номер

4. Заполните все остальные поля, как это показано на рисуноке 5.2 Закройте таблицу СТУДЕНТ. На запрос о сохранении данных ответьте ДА.

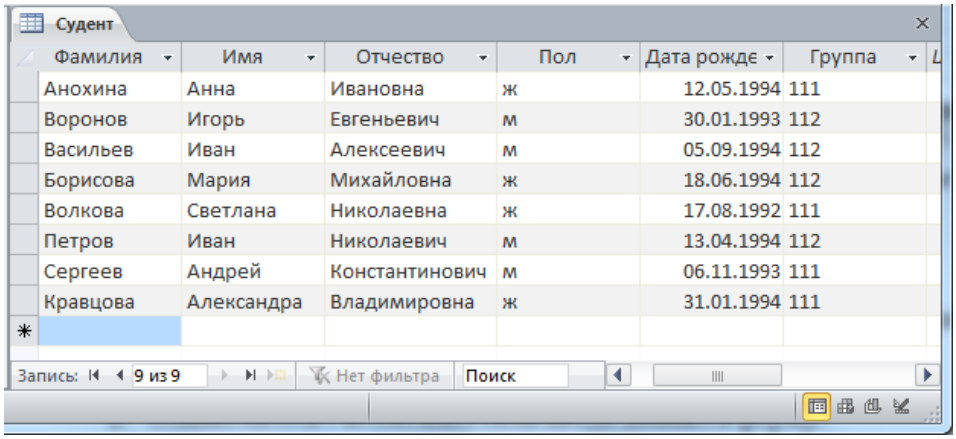


Рисунок 5.2. Заполненная данными таблица СТУДЕНТ в Режиме таблицы

Задание 4.

1. Создайте однотабличную пользовательскую форму для ввода и редактирования данных таблицы СТУДЕНТ при помощи Мастера форм.

2. Ознакомьтесь с возможностями ввода данных в форму.

Технология работы

1. На панели объектов выберите тип объекта Формы;

2. На панели Создание нажмите на кнопку Мастер Форм;

3. В первом диалоговом окне Мастера, убедитесь, что в качестве источника данных выбрана таблица СТУДЕНТ.

4. Из списка Доступные поля поместите все поля в список Выбранные поля, для этого щелкните левой кнопкой мыши по стрелке.

5. Нажмите кнопку Далее.

Справка: если вы не собираетесь использовать в форме какое-либо поле, можно при помощи кнопки убрать его из списка выбранных полей.

6. Выберите внешний вид формы: В один столбец. Далее.

7. Задайте имя формы: СТУДЕНТ - Готово.

На экране появится окно с выводом данных из таблицы в виде формы.

8. Познакомьтесь с возможностями перемещения по записям формы.

9. Добавьте в таблицу запись в режиме формы. Содержание добавляемой записи может быть произвольным

10. Сохраните созданную форму и закройте ее.

Задание 5.

С помощью мастера создайте детальный отчет для вывода данных таблицы СТУДЕНТ.

Технология работы

1. На панели Создание выберите Мастер отчетов;

2. В первом диалоговом окне Мастера, убедитесь, что в качестве источника данных выбрана таблица СТУДЕНТ.

3. Выберите необходимые для отчета поля. В списке Доступные поля щелкните поле Фамилия а затем по кнопке для перемещения поля в список полей, выбранных для создания отчета;

4. Аналогичным образом выберите для включения в отчет поля Имя, Отчество, Группа. По окончании этой операции щелкните на кнопке Далее;

5. Добавьте уровень группировки по полю Группа, для чего установите курсор на это поле и нажмите стрелку «>«, Далее;

6. Задайте порядок сортировки по полям Фамилия и Имя, Далее;

7. Выберите вид макета Ступенчатый и книжную ориентацию, Далее;

8. Задайте имя отчета СТУДЕНТ - Готово.

Задание 6.

1. Для данных, содержащихся в таблице СТУДЕНТ, в режиме формы осуществите поиск одной из записей.

2. В режиме таблицы отсортировать записи по возрастанию значений одного из полей.

3. Отфильтровать данные в соответствии с критерием отбора.

Технология работы

1. Откройте таблицу СТУДЕНТ в режиме формы. Для этого: в окне базы данных СЕССИЯ выберите объект Формы и дважды щелкните по форме СТУДЕНТ.

2. Найдите запись таблицы с информацией о студентке с фамилией Борисунокова. С этой целью выполните следующую группу действий:

- находясь в форме СТУДЕНТ, щелкните в строке поля Фамилия; Затем выполните команду Найти на вкладке Главная;
- задайте образец для поиска слово Борисунокова,
- щелкните на кнопке Найти далее. В форму выведется найденная запись.

3. Закройте окно формы.

4. Откройте таблицу СТУДЕНТ в табличном режиме.

5. Отсортируйте записи таблицы в соответствии с алфавитным порядком фамилий студентов, что потребует от вас следующих действий:

- щелкните на столбце Фамилия;
- щелкните по кнопке пиктографического меню По возрастанию или выберите пункт Сортировка от А до Я (рисунок 6). Записи таблицы будут выведены на экран в соответствии с алфавитным порядком фамилий.

Рисунок 6. Сортировка данных таблицы

6. Используйте фильтрацию, выведите на экран только записи, относящиеся к студентам, родившимся раньше 1994 года, женского пола. Для этого выполните следующий порядок действий:

- в окне с таблицей СТУДЕНТ на вкладке Главная выберите Дополнительно – Расширенный фильтр;
- в окне фильтра в строке Поле выберите поле с именем Пол, введите условие отбора - ж, Дата рождения – условие отбора <01.01.1994;
- щелкните на кнопке Применить фильтр.
- На экран выведутся только записи, соответствующие введенному критерию отбора.
- Удалите фильтр. Для этого щелкните по кнопке Удалить фильтр.

Задание 7.

Сформируйте запрос-выборку, позволяющий получить из таблицы СТУДЕНТ данные о студентах мужского пола, родившихся после 1993 г.

Технология работы

1. В окне базы данных СЕССИЯ выберите объект Запросы;

2. На вкладке Создание выберите Конструктор запросов;

3. Из диалогового окна Добавление таблицы выберите таблицу СТУДЕНТ и добавьте ее в бланк запроса при помощи кнопки Добавить. Закройте окно Добавление таблицы.

4. В первую ячейку строки Поле перетащить из списка полей таблицы СТУДЕНТ поле Фамилия, во вторую — Имя, в третью — Отчество в четвертую — Дата рождения, в пятую — Пол,

5. Напротив поля Пол в строку Условие отбора поместить выражение м и уберите признак вывода на экран информации из этого поля (Снимите «птичку» в строке Вывод на экран);

6. Напротив поля Дата рождение в строку Условие отбора поместить выражение: >31.12.93 (Рисунок 7)

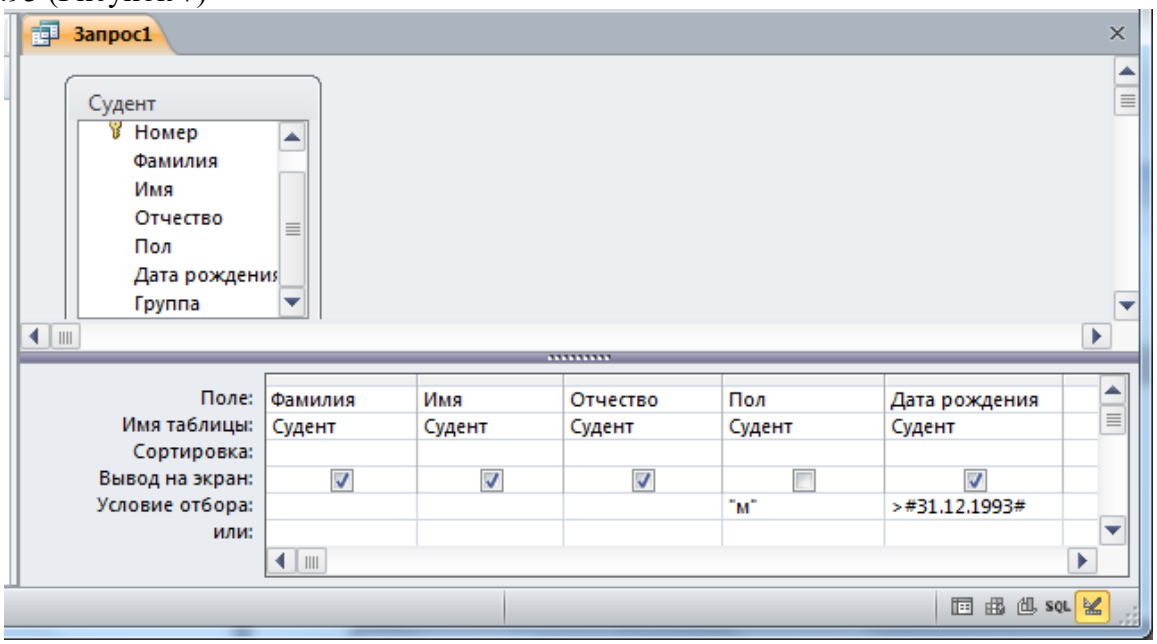


Рисунок 7. Запрос в режиме конструктора

7. Выполните запрос, для чего щелкните на кнопке Выполнить на вкладке Конструктор.

8. Сохранить запрос, для этого выполнить команду Сохранить. В появившемся окне введите имя запроса, например, можно оставить имя Запрос 1, предлагаемое по умолчанию. Закройте запрос.

9. Закройте базу данных. Для этого выполните команду меню Файл - Выход.

Сохраните результаты проделанной работы в своей папке под названием Сессия 1.

Задание для развития и контроля владения компетенциями:

1. Назовите элементы окна приложения Microsoft Access.
2. Опишите технологию создания таблицы в режиме конструктора.
3. Назовите известные вам типы данных для полей таблицы.
4. Опишите технологию создания формы и отчета при помощи мастера.
5. Опишите технологию использования формы при поиске данных и добавлении данных в таблицу.
6. Расскажите о возможностях фильтрации данных и проведите фильтрацию данных в таблице по своим критериям.

Задание 8.

Создайте структуры таблиц СЕССИЯ и СТИПЕНДИЯ в соответствии с таблицами 1 и 2, установите ключевые поля в таблицах.

Таблица 1. Структура таблицы СЕССИЯ

Признак ключа	Имя поля	Тип поля	Формат поля	Размер поля
Ключевое	Номер	Текстовое	-	5
	Оценка 1	Числовое	Фиксированный	Длинное целое
	Оценка 2	Числовое	Фиксированный	Длинное целое
	Оценка 3	Числовое	Фиксированный	Длинное целое
	Оценка 4	Числовое	Фиксированный	Длинное целое
	Результат	Текстовое	-	3

Таблица 2. Структура таблицы СТИПЕНДИЯ

Признак ключа	Имя поля	Тип поля	Формат поля	Размер поля
Ключевое	Результат	Текстовое	-	3
	Процент	Числовое	Процентный	Одинарное с плавающей точкой

Заполните вновь созданные таблицы СЕССИЯ и СТИПЕНДИЯ данными, как это показано на рисунке 1 и 2.

стипендия : таблица

результат	процент
нхр	0,00%
отл	200,00%
хор	100,00%
хр1	150,00%
	0,00%

Рисунок 1. Данные таблицы СТИПЕНДИЯ

сессия : таблица

номер	оценка1	оценка2	оценка3	оценка4	результат
16493	5,00	4,00	4,00	3,00	нхр
16593	4,00	4,00	5,00	5,00	хор
16693	5,00	5,00	5,00	5,00	отл
16793	5,00	5,00	5,00	4,00	хр1
16893	4,00	5,00	4,00	3,00	нхр
16993	4,00	4,00	3,00	4,00	нхр
17093	5,00	5,00	3,00	4,00	нхр
17193	4,00	4,00	5,00	4,00	хор
	0,00	0,00	0,00	0,00	

Рисунок 2. Данные таблицы СЕССИЯ

Задание 9.

Используя возможности MS Access, установите связи между созданными таблицами СТУДЕНТ, СЕССИЯ и СТИПЕНДИЯ базы данный СЕССИЯ.

1. Перейдите на вкладку Работа с базами данных нажмите кнопку Схема данных

2. Добавьте 3 таблицы в окно Схема данных. Для этого в окне Добавление таблицы из списка таблиц выберите и добавьте таблицы (Установите курсор на имя таблицы и нажмите кнопку Добавить). Закройте окно Добавление таблиц. Таблицы расположатся в окне Схема данных.

3. Установите связи между таблицами СТУДЕНТ и СЕССИЯ. Для этого установите указатель мыши на поле Номер таблицы СТУДЕНТ, нажмите левую клавишу мыши и удерживая ее протащите это поле на поле Номер таблицы СЕССИЯ; в появившемся диалоговом окне Изменение связей установите флажок Обеспечение целостности данных, обратите внимание, что тип отношений определен Один-к-одному, нажмите кнопку Создать.

4. Установите связь между таблицами СТИПЕНДИЯ и СЕССИЯ. Для этого протащите указатель мыши от поля Результат таблицы СТИПЕНДИЯ к полю Результат таблицы СЕССИЯ; в появившемся диалоговом окне Изменение связей установите флажок Обеспечение целостности данных, тип отношения определен, как Один-ко-многим, нажмите кнопку Создать.

В результате описанных действий окно Схема данных приобретает вид как на рисуноке 3.

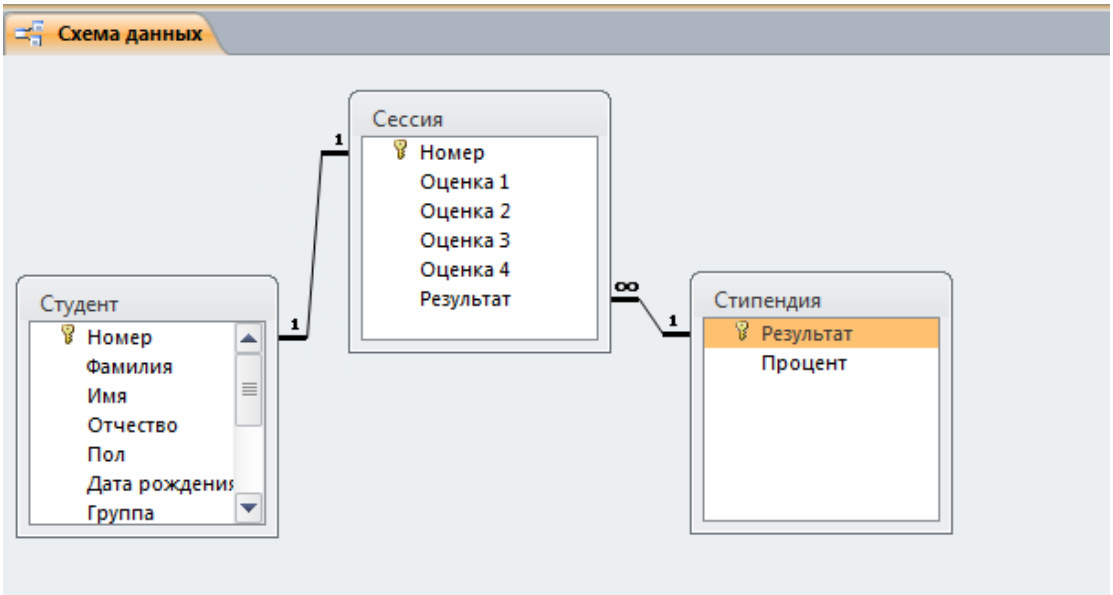


Рисунок 3. Установление связи между таблицами

5. Закройте окно Схема данных на запрос о сохранении ответьте Да.

Задание 10.

Постройте запрос, позволяющий выводить фамилию, имя, отчество и номер группы студентов, которым может быть назначена стипендия, а также размер назначаемой стипендии. Эти данные могут быть использованы при создании проекта приказа назначения студентов на стипендию по результатам экзаменационной сессии. Информация для получения таких данных содержится в трех связанных таблицах СТУДЕНТ, СЕССИЯ и СТИПЕНДИЯ базы данных СЕССИЯ.

1. Создайте запрос Выборка на основе связанных таблиц. Для этого выберите Тип объекта - Запросы;
2. На панели Создание нажмите кнопку Конструктор запросов ;
3. В окне Добавление таблицы выделите в списке таблицу СТУДЕНТ и щелкните на кнопке Добавить;
4. В том же списке выделите и добавьте таблицы СЕССИЯ и СТИПЕНДИЯ;
5. Закройте диалог щелчком по кнопке Закрыть.

Списки полей всех выбранных таблиц появляются в верхней части окна запроса Выборка. Между этими списками автоматически возникает соединительная линия, так как между таблицами уже установлена связь.

6. Присуноквойте запросу имя. Для этого выберите команду Сохранить из меню ФАЙЛ, и в окне Сохранение введите имя ПРОЕКТ ПРИКАЗА.

7. Включите поля из трех таблиц в запрос. Из таблицы СТУДЕНТ в бланк запроса по образцу (рисунок 4) в строку Поле перетащите следующие поля: Фамилия, Имя, Отчество, Группа. В следующее поле в запросе перетащите поле Процент из таблицы СТИПЕНДИЯ.

8. Установите Условие отбора для отбора студентов, подлежащих назначению на стипендию. В строке Условие отбора под полем Процент введите выражение >0.

9. Упорядочите выводимые в запросе данные по полю Фамилия в алфавитном порядке. Щелкните ячейку в строке Сортировка под полем Фамилия и в появившемся списке выберите По возрастанию.

10. Посмотрите сформированную запросом информацию. Для этого нажмите на кнопку Выполнить или выберите Режим таблицы в левом верхнем углу панели Конструктор.

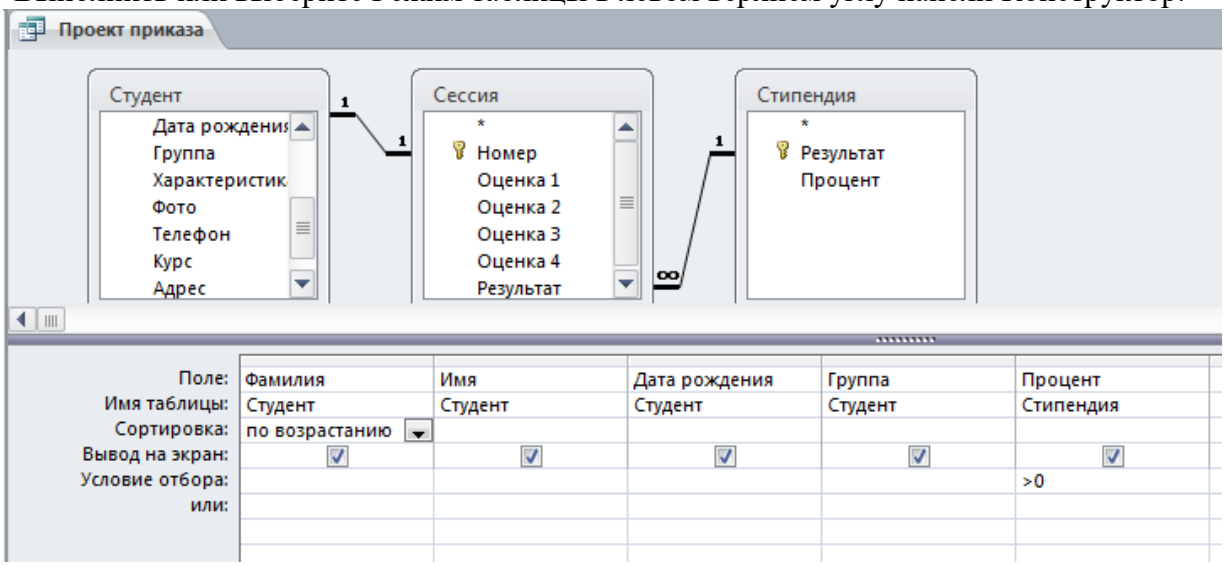


Рисунок 4. Многотабличный запрос на выборку

11. Закройте запрос, на вопрос о сохранение ответьте Да.

Задание 11.

Постройте отчет ПРОЕКТ ПРИКАЗА, основанный на сформированном ранее запросе ПРОЕКТ ПРИКАЗА, выбирающем из таблиц базы данных СТУДЕНТ, СЕССИЯ и СТИПЕНДИЯ информацию о студентах, которым по результатам экзаменационной сессии назначается стипендия, и о размере стипендии.

1. Выберите Тип объекта – Отчеты. На панели Создание выберите Мастер отчетов;

2. В диалоговом окне Создание отчета в поле Таблицы и запросы выберите запрос ПРОЕКТ ПРИКАЗА;

3. Перенесите все имеющиеся в запросе поля в выбранные, нажмите кнопку Далее;

4. В следующем диалоговом окне нажмите кнопку Далее ничего не выбирая;

5. В третьем диалоговом окне добавьте уровень группировки по полю Группа, для чего переместите поле Группа в правую часть окна;

6. В четвертом диалоговом окне установите сортировку по возрастанию для полей Фамилия и Имя;

7. Выберите макет отчета Ступенчатый, ориентацию Книжная и нажмите кнопку Далее;

8. Введите имя отчета ПРОЕКТ ПРИКАЗА, выберите дальнейшее действие Просмотреть отчет и нажмите кнопку Готово. В результате вы получите примерно такой отчет как нарисунок 5

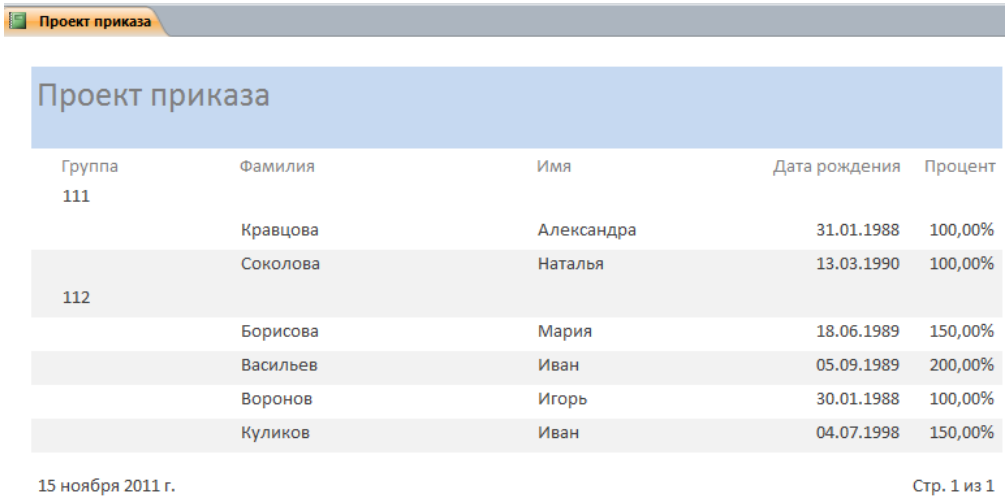


Рисунок 5 Отчет Проект приказа

9. Закройте отчет ПРОЕКТ ПРИКАЗА.

Сохраните результаты проделанной работы в своей папке под названием Сессия 2.

Задание 12.

1. Добавьте в таблицу Студент поля, имеющие следующие типы данных: текстовое, memo, объект OLE.

2. Установите свойства для этих полей: маска ввода, условие по умолчанию, условие на значение, сообщения об ошибке.

1. Откройте таблицу Студент в режиме конструктора и добавьте в нее следующие поля (Таблица 1):

Таблица 1. Перечень полей, добавляемых в таблицу Студент

Название поля	Тип поля	Описание
Характеристика	Поле MEMO	Внесите данные о характере, основных качествах личности, посещаемых студентом кружках, секциях и т.п.
Фото	Поле объекта OLE	Необходимо вставить фото студента, предварительно сохраненное в файле
Телефон	Текстовый	Введите код города и номер телефона

2. Заполните измененную таблицу данными:

2.1. В режиме таблицы в столбце Фото для каждого студента добавьте фотографию, для чего щелкните правой клавишей мыши по пустой ячейке и в контекстном меню выберите команду Вставить объект – Создать из файла. Укажите путь к фотографии (к любому графическому файлу). (Примечание! Изображение в режиме таблицы отображаться не будет. Увидеть изображение можно только в режиме формы).

2.2. В поле Характеристика заполнить характеристики на студентов, например, общительный, веселый, принимает участие в КВН, любит быть в центре внимания, обидчив.

2.3. Для поля Телефон создайте маску ввода. В режиме конструктора таблицы установите курсор в поле Телефон. Щелкните по значку … напротив свойства Маска ввода. В открывшемся окне щелкните по кнопке Список. Заполните поля по образцу (Рисунок 1). Нажмите кнопку Закрыть. В окне диалога Создание масок ввода установите курсор на созданную маску телефон и нажмите кнопку Готово.

2.4. В режиме таблицы заполните поле Телефон данными.

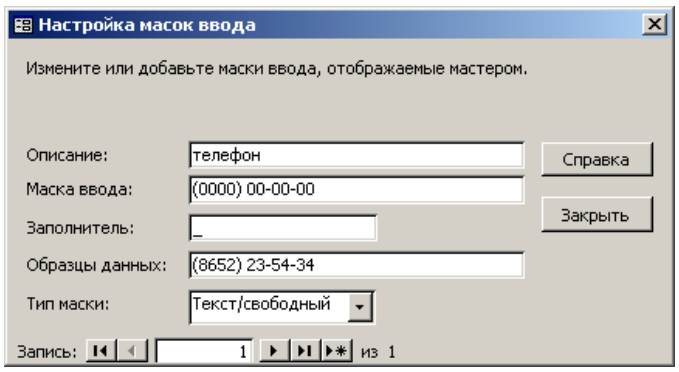


Рисунок 6. Образец для заполнения окна диалога «Настройка масок ввода»

3. В режиме конструктора установите для поля Пол свойство Значение по умолчанию «М». При внесении в таблицу данных о студенте в поле Пол автоматически будет появляться буква М, это даст экономию времени, так как только в половине случаев придется менять «М» на «Ж».

4. В режиме таблицы внесите в таблицу Студент данные о новых студентах: Куликове Иване Ивановиче и Соколовой Наталье Викторовне. Предварительно уберите ранее установленную связь между таблицами Сессия и Студент на схеме данных.

5. Откройте таблицу Сессия в режиме Конструктора. Установите для полей Оценка1, Оценка2, Оценка3, Оценка4 свойство Условие на значение: >=2 and <=5. Если по ошибке будет введено другое число, то Access должен выводить сообщение. Текст сообщения введите в свойство Сообщения об ошибке (например, текст может выглядеть так: Введите оценку от 2 до 5).

6. Заполните таблицу Сессия данными по Куликову И.И и Соколовой Н.В. Попытайтесь ввести оценку 6.

7. Восстановите связи между таблицами Сессия и Студент на схеме данных.

Задание 13.

Используя данные таблиц Студент и Сессия, создайте запрос, вычисляющий среднюю оценку за сессию каждого студента. Вычисления произведите в вычисляемом поле при помощи Построителя выражений.

1. Создайте запрос в режиме конструктора запросов. Добавьте в верхнюю часть бланка запроса таблицы Сессия и Студент. Внесите в нижнюю часть бланка запроса поля Номер, Фамилия, Имя из таблицы Студент (Рисунок 7).

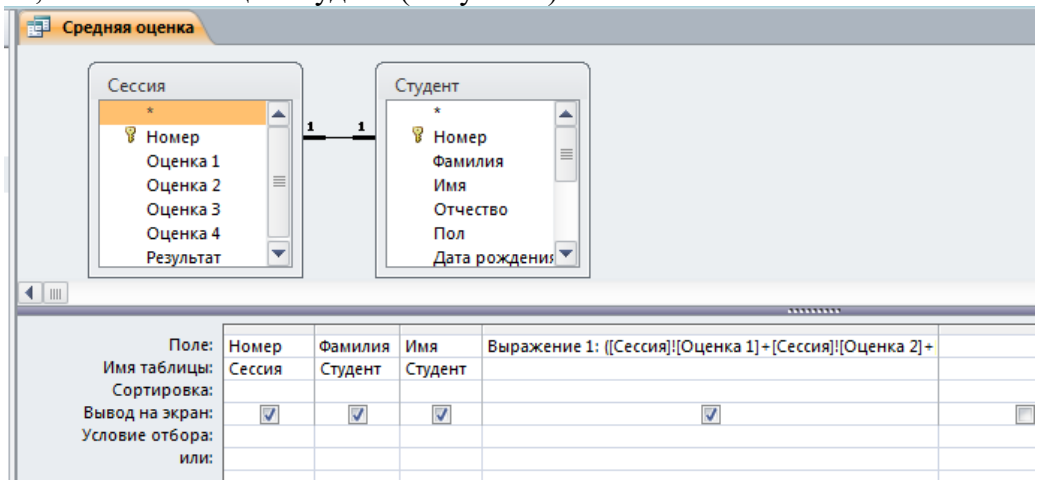


Рисунок 7. Запрос на выборку, содержащий вычисляемое поле

2. Для создания выражения Средняя оценка щелкните в верхней строке свободного столбца и вызовите Построитель выражения с помощью кнопки Построить на панели Конструктор.

3. Выражение вводится с помощью мыши и средств, предоставляемых Построителем выражений (Рисунок 3).

3.1. В окне диалога Построитель выражений выберите Таблицы – Сессия – Оцека1 – двойной щелчок мыши. Выбранное поле добавится в верхнюю часть Построителя. Нажмите «+». Затем выберите Оценку2 и вставьте ее в выражение, нажмите «+» и так далее, пока все четыре оценки не будут просуммированы в выражении.

3.2. Возьмите созданное выражение в круглые скобки и разделите его на «4». Нажмите ОК.

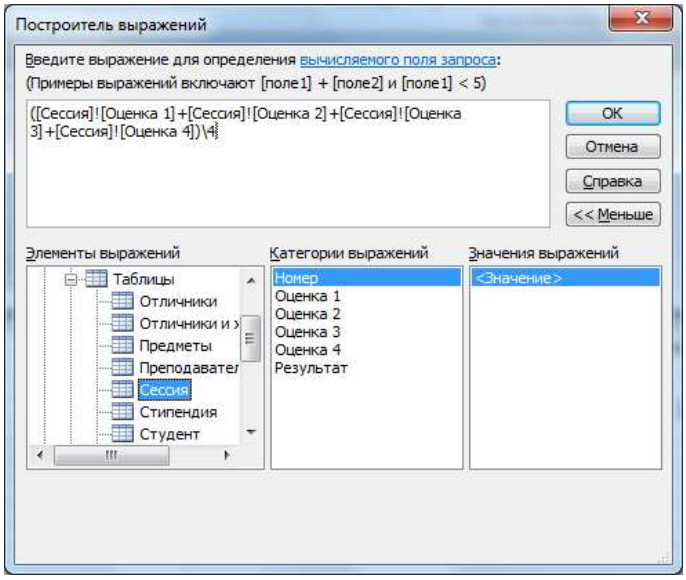


Рисунок 8. Окно построителя выражений

4. Для созданного вычисляемого поля необходимо ввести название Средняя оценка. Для этого, не выходя из конструктора запросов, откройте свойства с помощью контекстного меню и в свойстве Подпись напечатайте Средняя оценка.

5. Запустите запрос на выполнение и сохраните его.

Задание 14.

Создайте таблицу Отличники при помощи запроса на создание таблицы.

Технология работы

1. Создайте следующий запрос в режиме конструктора (Рисунок9).

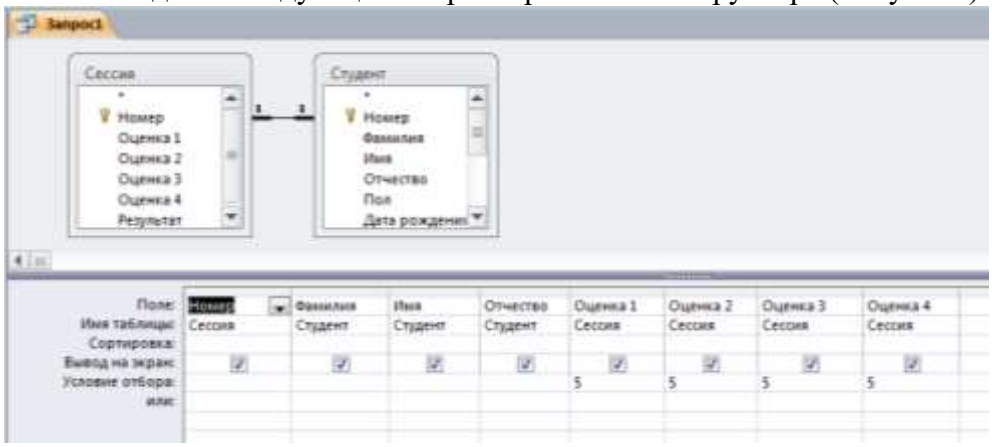


Рисунок 9. Запрос на выборку для создания таблицы Отличники

2. Затем преобразуйте его в «запрос на создание таблицы», нажав кнопку Создание таблицы панели Работа с запросами. В открывшемся диалоговом окне введите название таблицы Отличники. Выберите «в той же базе данных».

3. Выполните запрос. Access предупредит о создании таблицы, выберите ОК.

4. Закройте запрос с сохранением. Перейдите к объектам Таблицы и убедитесь в существовании таблицы Отличники. Просмотрите данные таблицы Отличники.

Задание 15.

Используя запрос на добавление, добавьте записи к таблице Отличники и хорошисты.

Технология работы

1. Перед созданием следующего запроса необходимо сделать копию таблицы Отличники и назвать ее Отличники и хорошисты. Для этого на вкладке Таблицы выделите таблицу Отличники, примените команду Копировать - Вставить. Назовите созданную копию Отличники и хорошисты.

2. Для заполнения таблицы необходимо выбрать студентов без троек и добавить их к отличникам. Для этого создайте запрос в режиме конструктора на основе таблиц Сессия и Студент, аналогичный предыдущему, с условиями отбора >=4 для полей Оценка1, Оценка2, Оценка3, Оценка4.

3. Преобразуйте запрос в «запрос на добавление» с помощью кнопки Добавление . В открывшемся окне выберите таблицу Отличники и хорошисты - ОК.

4. Запустите запрос. На предупреждение о добавлении записей в таблицу, нажмите ОК. Убедитесь, что в таблицу Отличники и хорошисты добавились новые записи.

Сохраните результаты проделанной работы в своей папке под названием Сессия 3.

Задание для развития и контроля владения компетенциями:

1. Расскажите о назначении типов данных Memo и объект OLE.
2. Какие свойства полей вам известны, расскажите об их назначении.
3. Опишите технологию создания маски для ввода данных.
4. Опишите технологию создания запросов с вычисляемыми полями.
5. Расскажите о назначении Построителя выражений и опишите технологию создания выражений в запросах с его помощью.
6. Опишите технологию создания запроса «На создание таблицы. Расскажите о назначении этого вида запросов.
7. Опишите технологию создания запроса «На добавление». Расскажите о назначении этого вида запросов.
8. Опишите технологию создания структуры таблицы в режиме конструктора.
9. Опишите технологию создания связей между таблицами в реляционной базе данных.
10. Назовите известные вам типы связей и дайте им характеристику.
11. Опишите технологию создания многотабличных запросов.

Раздел 5. Интернет

Тема 5.1. Организация глобальных сетей. Интернет как глобальная ИС

Цель: Научиться пользоваться поисковыми системами, обмену сообщениями в сети Интернет, создание электронной почты, настройка почтовых клиентов.

Теоретический материал:

Сеть Интернет растет очень быстрыми темпами, поэтому найти нужную информацию среди сотен миллиардов Web-страниц и сотен миллионов файлов становится все сложнее. Для поиска информации используются специальные поисковые системы, которые содержат постоянно обновляемую информацию о местонахождении Web-страниц и файлов на сотнях миллионов серверов Интернета (рисунок 1).

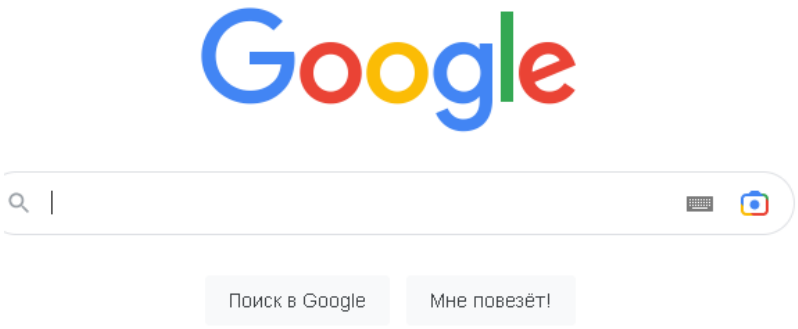


Рисунок 1 – Поисковая система Google

Поисковые системы содержат тематически сгруппированную информацию об информационных ресурсах Всемирной паутины в базах данных. Специальные программы-роботы периодически «обходят» Web-серверы Интернета, читают все встречающиеся документы, выделяют в них ключевые слова и заносят в базу данных Интернет-адреса документов.

Большинство поисковых систем разрешают автору Web-сайта самому внести информацию в базу данных, заполнив регистрационную анкету. В процессе заполнения анкеты разработчик сайта вносит адрес сайта, его название, краткое описание содержания сайта, а также ключевые слова, по которым легче всего будет найти сайт.

Поиск по ключевым словам. Поиск документа в базе данных поисковой системы осуществляется с помощью введения запросов в поле поиска.

Запрос должен содержать одно или несколько ключевых слов, которые являются главными для этого документа. Например, для поиска самих систем поиска в Интернете можно в поле поиска ввести ключевые слова «российская система поиска информации Интернет»

Через некоторое время после отправки запроса поисковая система вернет список Интернет-адресов документов, в которых были найдены заданные ключевые слова. Для просмотра этого документа в браузере достаточно активизировать указывающую на него ссылку

Если ключевые слова были выбраны неудачно, то список адресов документов может быть слишком большим (может содержать десятки и даже сотни тысяч ссылок). Для того чтобы уменьшить список, можно в поле поиска ввести дополнительные ключевые слова или воспользоваться каталогом поисковой системы.

Одной из наиболее полных и мощных поисковых систем является Google (www.google.ru), в базе данных которой хранятся 8 миллиардов Web-страниц и каждый месяц программы-роботы заносят в нее 5 миллионов новых страниц. В Рунете (российской части Интернета) обширные базы данных, содержащие по 200 миллионов документов, имеют поисковые системы Яndex (www.yandex.ru) (рисунок 2) и Rambler (www.rambler.ru) (рисунок 3).

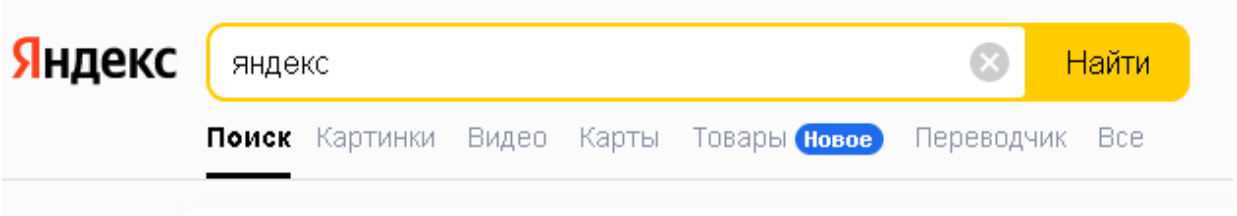


Рисунок 2 – Поисковая система яндекс.

Поиск в иерархической системе каталогов. В базе данных поисковой системы Web-сайты группируются в иерархические тематические каталоги, которые являются аналогами тематического каталога в библиотеке.

Тематические разделы верхнего уровня, например: Интернет, Компьютеры, Наука и образование и т. д., содержат вложенные каталоги. Например, каталог Интернет может содержать подкаталоги Поиск, Почта и др.

Поиск информации в каталоге сводится к выбору определенного каталога, после чего пользователю будет представлен список ссылок на Интернет-адреса наиболее посещаемых и содержательных Web-сайтов. Каждая ссылка обычно аннотирована, т. е. содержит короткий комментарий к содержанию документа.

Наиболее полный многоуровневый иерархический тематический каталог русскоязычных Интернет-ресурсов имеет поисковая система Апорт (www.aport.ru). Каталог содержит подробную аннотацию содержания Web-сайтов и указание на их географическое положение.

Поиск файлов. Для поиска файлов на серверах файловых архивов существуют специализированные поисковые системы, в том числе поисковая система FileSearch (www.filesearch.ru). Для поиска файла необходимо ввести имя файла в поле поиска, и поисковая система выдаст Интернет-адреса серверов файловых архивов, на которых хранится файл с заданным именем.

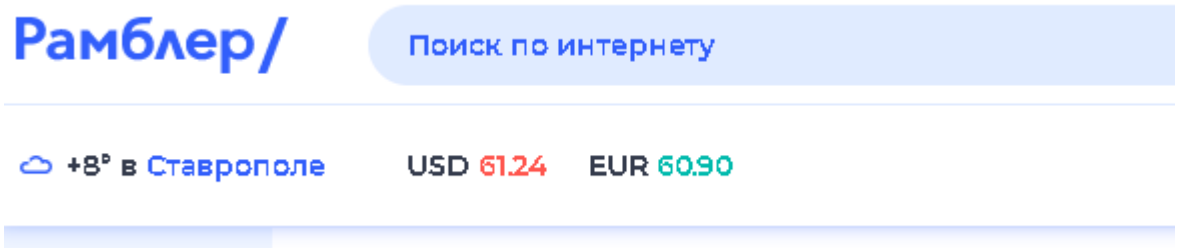


Рисунок 3 – Поисковая система Рамблер.

Поиск информации в русскоязычной части Интернета с помощью наиболее поисковых систем: Google, Rambler, Апорт, Япс1ех и файловой поисковой системы Research можно производить с использованием интегрированной поисковой системы Gogle.ru. Для этого достаточно ввести ключевые слова в строку поиска, с помощью переключателей установить тип необходимой информации и щелкнуть по кнопке с названием поисковой системы Gogle.ru. Для этого достаточно ввести ключевые слова в строку поиска, с помощью переключателей установить тип необходимой информации и щелкнуть по кнопке с названием поисковой системы.

Интернет в целом и Всемирная паутина, в частности, предоставляют абоненту доступ к тысячам серверов и миллионам Web-страниц, на которых хранится невообразимый объем информации. Как не потеряться в этом «информационном океане»? Для этого необходимо научиться искать и находить нужную информацию в сети.

Как уже было сказано, существуют три основных способа поиска информации в Интернете.

1. Указание адреса страницы. Это самый быстрый способ поиска, но его можно использовать только в том случае, если точно известен адрес документа.

2. Передвижение по гиперссылкам. Это наименее удобный способ, так как с его помощью можно искать документы, только близкие по смыслу текущему документу. Если текущий документ посвящен, например, музыке, то, используя гиперссылки этого документа, вряд ли можно будет попасть на сайт, посвященный спорту.

3. Обращение к поисковому серверу (поисковой системе). Использование поисковых серверов - наиболее удобный способ поиска информации. В настоящее время в русскоязычной части Интернета популярны следующие поисковые серверы:

Существуют и другие поисковые системы. Например, эффективная система поиска реализована на сервере почтовой службы mail.ru (рисунок 4).

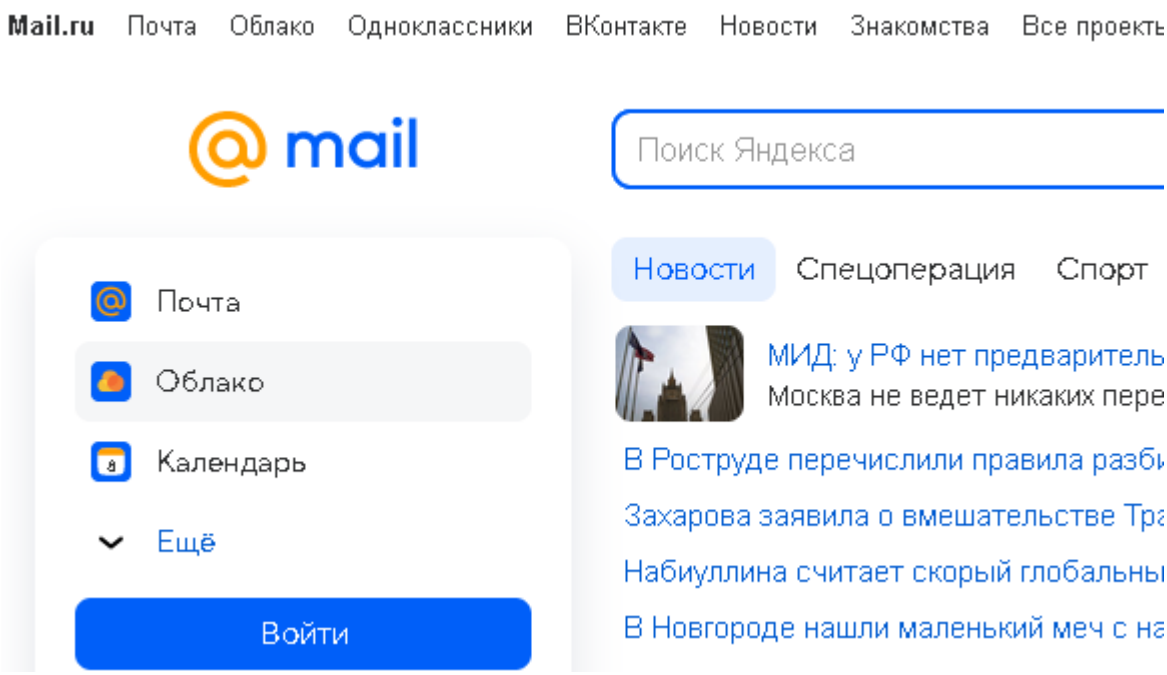


Рисунок 4 – Поисковая система mail.ru

Наиболее доступным и удобным способом поиска информации во Всемирной паутине является использование поисковых систем. При этом поиск информации можно осуществлять по каталогам, а также по набору ключевых слов, характеризующих отыскиваемый текстовый документ.

Рассмотрим использование поисковых серверов более подробно. Поисковый сервер содержит большое количество ссылок на самые различные документы, и все эти ссылки систематизированы в тематические каталоги. Например: спорт, кино, автомобили, игры, наука и др. Причем эти ссылки устанавливаются сервером самостоятельно, в автоматическом режиме путем регулярного просмотра всех появляющихся во Всемирной паутине Web-страниц. Кроме того, поисковые серверы предоставляют пользователю возможность поиска информации по ключевым словам. После ввода ключевых слов поисковый сервер начинает просматривать документы на других Web-серверах и выводить на экран ссылки на те документы, в которых встретились указанные слова. Обычно результаты поиска сортируются по убыванию специального рейтинга документов, который показывает, насколько полно заданный документ отвечает условиям поиска или насколько часто он запрашивается в сети.

Организация глобальных сетей, включая Интернет, является сложным процессом, который требует сотрудничества множества организаций и технологий.

1. Инфраструктура:

- Глобальные сети, такие как Интернет, основаны на физической инфраструктуре, включающей в себя оптоволоконные кабели, сетевые узлы, маршрутизаторы и другое оборудование.
- Сети развиваются на основе многоуровневой архитектуры, такой как OSI (модель взаимодействия открытых систем), которая определяет способы передачи данных и коммуникации между различными компонентами сети.

2. Протоколы:

- Протоколы являются правилами и стандартами, которые определяют, как данные передаются и обрабатываются в сети.
- Примеры протоколов включают TCP/IP (основной протокол Интернета), HTTP (протокол передачи гипертекста), SMTP (протокол передачи почты) и многие другие.

3. Доменные имена:

- Доменные имена используются для идентификации ресурсов в Интернете (веб-сайтов, серверов электронной почты и т.д.).

- DNS (система доменных имён) отвечает за преобразование человеко-понятных доменных имен в числовые IP-адреса, используемые компьютерами для поиска ресурсов.

4. Поставщики услуг Интернета (Интернет-провайдеры):

- Интернет-провайдеры предоставляют доступ к Интернету для отдельных пользователей, домов, организаций и провайдеров услуг.
- Они обеспечивают физическое подключение к глобальной сети и предоставляют средства коммуникации.

5. Безопасность:

- Безопасность играет важную роль в организации глобальных сетей. Она включает в себя меры защиты от хакеров, вредоносного программного обеспечения, фишинга и других киберугроз.
- Применение шифрования, брандмауэров, аутентификации и других методов помогает обеспечить безопасность данных и коммуникаций.

6. Международные организации:

- Существует несколько международных организаций, которые координируют развитие и стандартизацию глобальных сетей. Одной из таких организаций является Интернет-корпорация по присвоению имен и номеров (ICANN).

7. Развитие технологий:

- Технологии постоянно развиваются, и новые инновации позволяют сделать глобальные сети более быстрыми, надежными и доступными.

Организация глобальных сетей - это сложный и многогранный процесс, требующий согласованной работы различных участников для обеспечения стабильности и эффективности сетей, таких как Интернет.

Тема 5.2. WWW – Всемирная паутина

«WWW» расшифровывается как «World Wide Web», что на русском языке означает «Всемирная паутина» или «Всемирная сеть». Это термин, который часто используется для обозначения системы информационных ресурсов, связанных гиперссылками и URL-адресами, доступных через интернет.

Всемирная паутина (WWW) является одной из основных составляющих интернета и представляет собой огромное количество веб-сайтов, страниц, документов, изображений и другого контента, которые объединены между собой гиперссылками. Каждый веб-сайт имеет уникальный URL (Uniform Resource Locator), который позволяет пользователям находить и доступно обмениваться информацией.

Всемирная паутина значительно упрощает доступ к информации, так как пользователи могут переходить с одной страницы на другую, переходить по ссылкам, выполнять поиск и изучать разнообразные темы. Технологии, лежащие в основе Всемирной паутины, включают в себя HTML (язык гипертекстовой разметки), HTTP (протокол передачи гипертекста), браузеры и другие.

Всемирная паутина сыграла значительную роль в превращении интернета из простой коммуникационной сети в мощный инструмент для обмена информацией, коммуникаций, бизнеса и образования.

Инструменты для разработки веб-сайтов

Существует множество инструментов для разработки веб-сайтов, которые помогают создавать и настраивать интерфейсы, функциональность и контент веб-сайтов.

1. HTML и CSS Редакторы:

- Visual Studio Code: Мощный редактор кода с подсветкой синтаксиса, автозаполнением и расширениями для HTML и CSS.
- Sublime Text: Легкий и быстрый редактор с подсветкой синтаксиса, множеством плагинов и настраиваемой средой.

2. Графические Редакторы:

- Adobe Photoshop: Популярное приложение для обработки изображений и создания визуальных элементов веб-сайта.
- Adobe XD: Инструмент для проектирования пользовательских интерфейсов и прототипирования веб-сайтов.

3. Фреймворки и Библиотеки:

- Bootstrap: Популярный фреймворк для создания отзывчивых и стилизованных веб-сайтов.
- jQuery: Библиотека JavaScript для упрощения работы с DOM и анимациями.

4. CMS (Системы Управления Контентом):

- WordPress: Платформа для создания и управления веб-сайтами с использованием готовых тем и плагинов.
- Joomla: Еще одна популярная CMS с расширяемой архитектурой.

5. Веб-браузеры и Инструменты Разработчика:

- Google Chrome DevTools: Встроенный набор инструментов для разработчика в браузере Chrome для отладки и анализа веб-страниц.
- Mozilla Firefox Developer Edition: Версия Firefox с инструментами для разработчика, включая инспектор, консоль и отладчик.

6. Интегрированные Среды Разработки (IDE):

- WebStorm: IDE для разработки веб-приложений с поддержкой JavaScript, HTML, CSS и других языков.
- Visual Studio: Интегрированная среда разработки Microsoft с возможностью создания веб-приложений и сайтов.

7. Прототипирование и Дизайн:

- Figma: Инструмент для дизайна интерфейсов, прототипирования и совместной работы в реальном времени.
- Sketch: Программа для создания макетов и дизайна интерфейсов для macOS.

Тема 5.3. Инструменты для разработки web – сайтов

Цель: Научиться создавать простейший гипертекстовый документ средствами текстового редактора Notepad++. Научиться использовать теги форматирования шрифта и абзаца.

Теоретический материал:

HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML).

Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML до 5-й версии определялся как приложение SGML (стандартного обобщённого языка разметки по стандарту ISO 8879). Спецификации HTML5 формулируются в терминах DOM (объектной модели документа).

Язык XHTML является более строгим вариантом HTML, он следует синтаксису XML и является приложением языка XML в области разметки гипертекста. Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

1. Структура теговой системы

Тег это базовый элемент языка HTML. То есть HTML состоит именно из тегов и для того чтобы учить HTML, достаточно изучить основные теги, которые чаще всего используются. Если сравнивать с примером из реальной жизни, то тег в HTML - это как кирпич для дома. Кирпич это основной элемент, который используется для строительства дома. То же

самое и в HTML, тег это базовый элемент для создания html страницы. С помощью тега мы указываем браузеру, как правильно отобразить содержание html страницы. Тег – это символ разметки, имеющий имя и атрибут. Имя - это строка, состоящая из букв, цифр, строчек и тире (не более 72 символов) Атрибут – это конструкция вида: имя атрибута = значение. Теги пишутся в треугольных скобках <…..>

Существует два типа тегов это парные теги и одинарные теги:

• Парные теги это теги, которые состоит из открывающего тега и закрывающего тега. По короче это теги, которые имеют закрывающего тега. Примеры: <html> и </html>; <title> и </title>; <head> и <head>; <body> и </body>; и ;

• Одинарные теги это теги, которые не имею закрывающего тега. Примеры одинарных тегов: <br />. <hr />

У html страницы есть базовая структура. Базовая структура на всех страницах сайта, выглядит одинаково. Единственное что может отличаться в базовой структуре страниц от сайта к сайту, это элемент DOCTYPE. Вот как выглядит базовый каркас html страницы:

В начале страницы, перед тегом html всегда указываем DOCTYPE, то есть тип документа. Каждая html страница начинается с тега <html> и заканчивается тегом </html> и состоит из заголовка <head></head> и тела <body></body>. Внутри заголовка пишется служебные элементы, которые не отображаются на странице в браузере. В основном это:

• Мета теги. Главные из них это мета-тег кодировки, через которого указываем кодировку страницы (<meta http-equiv="Content-type" content="text/html; charset=utf-8"/>), мета теги для поисковых систем, это мета-тег с описанием страницы (description) и ключевые слова для страницы (keywords).

• Заголовок страницы, который указывается внутри тега <title></title>. Этот заголовок отображается во вкладке браузера.

• Блок с внутренними стилями. <style type=”text/css”> Внутренние стили. </style>

• Одинарный тег <link />, через которого подключаем внешние файлы.

• А также блок <scriptsrc="library.js" type="text/javascript"> </script> в котором пишутся различные скрипты JavaScript.

Внутри тега <body></body> пишется контент страницы. Контент может содержать текст, изображения, таблицы, списки, ссылки и другие элементы которые отображаются на странице в браузере. Имена тегов можно писать в любом регистре, то есть если напишем <BODY>, <bOdY>,<Body> или <body>, то результат будет один и тот же, но рекомендуется всегда писать в нижнем регистре, то есть <body>.

Когда Вы пишете любой текст на странице, будь это просто абзац или небольшая поясняющая строка под изображением, необходимо этот текст помещать внутрь тега .

Практический материал:

Задание 1.

Создать HTML-документ.

1. Создайте папку (в качестве имени папки выберите свою фамилию). Запустить редактор Notepad++. Ввести приведенные ниже стандартные теги разметки страницы теги

<HTML><HEAD><TITLE>Мой первый HTML-документ</TITLE></HEAD>

<BODY>ПРИВЕТ МИР!!!</BODY></HTML>

2. Сохранить документ под именем 1.html в созданной вами папке. Откройте папку, кликните правой кнопкой мыши по файлу 1.html и выберите команду «Открыть с помощью» и выберите удобный для просмотра браузер. В окне браузера вы можете посмотреть как будет выглядеть ваш файл.

3. Отредактируйте документ 1.html, кликните правой кнопкой мыши и выберите команду «Открыть с помощью» и выберите программу Блокнот. После слов ПРИВЕТ МИР!!! поставьте тег
 (тег перехода на новую строку), и наберите текст «Программирование сайтов и Web-дизайн». Не забывайте сохранять документ и обновлять браузер.

4. Втег<BODY> вставить атрибуты TEXT(цвет текста на всейhtml-странице) и BGCOLOR(цвет заливки на всейhtml-странице), и установите любое значения цветов

(безопасные коды цветов можно найти в интернете). Поставьте горизонтальную линию в документе с помощью соответствующего тега <HRSIZE=1>, где атрибут SIZE определяет толщину линии в пикселях.

Задание 2.

Создайте заголовки по образцу

Анкета

Петров Владимир Владимирович

17 лет

козерог

студент

люблю рок-музыку

Указания к выполнению

Откройте в Notepad++ файл 1.html и создайте анкету с помощью заголовочных тегов <H1-6 ALIGN=”LEFT| CENTER| RIGHT”> текст </H1-6>, с помощью атрибута ALIGN выровняйте текст по центру.

Задание 3.

Создайте текст на HTML-странице по образцу

ПАРУС

Белеет парус одинокой В тумане моря голубом!.. Что ищет он в стране далекой? Что кинул он в краю родном? Играют волны — ветер свищет, И мачта гнется и скрипит... Увы! он счастья не ищет, И не от счастья бежит! Под ним струя светлей лазури, Над ним луч солнца золотой... А он, мятежный, просит бури, Как будто в бурях есть покой!

В файле 1.html после анкеты наберите текст стихотворения «Парус» и отформатируйте шрифт текста по образцу. Название стихотворения заголовок 3-го уровня (<H3></H3>), первая строчка стихотворения выделена жирным шрифтом с помощью тегов <B></B> или <STRONG></STRONG>; вторая строчка выделена курсивом с помощью тега <I></I>; третья строчка подчеркнута с помощью тега <U></U>>; четвертая строчка перечеркнута с помощью тега <S></S> или <DEL></DEL>; пятая, шестая, седьмая и восьмая строчки выделены с помощью тегов с размерами шрифтов <FONTSIZE=1|3|5|7></FONT> соответственно; девятая строчка выделена с помощью тега <SMALL></SMALL>; десятая строчка создана при помощи тега <BIG></BIG>; одиннадцатая и двенадцатая строчки раскрашены с помощью тегов <FONTCOLOR=YELLOW|GREEN></FONT> соответственно. В конце стихотворения добавьте фамилию и инициалы автора стихотворения моноширинным шрифтом (шрифт имитирующий печатную машинку), с помощью контейнеров тегов <TT></TT>,<KBD></KBD>или<SAMP></SAMP>.

Задание 4.

Создание формулу по образцу $S_{квадрата}=a^2$

Указания к выполнению

Для создания нижнего индекса используйте теги в контейнер заключите индекс, для возведения в степень используйте теги верхнего индекса . Увеличьте шрифт не индексного текста до 5.

Задание 5.

Создание абзацев.

Указания к выполнению

Скопируйте любой текст и вставьте в файл 1.html после формулы, отформатируйте абзацы с помощью тега <P></P> для выравнивания абзаца используйте атрибут ALIGN=LEFT| CENTER| RIGHT | JUSTIFY.

Задание 6.

Создание списков.

Указания к выполнению

В файле 1.html после отредактированного текста создайте списки.

Для создания нумерованного списка используют теги <OL><LI>.

<OL><LI>Элемент списка 1 </LI>

<LI>Элемент списка 2</LI>

<LI>Элемент списка 3</LI></OL>

Список по умолчанию цифровой, но его можно сделать и буквенным с помощью атрибута TYPE .

<OLTYPE=a><LI>Элемент списка 1 </LI>

<LI>Элемент списка 2</LI>

<LI>Элемент списка 3</LI></OL>

Для создания маркированного списка используют теги <UL><LI>.

<UL>LI>Элемент списка 1 </LI>

<LI>Элемент списка 2</LI>

<LI>Элемент списка 3</LI></UL>

Маркеры в списке можно сделать разнообразными с помощью атрибута TYPE=circle|disc|square .

<UL><LI>Элемент списка 1 </LI>

<LI>Элемент списка 2</LI>

<LI>Элемент списка 3</LI></UL>

Для создания списков определений используют три тэга:

<DL>- начало/конец списка.

<DT>- начало/конец конкретного термина.

<DD>- начало/конец поясняющей статьи термина.

<DL><DT>ГРИБЫ</DT>

<DD>ОПЯТА</DD>

<DD>ГРУЗДИ</DD></DL>

Создайте формулу соляной кислоты и общий вид квадратного уравнения. Создайте смешанный список по образцу Я знаю как оформлять:

1. Шрифты
2. Размер
3. Цвет
4. Гарнитуру
5. Индексы
1. Заголовки
6. От 1-го до 6-го уровня
1. Абзацы
7. Выравнивание
8. Разрыв строк внутри абзаца
9. С использованием переформатирования.

Задание 7.

Создание аббревиатуры

Для создания аббревиатуры можно использовать либо контейнер тегов<ABBR></ABBR>, либо <ACRONYM></ACRONYM>, разница между ними лишь в том что <ACRONYM></ACRONYM>это элементы только спецификацииHTML4.01, А ТЕГИ <ABBR></ABBR>спецификацийHTML4.01/5.0 и расшифровать аббревиатуры можно с помощью атрибутаTITLE, например:

<ABBR TITLE=”Hyper Text Markup Language”>HTML</ABBR>

<ACRONYM>HTML</ACRONYM>.

Задание 8.
Подготовить отчет в форме доклада.

Тема 5.4. Создание сайта. Создание таблиц и списков на web – странице

Цель: Изучение тегов <TABLE>, <TH>, <TD> и их атрибутов приобретение умения задания таблицы, ее размера, шрифта записей в ячейках таблицы, расположения и вида таблицы.

Создание веб-сайта может быть разделено на несколько основных шагов.

1. Определение Цели и Целевой Аудитории:

- Определите, для чего вы создаете веб-сайт: информационный ресурс, интернет-магазин, портфолио и т.д.
- Определите свою целевую аудиторию: кто будут ваши пользователи и что именно они ищут.

2. Планирование и Проектирование:

- Создайте структуру сайта (навигационное меню, страницы, разделы).
- Сделайте макеты веб-страниц для каждой секции, учитывая дизайн, расположение контента и элементов.

3. Выбор Инструментов:

- Определитесь с инструментами, которые вы будете использовать для создания сайта (HTML, CSS, фреймворки, CMS и т.д.).

4. Разработка:

- Создайте HTML-страницы и добавьте контент.
- Примените CSS для стилизации страницы.
- Добавьте интерактивность с помощью JavaScript, если это требуется.

5. Адаптивный Дизайн:

- Сделайте ваш сайт отзывчивым, чтобы он хорошо отображался на различных устройствах и экранах.

6. Тестирование:

- Протестируйте сайт на разных браузерах и устройствах.
- Убедитесь, что все ссылки, формы и функциональность работают корректно.

7. Размещение на Хостинге:

- Выберите хостинг-провайдера и зарегистрируйте доменное имя для вашего сайта.
- Загрузите файлы сайта на хостинг.

8. Оптимизация и SEO:

- Оптимизируйте изображения и контент для ускорения загрузки страниц.
- Примените базовую оптимизацию для поисковых систем (SEO), чтобы ваш сайт был виден в результатах поиска.

9. Запуск и Обслуживание:

- После тщательного тестирования и оптимизации, запустите ваш сайт.
- Регулярно обновляйте контент и следите за работоспособностью сайта.

10. Продвижение: - Продвигайте ваш сайт через социальные сети, контент-маркетинг, рекламу и другие каналы.

Не забывайте, что создание веб-сайта — это итеративный процесс. Вам, возможно, потребуется вносить изменения и улучшать сайт по мере его развития и отзывов пользователей.

Создание таблиц и списков

Создание таблиц и списков является важной частью разработки веб-сайта, так как это позволяет структурировать и представлять информацию более организованным и понятным образом.

Создание Таблиц:

Для создания таблиц на веб-странице используется элемент <table>, а также элементы <tr> (строка таблицы), <th> (ячейка заголовка) и <td> (ячейка данных). Вот пример кода для создания простой таблицы:

```
html
<table>
  <tr>
    <th>Заголовок 1</th>
    <th>Заголовок 2</th>
  </tr>
  <tr>
    <td>Ячейка 1</td>
    <td>Ячейка 2</td>
  </tr>
  <tr>
    <td>Ячейка 3</td>
    <td>Ячейка 4</td>
  </tr>
</table>
```

Создание Нумерованных Списков:

Для создания нумерованных списков используется элемент <ol>, а элементы списка обозначаются <li>. Вот пример:

```
html
<ol>
  <li>Первый элемент</li>
  <li>Второй элемент</li>
  <li>Третий элемент</li>
</ol>
```

Создание Маркированных Списков:

Для создания маркированных списков используется элемент <ul>, а элементы списка также обозначаются <li>. Вот пример:

```
html
<ul>
  <li>Элемент списка

продолжай
```

Создание Описательных Списков:

Описательные списки используются для предоставления описаний элементов. Они состоят из пар элементов <dt> (термин) и <dd> (определение). Вот пример:

```
html
<dl>
  <dt>HTML</dt>
  <dd>Язык разметки для создания веб-страниц.</dd>
  <dt>CSS</dt>
  <dd>Каскадные таблицы стилей, используются для стилизации веб-содержимого.</dd>
</dl>
```

Вложенные Списки:

Вы также можете создавать вложенные списки, то есть списки, внутри которых есть другие списки. Вот пример:

```
html
<ul>
  <li>Главный элемент 1
    <ul>
      <li>Вложенный элемент 1</li>
      <li>Вложенный элемент 2</li>
    </ul>
  </li>
  <li>Главный элемент 2
    <ul>
      <li>Вложенный элемент 3</li>
      <li>Вложенный элемент 4</li>
    </ul>
  </li>
</ul>
```

Стилизация Таблиц и Списков с помощью CSS:

Вы можете применять стили к таблицам и спискам, чтобы изменить их внешний вид и расположение на странице. Например:

```
css
/* Стилизация таблицы */
table {
  width: 100%;
  border-collapse: collapse;
}

th, td {
  border: 1px solid #ccc;
  padding: 8px;
}

/* Стилизация списка с точками */
.dot-list {
  list-style-type: disc;
}

/* Стилизация списка с номерами */
.ordered-list {
  list-style-type: decimal;
}

/* Стилизация описательного списка */
dl {
  margin-top: 10px;
}

dt {
  font-weight: bold;
}

dd {
```

margin-left: 20px;
}
Практический материал

Задание 1. Создание таблицы.

Указания к выполнению

Откройте Notepad++ и наберите следующий текст

<HTML>
<TITLE>таблицав HTML</TITLE><HEAD></HEAD>
<BODY ><H3 ALIGN=CENTER>ТАБЛИЦА № 1</H3>

1. Для создания таблицы используется тег <TABLE>, атрибуты данного тега определяют:

ALIGN- выравнивание таблицы
BACKGROUND- Задает фоновый рисунок в таблице
BGCOLOR - Цвет фона таблицы
BORDER - Толщина рамки в пикселях
BORDERCOLOR - Цвет рамки
CELLPADDING - Отступ от рамки до содержимого ячейки
CELLSPACING - Расстояние между ячейками
COLS - Число колонок в таблице
FRAME - Сообщает браузеру, как отображать границы вокруг таблицы
HEIGHT - Высота таблицы
RULES - Сообщает браузеру, где отображать границы между ячейками
SUMMARY - Краткое описание таблицы
WIDTH - Ширина таблицы

<TABLE ALIGN=CENTER BORDER="5" BGCOLOR =CCCC00 CELLPADDING =1
CELLSPACING=1
BORDERCOLOR=GOLD><CAPTION>ТАБЛИЦАИСТИННОСТИ</CAPTION>

2. Теги <CAPTION>определяет название таблицы. Тег<THEAD></THEAD>определяет верхнюю часть таблицы. Теги <TR></TR>определяет табличный ряд.
Теги<TH></TH>определяют заголовок таблицы. Теги<TD></TD>определяют ячейки таблицы. Теги <TBODY></ TBODY>определяют тело таблицы

<THEAD><TR><TH><B>A</B></TH><TH><B>B</B></TH><TH><B>A and B</B></TH><TH><B>A or B</B></TH><TH><B>not A</B></TH><TH><B>not B </B></TH></TR></THEAD>
<TBODY><TD>0</TD><TD>0</TD><TD>0</TD><TD>0</TD><TD>1</TD><TD>1</TD></ TR>
<TR><TD>0</TD><TD>1</TD><TD>0</TD><TD>1</TD><TD>1</TD><TD>0</TD></TR>
<TR><TD>1</TD><TD>0</TD><TD>0</TD><TD>1</TD><TD>0</TD><TD>1</TD></TR>
<TR><TD>1</TD><TD>1</TD><TD>1</TD><TD>1</TD><TD>0</TD><TD>0</TD></TR>
</ TBODY></TABLE></BODY></HTML>

3. Сохраните полученный файл как 2. html в вашей папке, откройте его с помощью браузера, в результате вы должны получить следующую таблицу. Количество тегов <TD></TD>должно соответствовать количеству ячеек с данными в таблице, а количество строк равно количеству тегов <TR></TR>. Содержимое ячеек заключено в тегах<TD></TD>.

ТАБЛИЦА № 1

ТАБЛИЦА ИСТИННОСТИ					
A	B	A and B	A or B	not A	not B
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

Задание 2. Создать таблицу по образцу

ТАБЛИЦА № 2

Густонаселенные города			
Город	Население	Страна	Координаты
Буэнос-Айрес	12 955 300	Аргентина	34° 37' ю.ш. 58° 22' з.д.
Бомбей	12 147 100	Индия	18° 58' с.ш. 72° 49' в.д.
Дели	10 009 200		28° 40' с.ш. 77° 13' в.д.
Сеул	11 153 200	Корея Южная	37° 34' с.ш. 126° 59' в.д.
Карачи	10 272 500	Пакистан	24° 52' с.ш. 67° 1' в.д.
Манила	10 133 200	Филипинны	14° 37' с.ш. 120° 58' в.д.
Сан-Паулу	10 057 700	Бразилия	23° 32' ю.ш. 46° 38' з.д.
Рио-де-Жанейро	6 029 300		22° 55' ю.ш. 43° 12' з.д.

Указания к выполнению

1. Откройте файл 2. html в Notepad++ , создайте таблицу № 2 из девяти строчек и четырех столбцов. Обратите внимание, что в 3-м столбце третья и четвертая строчки объединены в одну ячейку. Объединение ячеек производится с помощью атрибутов ROWSPAN и COLSPAN тегов <TD></TD>. В конкретном случае объединяются две строки, поэтому мы используем атрибут <TD ROWSPAN=2>, если нужно объединить три столбца, то можно использовать следующий атрибут тега <TD COLSPAN =3>. Ячейки, участвующие в объединении на последующих строчках нужно удалить.
2. Чтобы поменять цвет и шрифт текста в ячейках нужно теги <FONT FACE="TAHOMA" SIZE="5" COLOR="CYAN"></FONT>вставить в контейнер тегов<TD></TD>.

Самостоятельная работа. Создайте таблицу вашего расписания на неделю.

Составить отчет о выполнении.

Раздел 6. Информационное моделирование

Тема 6.1. Компьютерное информационное моделирование. Моделирование зависимостей между величинами

Информационное моделирование — это процесс создания абстрактных моделей, которые описывают организацию, структуру и потоки информации в системе или процессе. Это важная часть проектирования информационных систем, баз данных и других информационных архитектур. Информационные модели помогают понять, как данные и информация связаны, как они обрабатываются и как взаимодействуют внутри системы.

Существует несколько видов информационного моделирования:

1. Концептуальное моделирование:

- Этот уровень моделирования описывает основные понятия, сущности и их взаимосвязи в системе. Примеры таких моделей включают диаграммы сущность-связь (Entity-Relationship Diagrams, ERD).

2. Логическое моделирование:

- На этом уровне создаются модели данных, которые описывают структуру и организацию данных, но не зависят от конкретной реализации. Примеры: реляционные модели данных, диаграммы классов.

3. Физическое моделирование:

- Этот уровень описывает, как данные будут храниться и структурированы в конкретной базе данных или системе хранения. Примеры: диаграммы таблиц и связей для реляционных баз данных.

4. Процессное моделирование:

- Этот вид моделирования фокусируется на процессах и потоках информации. Примеры: диаграммы потоков данных (Data Flow Diagrams, DFD), бизнес-процессы.

5. Информационное моделирование в UML:

- Unified Modeling Language (UML) позволяет создавать разнообразные диаграммы, включая диаграммы классов, компонентов, последовательности и другие, которые включают информационные аспекты системы.

6. Информационное моделирование в CASE-средствах:

- Множество инструментов для CASE (Computer-Aided Software Engineering) предоставляют возможности для создания и визуализации информационных моделей, что упрощает процесс проектирования.

Информационное моделирование помогает разработчикам, аналитикам и другим заинтересованным сторонам лучше понять структуру данных и информационные потоки в системе, что способствует более эффективному и точному проектированию и разработке информационных ресурсов.

Компьютерное информационное моделирование

Компьютерное информационное моделирование — это использование компьютеров и специальных программных инструментов для создания и анализа информационных моделей, которые описывают различные аспекты систем, процессов или явлений. Это включает в себя создание компьютерных моделей, которые могут имитировать поведение реальных систем и позволяют проводить эксперименты, тестирование и прогнозирование.

Примеры компьютерного информационного моделирования включают:

1. Компьютерное моделирование систем:

- Моделирование физических систем (например, динамика движения тел) с помощью математических уравнений и численных методов.
- Моделирование социальных систем (например, экономические процессы, популяционные динамики).

2. Моделирование процессов:

- Моделирование бизнес-процессов для оптимизации рабочих процедур и выявления узких мест.
- Моделирование процессов производства и поставок для оптимизации логистики.

3. Компьютерное моделирование в науке и исследованиях:

- Моделирование в физике для изучения физических явлений, которые трудно или опасно изучать экспериментально.
- Моделирование в медицине для анализа биологических процессов и прогнозирования эффективности лечения.

4. Компьютерное моделирование в инженерии:

- Моделирование проектирования и тестирование изделий до их фактического создания.
- Моделирование и симуляция работы сложных систем, таких как авиационные системы.

5. Компьютерное моделирование в экологии:

- Моделирование экосистем для изучения влияния человеческой деятельности на окружающую среду.
- Прогнозирование изменений климата и их влияния на планету.

6. Виртуальная реальность (VR) и аугментированная реальность (AR):

- Создание виртуальных окружений и объектов для симуляции реальных ситуаций (VR) или добавления виртуальных элементов к реальной среде (AR).

Компьютерное информационное моделирование играет ключевую роль в научных исследованиях, проектировании, анализе данных, принятии решений и обучении. Оно позволяет создавать условия для тестирования различных сценариев и анализа результатов без необходимости фактического вмешательства в реальные системы или явления.

Моделирование зависимостей между величинами

Моделирование зависимостей между величинами — это процесс создания абстрактных моделей, которые описывают связи, взаимодействия и влияния между различными величинами или переменными. Эти модели помогают понять, как изменение одной величины может повлиять на другие и какие закономерности существуют в системе.

Существует несколько способов моделирования зависимостей:

1. Математическое Моделирование:

- Использует математические уравнения, формулы и функции для описания зависимостей между величинами. Примеры включают линейные и нелинейные уравнения, дифференциальные уравнения и другие.

2. Графическое Моделирование:

- Использует графику и диаграммы для визуализации зависимостей. Примеры включают графики, диаграммы рассеяния (scatter plots), графы и другие визуальные представления.

3. Статистическое Моделирование:

- Основано на анализе статистических данных для определения связей между величинами. Примеры включают линейную и множественную регрессию, анализ корреляции и др.

4. Моделирование на основе Правил и Логики:

- Описывает зависимости на основе логических условий и правил. Примеры включают логические цепи, булеву алгебру и правила продукции.

5. Моделирование Машинного Обучения:

- Использует алгоритмы машинного обучения для выявления и предсказания сложных зависимостей в данных. Примеры включают решающие деревья, нейронные сети, методы кластеризации и др.

6. Физическое Моделирование:

- Создает физические модели, которые демонстрируют зависимости в реальных физических процессах. Примеры включают макеты, эксперименты и физические симуляции.

7. Симуляционное Моделирование:

- Использует компьютерные симуляции для воссоздания сложных систем и процессов с учетом их взаимодействий и зависимостей.

Моделирование зависимостей позволяет анализировать и прогнозировать поведение системы при изменении входных данных или условий. Оно широко используется в науке, инженерии, экономике, биологии, социальных науках и других областях для более глубокого понимания сложных взаимосвязей.

Тема 6.2. Модели статического прогнозирования

Цель: изучить рабочее пространство приложения MS Excel, научиться применять различные параметры форматирования к данным, сортировать данные и проводить их фильтрацию по заданным условиям.

Теоретический материал:

Модели статистического прогнозирования — это математические и статистические методы, используемые для анализа и прогнозирования будущих значений переменных на основе исторических данных. Эти модели помогают предсказать тренды, шаблоны и зависимости в данных, что может быть полезным для принятия решений, планирования и управления ресурсами.

1. Экспоненциальное Сглаживание:

- Экспоненциальные модели используют экспоненциально взвешенные средние для прогнозирования будущих значений на основе предыдущих. Они подходят для данных без явных сезонных паттернов.

2. Модели Регрессии:

- Линейная и множественная регрессия используют статистические связи между зависимой переменной и независимыми переменными для прогнозирования. Это может быть полезно, когда есть явные зависимости.

3. ARIMA (Авторегрессионное Интегрированное Скользящее Среднее):

- ARIMA-модели применяются к временным рядам и учитывают как авторегрессию (зависимость текущих значений от предыдущих), так и скользящее среднее (зависимость от прошлых остатков).

4. GARCH (Обобщенная Авторегрессионная Скользящая Гетероскедастичность):

- GARCH-модели используются для прогнозирования волатильности в финансовых временных рядах.

5. Модели для Панельных Данных:

- Используются для анализа данных, собранных в разные моменты времени и на разных объектах, например, для анализа данных о компаниях из разных секторов экономики.

6. Модели Машинного Обучения:

- Нейронные сети, решающие деревья, случайные леса и другие методы машинного обучения также могут применяться для статистического прогнозирования.

7. Прогнозирование сезонности и трендов:

- Временные ряды часто содержат сезонные паттерны и общие тренды. Специализированные методы могут выделять и учитывать эти характеристики.

Выбор конкретной модели зависит от характера данных, целей прогнозирования и области применения. Прогнозы могут быть использованы для оптимизации запасов, финансового планирования, анализа рисков, предсказания тенденций и других целей.

Табличный процессор – комплекс программ, предназначенных для создания и обработки электронных таблиц.

Электронная таблица – компьютерный эквивалент обычной таблицы.

Электронная таблица (ЭТ) позволяет хранить в табличной форме большое количество исходных данных, результатов, а также связей (алгебраических или логических соотношений) между ними. При изменении исходных данных все результаты автоматически пересчитываются и заносятся в таблицу. Электронные таблицы не только автоматизируют расчеты, но и являются эффективным средством моделирования различных вариантов и ситуаций. Меняя значения исходных данных, можно следить за изменением получаемых результатов и из множества вариантов решения задачи выбрать наиболее приемлемый.

Табличный процессор MS Excel позволяет:

1.Решать математические задачи: выполнять разнообразные табличные вычисления, вычислять значения функций, строить графики и диаграммы и т.п.;

2.Осуществлять численное исследование (Что будет, если? Как сделать, чтобы?);

3.Проводить статистический анализ;

4.Реализовать функции базы данных – ввод, поиск, сортировку, фильтрацию (отбор) и анализ данных;

5.Устанавливать защиту на отдельные фрагменты таблицы, делать их невидимыми;

6.Наглядно представлять данные в виде диаграмм и графиков;

7.Вводить и редактировать тексты;

8.Осуществлять обмен данными с другими программами, например, вставлять текст, рисунокунки, таблицы, подготовленные в других приложениях;

9.Осуществлять многотабличные связи.

Практический материал:

Задание 1.

Запустите приложение MS Excel и изучите специальные элементы интерфейса.

Запустите электронный процессор MS Excel из главного меню Windows (Пуск – Программы – Microsoft Excel). В окне программы по умолчанию будет раскрыт документ, называемый Книга 1 и содержащий некоторое, установленное ранее, количество листов.

Изучите окно программы MS Excel и его специальные элементы (рисунок1).

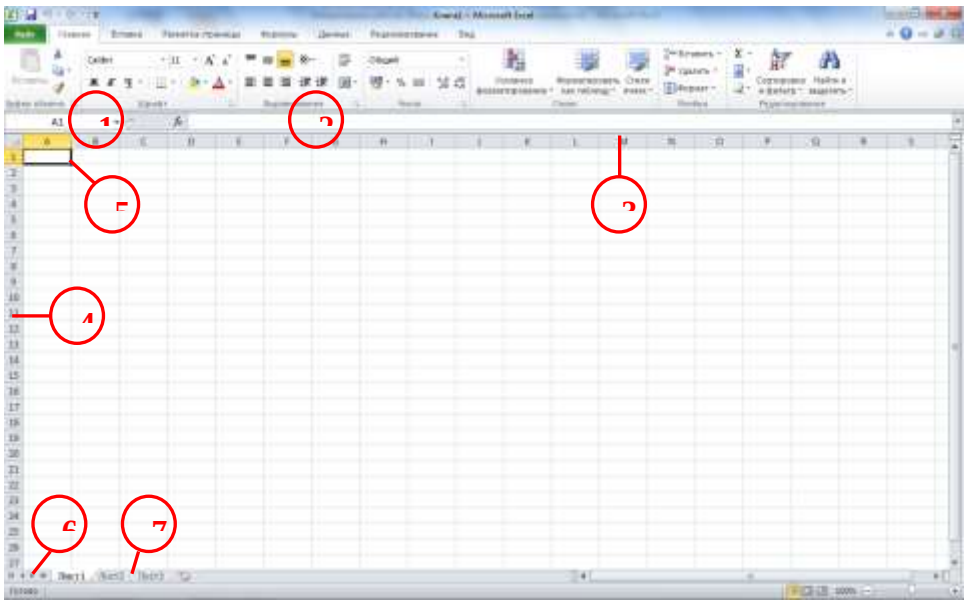


рисунок1 - Окно программы MS Excel

1. Поле имени
2. Строка формул
3. Кнопки заголовков столбцов
4. Кнопки заголовков строк
5. Текущая активная ячейка
6. Кнопки прокрутки листа
7. Ярлычки листов

Задание 2.

Создание рабочей книги с заданным количеством листов.

1. Для изменения количества листов, создаваемых по умолчанию в новой рабочей книге, выберите в меню Файл команду Параметры вкладку Общие.
2. Задайте Число листов – 10, Ок.
3. Выполните команду Файл – Создать – Новая книга.

В результате проделанных действий появится документ Книга 2 с десятью рабочими листами.

Задание 3.

Переименование листов.

1. Щелкните два раза на ярлычке Лист 1 (Или щелкните правой кнопкой мыши по ярлычку Лист 1 и выберите команду Переименовать).
2. В поле Имя листа наберите новое имя СПИСОК
3. Нажмите клавишу Enter. Рабочему Листу 1 присуноквоено имя СПИСОК.

Задание 4.

Вставка листа.

1. Щелкните правой кнопкой мыши на ярлычке Лист 2.
2. В контекстном меню выберите команду Вставить.
3. В открывшемся диалоговом окне Вставка, убедитесь, что значок Лист выделен и щелкните по кнопке ОК. Новый Лист 11 будет вставлен слева от текущего листа.

Задание 5.

Удаление листа.

1. Щелкните правой кнопкой мыши на ярлычке Лист 10.
2. В контекстном меню выберите команду Удалить.

Задание 6.

Выделение отдельных ячеек и блоков ячеек на рабочем листе.

1. Для выделения ячейки на рабочем листе достаточно щелкнуть по ней левой клавишей мыши.

2. Для одновременного выделения нескольких ячеек на рабочем листе эту же операцию необходимо произвести, удерживая клавишу Ctrl.

3. Для выделения блока ячеек необходимо щелкнуть левой клавишей мыши по первой ячейке, входящей в этот блок, а за тем, удерживая Shift, по последней ячейке из этого блока. Например: Выделите блок B4 : C10. (Примечание: Эту же операцию можно проделать, удерживая нажатой левую клавишу мыши).

4. Для выделения несмежных блоков удерживайте клавишу Ctrl. Выделите одновременно блоки: A4:C8; C10: E15; D1:F9.

5. Для выделения строки, щелкните на кнопке Заголовка строки, для выделения столбца, щелкните на кнопке Заголовка столбца.

В любую ячейку можно вводить текст, числа или формулы. Для этого достаточно активизировать нужную ячейку и набрать с клавиатуры необходимую информацию. Затем нажать Enter или осуществить щелчок мыши по любой другой ячейке таблицы.

Если вводимая информация не соответствует размеру ячейки можно:

1. Изменить ее размеры, для чего установить курсор мыши на границу между кнопками заголовка столбцов и, поймав двунаправленную стрелку, увеличить или уменьшить размеры ячейки.

2. На вкладке Главная в разделе Ячейки выбрать Формат и задать необходимые параметры для ширины и высоты ячейки (рисунок 2).

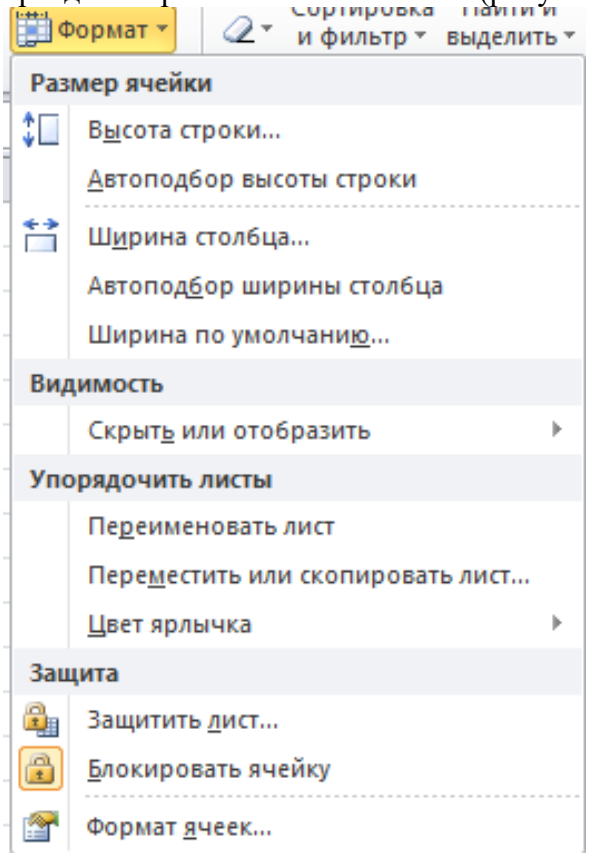


рисунок 2. Параметры форматирования ячеек

3. Задайте другие параметры форматирования: расположить текст в ячейке в две строки. Для чего выберите опцию Формат ячеек вкладка Выравнивание, включите команду Переносить по словам.

4. Расположить текст по вертикали, для чего в диалоговом окне Формат ячеек вкладка Выравнивание выбрать соответствующую ориентацию текста.

Объединение ячеек

Объединенная ячейка строится из ячеек смежных строк и столбцов, обводится на рабочем листе единой границей, и имеет адрес, совпадающей с адресом ее левой верхней ячейки. Для создания объединенной ячейки необходимо выделить ячейки, которые войдут в ее состав, а затем выполнить одно из следующих действий:

- Установите флажок Объединение ячеек на вкладке Выравнивание.
- Нажать кнопку Объединить и поместить в центре на панели Главная.

Задание 7.

На листе СПИСОК создайте таблицу Список сотрудников по образцу (рисунок 3):

	A	B	C	D	E	F	G
1	**№ п/п**	**Фамилия**	**Имя**	**Отчество**	**Пол**	**Год рождения**	**Отдел**
2	1	Акимова	Вера	Федоровна	ж	1973	2
3	2	Анисимов	Андрей	Александрович	м	1956	1
4	3	Балаев	Игорь	Сергеевич	м	1954	3
5	4	Бореев	Виктор	Александрович	м	1965	3
6	5	Боркут	Ирина	Александровна	ж	1974	2
7	6	Воронова	Надежда	Ивановна	ж	1984	2
8	7	Ворошилов	Петр	Ильич	м	1990	3
9	8	Иванов	Илья	Алексеевич	м	1983	1
10	9	Попов	Виктор	Сергеевич	м	1973	2
11	10	Щербакова	Елена	Ивановна	ж	1962	1

рисунок 3. Образец таблицы для заполнения

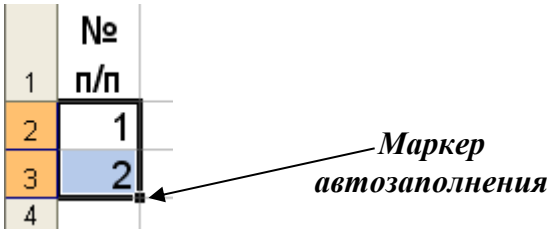
1. Используя, полученные знания по вводу текстовой информации в ячейки, заполните заголовки столбцов таблицы. Оформите заголовки столбцов таблицы, используя вкладки Выравнивание и Шрифт меню Формат ячеек панель Главная. Для этого:
   - Выделите все заполненные ячейки;
   - Во вкладке Выравнивание установите: Выравнивание по горизонтали и Выравнивание по вертикали - По центру; Установите флажок Переносить по словам.
   - Во вкладке Шрифт установите начертание Полужирный курсив.
2. Внесите в таблицу данные не менее чем по 30 сотрудникам.
3. Создайте вокруг данных таблицы рамку, используя кнопку Границы – Все границы на панели Главная или вкладку Граница в меню Формат ячейки.

В Excel предусмотрено средство для ввода часто используемых последовательностей данных (натуральных чисел, дней недели, месяцев и т.д.).

Задание 8.

На Листе 11 введите последовательность, состоящую из натуральных чисел (от 1 до 12) и месяцев.

1. В любые две рядом стоящие ячейки введите два числа. Например: В ячейку А1 – 1; в ячейку А2 – 2;
2. Выделите (замаркируйте) заполненные ячейки с помощью мыши;
3. Подведите курсор к маркеру автозаполнения, находящемся в правом нижнем углу выделенных ячеек, курсор приобретет вид тонкого крестика.
4. Нажмите левую клавишу мыши и, удерживая ее нажатой, протяните курсор вниз, пока не появится цифра 12.



5. В ячейку B1 введите название первого месяца – январь;

6. Подведите курсор к маркеру заполнения и протяните его вниз до ячейки B12. В результате должна получится последовательность, состоящая из месяцев года.

Задание 9.

Создание собственной последовательности данных.

Вы можете создать свою последовательность данных, которую вы предполагаете часто использовать. Делается это одним из двух следующих способов:

I способ – Импорт последовательности:

1. Выделите (замаркируйте) блок ячеек, содержащий фамилии сотрудников на листе СПИСОК;

2. Выполните команду меню Файл – Параметры – Дополнительно – Изменить списки;

3. В поле окна Импорт списка из ячеек будет находиться адрес выделенного диапазона ячеек.

4. Щелкните по кнопке Импорт и убедитесь, что на вкладке Списки появилась новая импортированная последовательность.

5. Закройте диалог Параметры щелчком по кнопке ОК.

II способ – Ручной ввод последовательности:

Введем последовательность учебных предметов, которую будем использовать в дальнейшей работе:

1. Откройте диалоговое окно Списки (меню Файл – Параметры – Дополнительно – Изменить списки);

2. Щелкните мышью в правом поле Элементы списка;

3. Наберите Информатика, Психология, История, Математика, Иностранный язык, отделяя каждый элемент списка от предыдущего запятой или нажимая после каждого элемента Enter.

4. Щелкните на кнопке Добавить – и новая последовательность окажется в левом поле списка.

5. Попробуйте воспроизвести вновь созданные списки на Листе 11.

Следует различать сортировку, производимую по одному столбцу, и сортировку по нескольким столбцам таблицы. Команды сортировки доступны с панелей Главная и Данные.

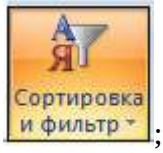
Задание 10.

Проведите сортировку данных таблицы.

1. Проведите сортировку по полю Отдел по возрастанию:

для чего:

- установите курсор в пределах поля Отдел;
- активизируйте панель Главная (если она не является активной по умолчанию);
- откройте список Сортировки и фильтрации, нажав соответствующую кнопку



;

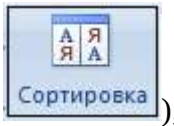
● для сортировки по одному полю используются кнопки Сортировать от минимального к максимальному и Сортировать от максимального к минимального . Нажмите кнопку Сортировать от минимального к максимальному.

Внимание! Сортировку также можно провести при помощи команд сортировки панели Данные.

2. Проведите одновременно сортировку по полю Отдел, затем по полю Год рождения. для чего:

● установите курсор в пределы сортируемой таблицы;

● откройте список Сортировки и фильтрации и выберите пункт Настраиваемая сортировка

(или на панели Данные щелкните по кнопке Сортировка 

);

● в диалоговом окне Сортировка установите Сортировать по - Отдел, Сортировка – Значения, Порядок – По возрастанию;

● щелкните по кнопке Добавить уровень, установите Затем по – Год рождения, Сортировка – Значения, Порядок – По возрастанию, нажмите ОК (рисунок 4).

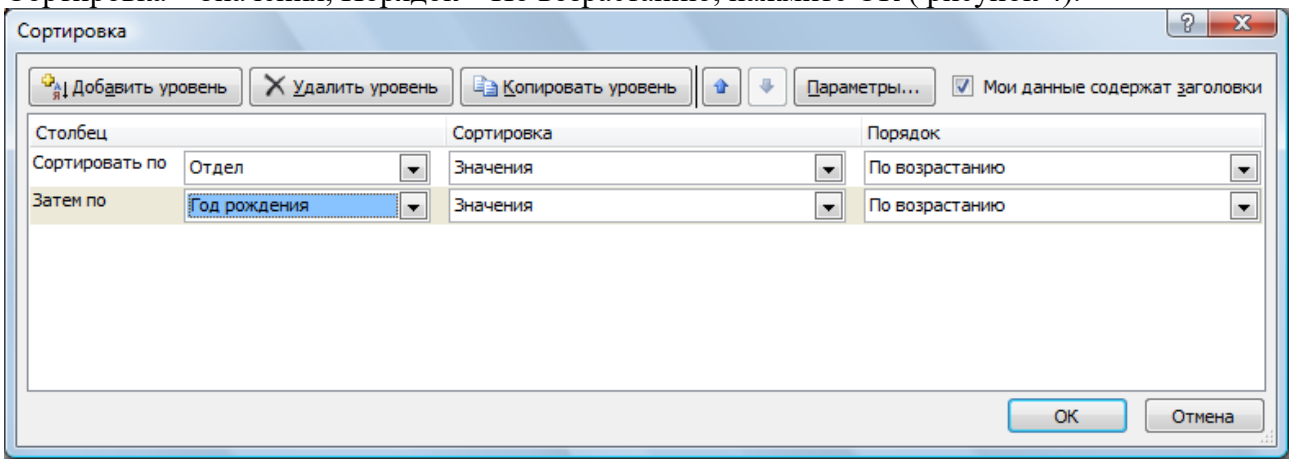


рисунок 4. Настройки диалогового окна Сортировка

Внимание! При желании можно добавить и другие уровни сортировки.

Отбор записей или, другими словами, фильтрация представляет собой выделение из списка определенных записей, которые удовлетворяют заданным критериям.

Задание 11.

Фильтрация данных с помощью автофильтра.

1. Поместите курсор в пределы Таблицы 1. На вкладке Данные выберите Фильтр

, справа от названия каждого столбца появится кнопка автофильтра .

2. Щелкните мышью по кнопке автофильтра, находящейся в столбце Фамилия. Раскроется список автофильтра.

3. Выберите пункт Текстовые фильтры – начинается с. В результате откроется диалоговое окно Пользовательского автофильтра (рисунок 5). Введите букву, с которой должны начинаться искомые фамилии, например А. Нажмите ОК.

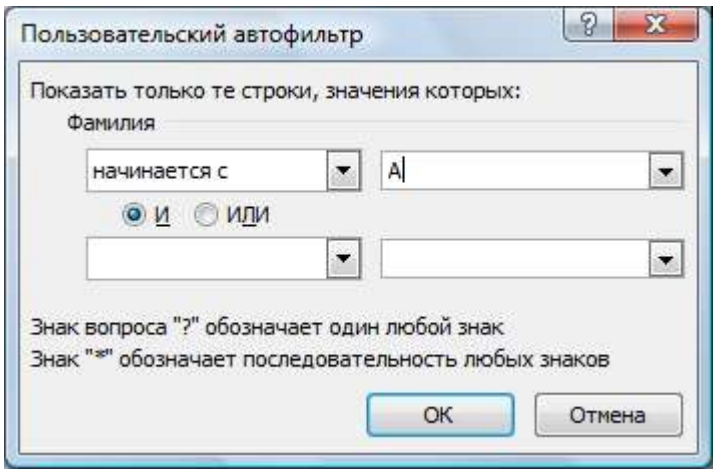


рисунок 5. Окно пользовательского автофильтра

4. Восстановите все записи, выбрав в списке автофильтра пункт Снять фильтр с Фамилия.

5. Попробуйте задать два условия для фильтрации. Например, выберите всех сотрудников фамилии которых начинаются на буквы А и В. Для чего в окне пользовательского автофильтра (Текстовые фильтры – Настраиваемый фильтр) в первой строке установите начинается с А, затем щелкните по переключателю ИЛИ и во второй строке установите начинается с В.

6. Восстановите все записи

7. Самостоятельно, продумав критерии фильтрации, отфильтруйте записи таблицы по другим столбцам.

8. Для полной отмены процедуры фильтрации на панели Данные нажмите кнопку Очистить. Отключите кнопки автофильтра.

Задание 12.

Используя расширенный фильтр, отберите сотрудников мужского пола и родившихся в 1983 году

1. Создайте таблицу условий, в которую скопируйте название столбцов Пол и Год рождения;

Внесите в эту таблицу критерии для отбора данных: пол – м; год рождения – 1983 (рисунок 6);

Примечание: если записи в таблице условий расположены в одной строке, то это условие И, а если на разных строках, то условие Или.

	A	B	C	D	E	F	G	H	I	J
1	*№ п/п*	*Фамилия*	*Имя*	*Отчество*	*Пол*	*Год рождения*	*Отдел*		*Пол*	*Год рождения*
2	1	Акимова	Вера	Федоровна	ж	1978	2		м	1983
3	2	Анисимов	Андрей	Александрович	м	1956	1			
4	3	Балаев	Игорь	Сергеевич	м	1954	3			
5	4	Бореев	Виктор	Александрович	м	1965	3			
6	5	Боркут	Ирина	Александровна	ж	1974	2			
7	6	Воронова	Надежда	Ивановна	ж	1984	2			
8	7	Ворошилов	Петр	Ильич	м	1969	3			
9	8	Иванов	Илья	Алексеевич	м	1983	1			
10	9	Попов	Виктор	Сергеевич	м	1973	2			
11	10	Щербакова	Елена	Ивановна	ж	1962	1			
12										
13				*Исходная таблица*					*Таблица условий*	
14										

рисунок 6. Работа с расширенным фильтром

3. Установите курсор в пределы исходной таблицы. На панели Данные щелкните по кнопке Дополнительно. Раскроется диалоговое окно Расширенного фильтра, в котором нужно указать: Исходный диапазон (адрес Таблицы из которой будет производится выборка данных – таблица Список сотрудников); Диапазон условий (адрес созданной таблицы условий); Диапазон куда поместить результат выборки (Диапазоны указываются путем выделения мышью) (рисунок 7).

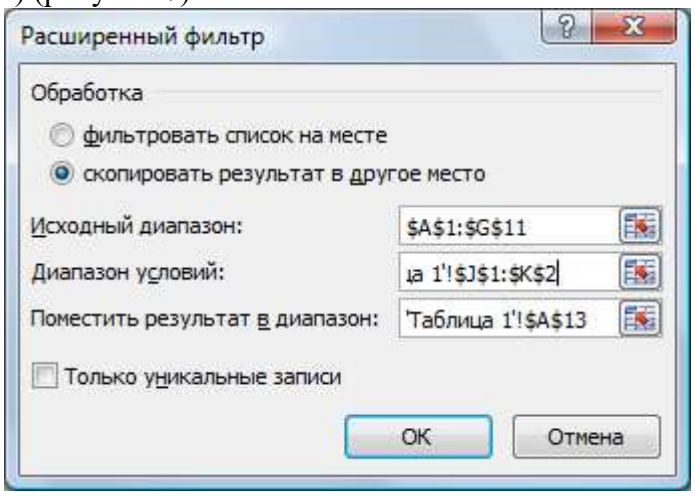


рисунок 7. Диалоговое окно Расширенного фильтра

4. После указания всех необходимых диапазонов нажмите кнопку ОК.

5. Используя расширенный фильтр, произведите фильтрацию данных таблицы по другим критериям, придуманным вами.

Задание 13.

Подведите промежуточные итоги в таблице.

1. Добавьте к таблице еще один столбец Зарплата и произвольно заполните его данными.

2. Подсчитайте сумму зарплаты отдельно для сотрудников из 1, 2, 3 отделов. Для чего:

- Произведите сортировку по возрастанию данных таблицы по столбцу Отдел: установите курсор в предел столбца Отдел и нажмите кнопку Сортировка по возрастанию на панели Данные;
- Выберите на панели Данные команду Промежуточные итоги ;
- В появившемся окне Промежуточные итоги выберите: Отдел в поле При каждом изменении в; Операция – Сумма; В поле Добавить итоги по: поставьте птичку напротив Зарплата (рисунок 8);

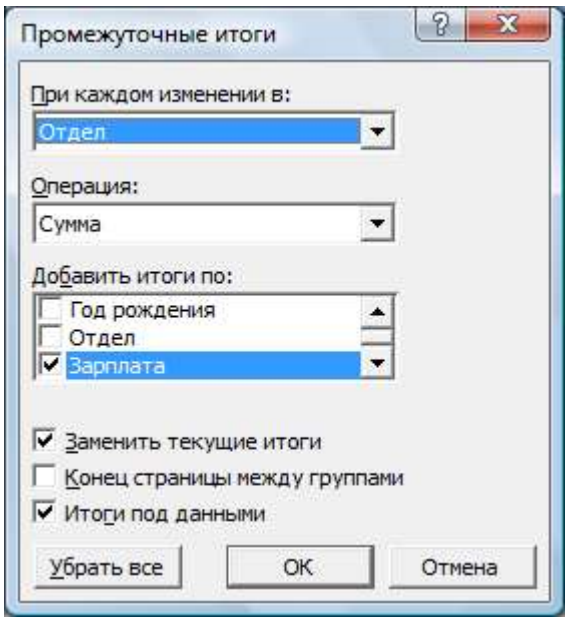


рисунок 8. Диалоговое окно Промежуточные итоги

- Нажмите ОК, обратите внимание на произведенные программой подсчеты (рисунок 9).

	A	B	C	D	E	F	G	H
1	№ п/п	Фамилия	Имя	Отчество	Пол	Год рождения	Отдел	Зарплата
2	2	Анисимов	Андрей	Александрович	м	1956	1	7500
3	8	Иванов	Илья	Алексеевич	м	1983	1	6800
4	10	Щербакова	Елена	Ивановна	ж	1962	1	12300
5							1 Итог	26600
6	1	Акимова	Вера	Федоровна	ж	1978	2	8300
7	5	Боркут	Ирина	Александровна	ж	1974	2	10500
8	6	Воронова	Надежда	Ивановна	ж	1984	2	6000
9	9	Попов	Виктор	Сергеевич	м	1973	2	8350
10							2 Итог	33150
11	3	Балаев	Игорь	Сергеевич	м	1954	3	4800
12	4	Бореев	Виктор	Александрович	м	1965	3	12700
13	7	Ворошилов	Петр	Ильич	м	1969	3	5800
14							3 Итог	23300
15							Общий итог	83050

рисунок 9. Результат подведения промежуточных итогов

Задание 14.

Подготовить ответы на вопросы

7. Назовите элементы окна приложения Excel.
8. Расскажите о возможностях форматирования данных в Excel.
9. Создайте любую последовательность данных одним из изученных способов и воспроизведите ее на листе Excel.
10. Проведите фильтрацию данных с помощью автофильтра и расширенного фильтра, используя для фильтрации собственные критерии (критерии придуманные вами).

Подведите промежуточные итоги по столбцу зарплата для сотрудников мужского и женского пола.

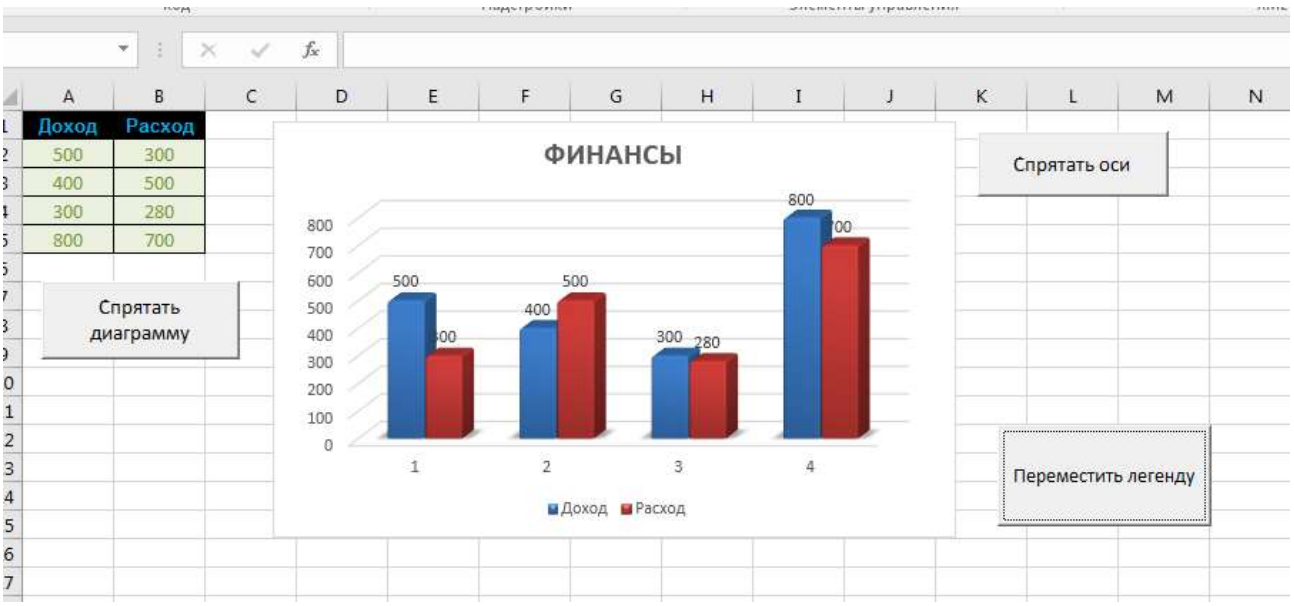
```
Option Explicit

Private Sub tglChart_Click()
  With ActiveSheet.ChartObjects(1)
    If .Visible Then
      .Visible = False
      tglChart.Caption = "Показать диаграмму"
```

```
    Else
      .Visible = True
      tglChart.Caption = "Спрятать диаграмму"
    End If
  End With
End Sub

Private Sub tglAxis_Click()
  With ActiveSheet.ChartObjects(1).Chart
    If .HasAxis(xlCategory) And .HasAxis(xlValue) Then
      .HasAxis(xlCategory) = False
      .HasAxis(xlValue) = False
      tglAxis.Caption = "Показать оси"
    Else
      .HasAxis(xlCategory) = True
      .HasAxis(xlValue) = True
      tglAxis.Caption = "Спрятать оси"
    End If
  End With
End Sub

Private Sub Кнопка1_Click()
  With ActiveSheet.ChartObjects(1).Chart
    .HasLegend = True ' Проверка того, что легенда отображается
    .Legend.Position = xlBottom ' Разместить легенду внизу
  End With
End Sub
```



Тема 6.3. Модели корреляционных зависимостей

Моделирование корреляционных зависимостей включает в себя анализ и визуализацию взаимосвязей между двумя или более переменными. Корреляционные зависимости могут помочь выявить, насколько и какие переменные связаны между собой, что может быть полезным для анализа данных, прогнозирования и принятия решений. Для моделирования корреляционных зависимостей используются различные методы и инструменты:

1. Коэффициент Корреляции:

- Коэффициент корреляции измеряет степень линейной зависимости между двумя переменными. Наиболее известным является коэффициент корреляции Пирсона.

2. Диаграммы Рассеяния:

- Диаграммы рассеяния визуализируют точки данных на плоскости и помогают оценить, как одна переменная меняется относительно другой. Форма диаграммы может указывать на характер зависимости.

3. Матрица Корреляции:

- Матрица корреляции предоставляет коэффициенты корреляции между всеми парами переменных в наборе данных. Она позволяет быстро оценить связи между переменными.

4. Регрессионный Анализ:

- Регрессионный анализ может выявить не только наличие корреляции, но и предсказывать значения одной переменной на основе другой.

5. Коррелограммы:

- Коррелограммы представляют собой графическое представление матрицы корреляции, используя цвета или размеры ячеек для отображения степени корреляции.

6. Нонпараметрические Методы:

- Когда данные не соответствуют нормальному распределению, используются непараметрические методы для оценки корреляций, такие как коэффициент Спирмена и коэффициент Кендалла.

7. Методы Машинного Обучения:

- Некоторые методы машинного обучения, такие как случайные леса, могут автоматически выявлять и оценивать сложные нелинейные зависимости между переменными.

Моделирование корреляционных зависимостей позволяет понять, какие переменные взаимосвязаны и как одна переменная может влиять на другую. Это важно для принятия решений, стратегического планирования и разработки моделей прогнозирования.

Практический материал

```
Option Explicit

Sub Экспорт_в_Word()
  Dim WordApp As Object
  ' Копирование данных_в_Excel
  ActiveSheet.Range("A1:B15").Copy
  Set WordApp = CreateObject("Word.Application")
  With WordApp
    .Visible = True      ' Запуск приложения Word
    .Documents.Add       ' Добавление нового документа
    .Selection.Paste     ' Вставка скопированной области
  End With
End Sub
```

Тема 6.4. Модели оптимального планирования

Модели оптимального планирования — это математические и статистические методы, используемые для определения наилучших решений в условиях ограничений и целей. Они позволяют оптимизировать ресурсы, принимать решения и разрабатывать стратегии так, чтобы достичь наилучших результатов.

Примеры моделей оптимального планирования:

1. Линейное программирование (ЛП):

- Линейное программирование используется для оптимизации ресурсов в условиях линейных ограничений. Примеры включают оптимизацию затрат производства или максимизацию прибыли.

2. Целочисленное программирование (ЦП):

- Целочисленное программирование расширяет концепцию линейного программирования, позволяя решать задачи с целочисленными переменными. Это может быть полезно при планировании распределения ресурсов между различными вариантами.

3. Динамическое программирование:

- Динамическое программирование используется для оптимизации последовательных решений во времени или пространстве. Примеры включают оптимальное управление запасами, управление производственными процессами и принятие решений в условиях неопределенности.

4. Методы Марковских процессов:

- Марковские процессы используются для моделирования принятия решений в условиях стохастических изменений. Это может быть полезно при планировании ресурсов в изменяющейся среде.

5. Оптимизация с использованием алгоритмов эволюционной оптимизации:

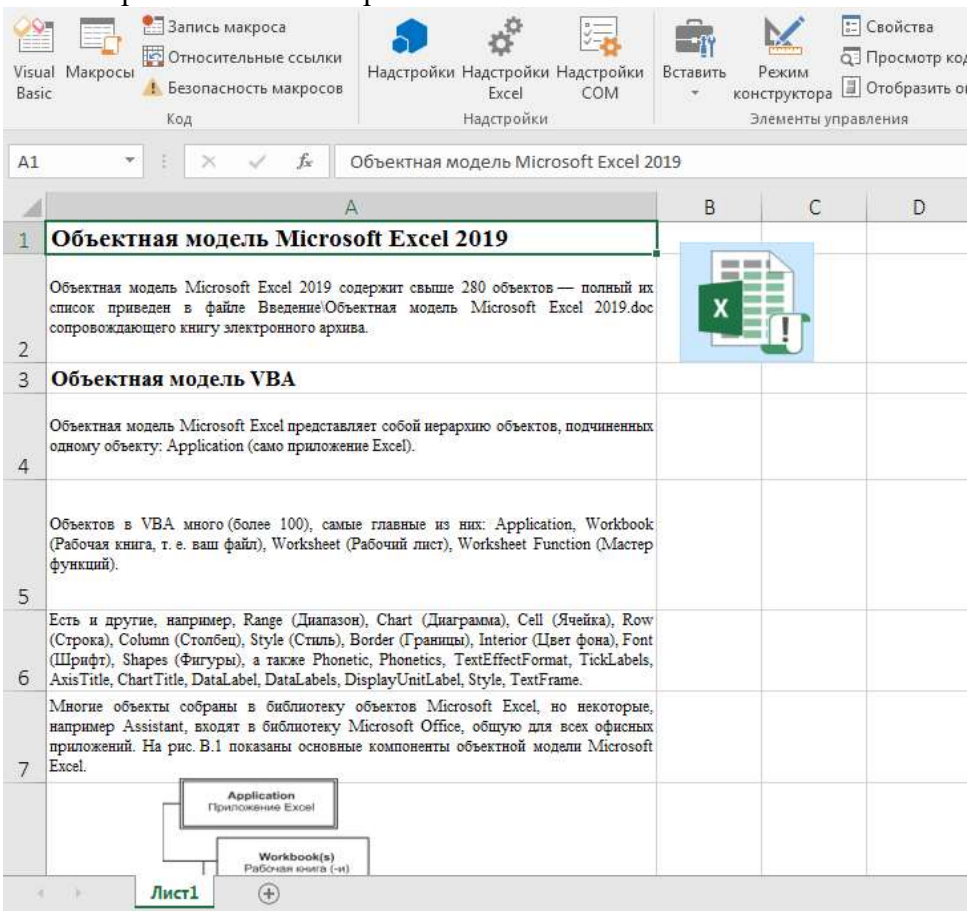
- Эволюционные алгоритмы, такие как генетические алгоритмы или метод частиц, используют принципы эволюции для поиска оптимальных решений в сложных и динамичных задачах.

6. Методы Машинного Обучения:

- Нейронные сети, методы регрессии и классификации, алгоритмы кластеризации и другие методы машинного обучения могут быть применены для оптимизации различных бизнес-процессов и стратегий.

Модели оптимального планирования помогают рационально использовать ресурсы, минимизировать затраты, максимизировать выгоды и достигать целей в условиях ограничений. Они широко применяются в экономике, бизнесе, инженерии, производстве, логистике и других областях, где требуется эффективное управление ресурсами и принятие обоснованных решений.

Практический материал



```
Option Explicit

Sub Импорт_из_Word()
  Dim WordApp As Object
  Dim str1 As String
  Const strFile As String = "Пример1.docx"
  str1 = ThisWorkbook.Path
  ' Операции с Word
  Set WordApp = CreateObject("Word.Application")
  With WordApp
    .ChangeFileOpenDirectory str1
    .Documents.Open strFile
    With .Selection
      .WholeStory
      .Copy
    End With
  End With
  ' Операции с Excel
  Workbooks.Add
  ActiveSheet.Paste
End Sub
```

Раздел 7. Социальная информатика

Тема 7.1. Информационные ресурсы

Цель: научиться пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью, овладеть методами работы с программным обеспечением.

Теоретический материал:

Понятие «информационного ресурса общества» (ИРО) является одним из ключевых понятий социальной информатики.

Информационные ресурсы–это знания, подготовленные для целесообразного социального использования.

ИРО в узком смысле слова – это знания, уже готовые для целесообразного социального использования, то есть отчужденные от носителей и материализованные знания.

ИРО в широком смысле слова включают в себя все отчужденные от носителей и включенные в информационный обмен знания, существующие как в устной, так и в материализованной форме.

Понятие ресурс определяется в словаре русского языка С.И. Ожегова как запас, источник чего-нибудь.

Что же касается информационных ресурсов, то это понятие является сравнительно новым. Оно еще только начинает входить в жизнь современного общества, хотя в последние годы становится все более употребительным не только в научной литературе, но и в общественно-политической деятельности.

Причиной этого, безусловно, является глобальная информатизация общества, в котором все больше начинает осознаваться особо важная роль информации и научных знаний.

Для классификации информационных ресурсов используют следующие их наиболее важные параметры:

тематика хранящейся в них информации;

форма собственности – государственная (федеральная, субъекта федерации, муниципальная), общественных организаций, акционерная, частная;

доступность информации – открытая, закрытая, конфиденциальная;

принадлежность к определенной информационной системе – библиотечной, -архивной, научно-технической;

источник информации – официальная информация, публикации в СМИ, статистическая отчетность, результаты социологических исследований;

назначение и характер использования информации – массовое региональное, ведомственное;

форма представления информации – текстовая, цифровая, графическая, мультимедийная;

вид носителя информации – бумажный, электронный.

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 определенной предметной области.

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).

К образовательным электронным ресурсам относят:

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.

Практический материал:

Задание 1.

Приведите примеры:

1) достоверной, но необъективной информации;
2) объективной, но
3) недостоверной информации;
4) полной, достоверной, но бесполезной информации;
5) неактуальной информации;
6) актуальной, но непонятной информации.

Задание 2.

Пользуясь любыми поисковыми системами, дополните таблицу найденными Интернет - ресурсами в соответствии с Вашими профессиональными интересами.

Тип Интернет-ресурса	Вид Интернет-ресурса		Примеры Интернет-ресурсов
Учебные материалы	1	Учебник, учебное пособие	
	2	Электронный учебный курс	
	3	Текст лекций	
	4	Лабораторный практикум	
	5	Задачник	
	6	Тест, контрольные вопросы	
Справочные материалы	7	Энциклопедия	
	8	Словарь	
	9	Справочник	
	10	База данных	
	11	Геоинформационная/картографическая система	

Задание 3.

С помощью Универсального справочника-энциклопедии найдите ответы на следующие вопросы:

1) Что такое WWW?
2) Кто разработчик первого компьютера?
3) Когда отмечают Всемирный день информации?
4) Кто такой К.Э.Циалковский? Годы его жизни. Место работы.
5) Дата первых Олимпийских игр.
6) Микенская культура
7) Когда была Троянская война?

Задание 4.

Запишите ответы на вопросы:

1) Что такое информационные ресурсы?
2) Что такое образовательные информационные ресурсы?
3) Что относится к образовательным информационным ресурсам?

Задание 5.

Работа с тестовой программой MYTEST

Тема 7.2. Информационное общество

Цель: Изучить влияния информационных технологий на социальные структуры, общественные процессы и взаимодействие между людьми. Провести анализ воздействия информационных технологий на поведение людей, образ жизни, культуру и коммуникацию.

Теоретический материал

Информационное общество - это концепция, которая описывает современное общество, в котором информация и информационные технологии занимают центральное место в различных аспектах жизни. В информационном обществе информация становится основным ресурсом, оказывающим влияние на экономику, культуру, образование, политику, коммуникацию и другие сферы деятельности.

Основные черты информационного общества:

1. **Зависимость от информации:** Информация становится ключевым ресурсом, а способность обрабатывать, анализировать и передавать информацию становится критической компетенцией.

2. **Информационные технологии:** Развитие и широкое использование информационных технологий, таких как компьютеры, интернет, мобильные устройства, приводит к изменению способов взаимодействия и коммуникации.

3. **Глобальная связанность:** Информационные технологии позволяют людям взаимодействовать, обмениваться информацией и сотрудничать на мировом уровне, что приводит к глобальной связанности.

4. **Цифровые навыки:** Важность компьютерных навыков и грамотности в цифровой среде становится все более значимой для успешной адаптации в информационном обществе.

5. **Электронное правительство:** Государственные и административные процессы становятся более эффективными и открытыми благодаря цифровизации государственных услуг и коммуникации с гражданами.

6. **Цифровая экономика:** Информационные технологии становятся движущей силой экономического роста и инноваций, приводя к развитию новых бизнес-моделей и технологических стартапов.

7. **Доступ к знаниям:** Интернет и онлайн-ресурсы обеспечивают беспрецедентный доступ к знаниям и образованию, что приводит к образованности и самосовершенствованию.

8. **Изменение способов работы:** Гибкость в работе, удаленное сотрудничество, фриланс и другие новые формы работы становятся более распространенными.

Информационное общество привносит большие возможности и вызовы. С одной стороны, оно открывает новые пути для развития, инноваций и сотрудничества, с другой стороны, оно может вызвать вопросы касательно приватности, безопасности данных, цифрового неравенства и других социальных аспектов.

Тема 7.3. Правовое регулирование в информационной сфере

Цель: Научиться определять лицензионные и свободно распространяемые программные продукты. Формировать навыки по обновлению программного обеспечения с использованием сети Интернет.

Теоретический материал:

Правовое регулирование в информационной сфере охватывает набор законов, норм и правил, которые регулируют использование, передачу, хранение и защиту информации, а также деятельность в области информационных технологий. Это важный аспект в современном обществе, где информация играет ключевую роль во многих аспектах жизни, включая коммуникацию, бизнес, образование, медиа и многие другие.

Правовое регулирование в информационной сфере включает следующие аспекты:

1. **Авторские права:** Законы об авторских правах регулируют права и обязанности авторов произведений, таких как литературные, музыкальные и художественные произведения. Они также определяют права пользователей на использование материалов и защиту интеллектуальной собственности.

2. **Защита персональных данных:** Законы о защите персональных данных определяют, как организации могут собирать, использовать и хранить личные данные граждан. Они обеспечивают конфиденциальность информации и права на ее удаление.

3. **Кибербезопасность:** Законы о кибербезопасности регулируют защиту информационных систем от киберугроз, включая хакерские атаки, вирусы и другие вредоносные программы. Они могут включать требования по обязательной защите данных и обязанности уведомления о нарушениях.

4. **Электронная коммерция:** Законы об электронной коммерции устанавливают правила для совершения онлайн-торговли, включая правила о защите прав потребителей, электронной подписи и договорах.

5. **Свобода слова и информации:** Законы о свободе слова и информации определяют права и обязанности медиа, журналистов и граждан на доступ к информации и свободное выражение мнений.

6. **Электронная подпись и документы:** Законы о электронной подписи регулируют использование электронных подписей для заключения сделок и подписания документов в электронной форме.

7. **Законодательство о киберпреступлениях:** Законы о киберпреступлениях определяют правила и наказания за незаконное вторжение в компьютерные системы, кражу данных, мошенничество и другие преступления в сфере информационных технологий.

Правовое регулирование в информационной сфере имеет важное значение для обеспечения защиты прав и свобод граждан, обеспечения безопасности данных, стимулирования инноваций и развития цифровой экономики. Оно также помогает балансировать интересы разных сторон, таких как авторы, пользователи, организации и государство, и создавать правила для эффективного функционирования цифрового общества.

Программы по их правовому статусу можно разделить на три большие группы: лицензионные, условно бесплатные и свободно - распространяемые.

1. Лицензионные программы. В соответствии с лицензионным соглашением разработчики программы гарантируют её нормальное функционирование в определенной операционной системе и несут за это ответственность.

Лицензионные программы разработчики обычно продают в коробочные дистрибутивы. В коробочке находятся CD-диски, с которых производится установка программы на компьютеры пользователей, и руководство пользователей по работе с программой.

Довольно часто разработчики предоставляют существенные скидки при покупке лицензий на использовании программы на большом количестве компьютеров или учебных заведениях.

2. Условно бесплатные программы. Некоторые фирмы разработчики программного обеспечения предлагают пользователям условно бесплатные программы в целях рекламы и продвижения на рынок. Пользователю предоставляется версия программы с определённым сроком действия (после истечения указанного срока действия программы прекращает работать, если за неё не была произведена оплата) или версия программы с ограниченными функциональными возможностями (в случае оплаты пользователю сообщается код, включающий все функции программы).

3. Свободно распространяемые программы. Многие производители программного обеспечения и компьютерного оборудования заинтересованы в широком бесплатном распространении программного обеспечения. К таким программным средствам можно отнести:

− Новые недоработанные (бета) версии программных продуктов (это позволяет провести их широкое тестирование).

− Программные продукты, являющиеся частью принципиально новых технологий (это позволяет завоевать рынок).

− Дополнения к ранее выпущенным программам, исправляющие найденные ошибки или расширяющие возможности.

− Драйверы к новым или улучшенные драйверы к уже существующим устройствам.

− Но какое бы программное обеспечение вы не выбрали, существуют общие требования ко всем группам программного обеспечения:

− Лицензионная чистота (применение программного обеспечения допустимо только в рамках лицензионного соглашения).

− Возможность консультации и других форм сопровождения.

− Соответствие характеристикам, комплектации, классу и типу компьютеров, а также архитектуре применяемой вычислительной техники.

− Надежность и работоспособность в любом из предусмотренных режимов работы, как минимум, в русскоязычной среде.

− Наличие интерфейса, поддерживающего работу с использованием русского языка. Для системного и инструментального программного обеспечения допустимо наличие интерфейса на английском языке.

− Наличие документации, необходимой для практического применения и освоения программного обеспечения, на русском языке.

− Возможность использования шрифтов, поддерживающих работу с кириллицей.

− Наличие спецификации, оговаривающей все требования к аппаратным и программным средствам, необходимым для функционирования данного программного обеспечения.

4. Преимущества лицензионного и недостатки нелицензионного программного обеспечения.

Лицензионное программное обеспечение имеет ряд преимуществ.

− Техническая поддержка производителя программного обеспечения.

− При эксплуатации приобретенного лицензионного программного обеспечения у пользователей могут возникнуть различные вопросы. Владельцы лицензионных программ имеют право воспользоваться технической поддержкой производителя программного обеспечения, что в большинстве случаев позволяет разрешить возникшие проблемы.

− Обновление программ.

− Легальные пользователи оперативно и бесплатно получают все вышедшие обновления.

− Законность и престиж.

− Производители программного обеспечения регулярно выпускают пакеты обновлений лицензионных программ (patch, service-pack). Их своевременная установка - одно из основных средств защиты персонального компьютера (особенно это касается антивирусных программ).

− Покупая нелицензионное программное обеспечение, вы нарушаете закон, так как приобретаете «ворованные» программы. Вы подвергаете себя и свой бизнес риску юридических санкций со стороны правообладателей. У организаций, использующих нелегальное программное обеспечение, возникают проблемы при проверках лицензионной чистоты программного обеспечения, которые периодически проводят правоохранительные органы. За нарушение авторских прав в ряде случаев предусмотрена не только административная, но и уголовная ответственность. Нарушение законодательства, защищающего авторское право, может негативно отразиться на репутации компании. Нелицензионные копии программного обеспечения могут стать причиной несовместимости программ, которые в обычных условиях хорошо взаимодействуют друг с другом.

− В ногу с техническим прогрессом

− Управление программным обеспечением поможет определить потребности компании в программном обеспечении, избежать использования устаревших программ и будет способствовать правильному выбору технологии, которая позволит компании достичь поставленных целей и преуспеть в конкурентной борьбе.

− Профессиональные предпродажные консультации

− Преимущества приобретения лицензионного программного обеспечения пользователи ощущают уже при его покупке. Продажу лицензионных продуктов осуществляют сотрудники компаний - авторизованных партнеров ведущих мировых производителей программного обеспечения, квалифицированные специалисты. Покупатель может рассчитывать на профессиональную консультацию по выбору оптимального решения для стоящих перед ним задач.

− Повышение функциональности

− Если у вас возникнут пожелания к функциональности продукта, вы имеете возможность передать их разработчикам; ваши пожелания будут учтены при выпуске новых версий продукта. Приобретая нелицензионное программное обеспечение, вы очень рискуете.

При использовании нелицензионного, то есть измененной пиратами версии, программного продукта, могут возникнуть ряд проблем.

− Некорректная работа программы. Взломанная программа– это изменённая программа, после изменений не прошедшая цикл тестирования.

− Нестабильная работа компьютера в целом.

− Проблемы с подключением периферии (неполный набор драйверов устройств).

− Отсутствие файла справки, документации, руководства.

− Невозможность установки обновлений.

− Отсутствие технической поддержки продукта со стороны разработчика.

− Опасность заражения компьютерными вирусами (от частичной потери данных до полной утраты содержимого жёсткого диска) или другими вредоносными программами.

− Организация обновления программного обеспечения с использованием сети Интернет

− Любая операционная система, как и программные продукты, через какое-то время после установки должна обновляться. Обновления выпускаются для:

− устранения уязвимости в системе безопасности;

− обеспечения совместимости со вновь появившимися на рынке комплектующими компьютеров;

− оптимизации программного кода;

− повышения производительности всей системы.

Желательно обновлять компьютер как можно чаще. В этом случае использования автоматического обновления, операционная система Windows устанавливает новые обновления, как только они становятся доступными. Если не устанавливать обновления, то компьютер может подвергнуться риску в плане безопасности или же могут возникнуть нежелательные неполадки в работе Windows или программ. Каждый день появляется все больше и больше новых вредоносных программ, использующих уязвимости Windows и другого программного обеспечения для нанесения ущерба, и получения доступа к компьютеру и данным. Обновления Windows и другого программного обеспечения позволяют устранить уязвимости вскоре после их обнаружения. Если отложить установку обновлений, компьютер может стать уязвимым для таких угроз.

Практический материал:

Задание 1.

Найти в Интернет закон РФ «Об информации, информатизации и защите информации». Заполнить таблицу.

Алгоритм действий при выполнении самостоятельной работы

1. Найти в Интернет закон РФ «Об информации, информатизации и защите информации».

2. Найти определения понятиям, приведенным в таблице.

Понятие	**Определение**
информация	
информационные технологии	
информационно-телекоммуникационная сеть	
доступ к информации	
конфиденциальность информации	

электронное сообщение	
документированная информация	

Задание 2.

Работа с ресурсами в сети Интернет.

Алгоритм действий при выполнении самостоятельной работы

1. Открыть браузер и найти нужные понятия.
2. Ознакомиться с новыми понятиями.
3. Заполните таблицу.

Понятие	**Значение понятия**
Программное обеспечение (ПО) – это	
Виды ПО	
Системное программное обеспечение включает в себя	
Операционная система -это	
Состав ОС	
Задачи ОС	
Прикладное программное обеспечение - это	
К ПО общего назначения относятся:	
К ПО специального назначения относятся:	

Задание 3.

Изучите инструкцию по обновлению программного обеспечения через Интернет.

Рекомендуется включить автоматическое обновление.

− Для автоматического обновления программ необходимо войти в систему с учетной записью «Администратор».

− Нажмите кнопку Пуск, выберите команду Панель управления и два раза щелкните значок Автоматическое обновление.

− Выберите вариант Автоматически (рекомендуется).

− Под вариантом автоматически загружать и устанавливать на компьютер рекомендуемые обновления выберите день и время, когда операционная система Windows должна устанавливать обновления.

− Автоматическое обновление обеспечивает установку первоочередных обновлений, которые включают в себя обновления безопасности и другие важные обновления, помогающие защитить компьютер. Также рекомендуется регулярно посещать веб-узел Windows Update (http://www.microsoft.com/) для получения необязательных обновлений, например, рекомендованных обновлений программного обеспечения и оборудования, которые помогут улучшить производительность компьютера.

Тема 7.4. Проблема информационной безопасности

Цель: Понимать проблемы информационной безопасности, настраивать антивирусное программное обеспечение.

Теоретический материал:

Проблема информационной безопасности - это одно из наиболее актуальных и важных вызовов в современном цифровом обществе. Она связана с защитой информации, данных, информационных систем и сетей от угроз, атак, несанкционированного доступа и других

рисков, которые могут нанести ущерб как индивидуальным пользователям, так и организациям, государствам и обществу в целом.

Основные аспекты проблемы информационной безопасности:

1. **Киберпреступления:** Различные формы киберпреступлений, такие как хакерские атаки, вирусы, фишинг, кража данных и деньги, наносят огромные убытки компаниям и частным лицам.
2. **Утечки данных:** Несанкционированный доступ к личным данным, финансовой информации, медицинским записям и другим конфиденциальным данным может привести к серьезным последствиям для личной жизни и безопасности.
3. **Кибершпионаж:** Государства и хакерские группы могут вести кибершпионаж с целью получения конфиденциальной информации, в том числе государственных секретов и коммерческих данных.
4. **Отказ в обслуживании (DDoS-атаки):** Атаки, направленные на перегрузку серверов и сетей, могут привести к недоступности веб-сайтов и онлайн-сервисов.
5. **Недостаточная защита:** Несоблюдение стандартов безопасности и уязвимости в программном обеспечении и системах может создать точки входа для атакующих.
6. **Социальная инженерия:** Злоумышленники могут использовать манипуляции и обман, чтобы получить доступ к конфиденциальной информации.
7. **Цифровое неравенство:** Некоторые группы и регионы могут иметь ограниченный доступ к информационным технологиям и знаниям о безопасности, что делает их более уязвимыми к киберугрозам.
8. **Сложность технологий:** Рост сложности информационных технологий и внедрение «Интернета вещей» приводит к увеличению поверхности атак и рисков.

Решение проблемы информационной безопасности требует комплексного подхода, который включает в себя технические, организационные, образовательные и правовые меры. Это включает в себя усиление сетевых защит, внедрение многофакторной аутентификации, обучение пользователям правилам безопасности, разработку законов и политик, регулирующих использование информационных технологий, и многое другое.

Список информационных источников:

3.2.1. Основные печатные издания

1. Информатика: учебное пособие для СПО / составители С.А. Рыбалка, Г.А. Шкатова. — Саратов: Профобразование, 2021. — 171 с. — ISBN 978-5-4488-0925-5. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROF образование: [сайт]. — URL: https://profspo.ru/books/99928

2. Борисов, Р.С. Информатика (базовый курс): учебное пособие / Р.С. Борисов, А.В. Лобан. — Москва: Российский государственный университет правосудия, 2021. — 304 с. — ISBN 978-5-93916-445-0. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROF образование: [сайт]. — URL: https://profspo.ru/books/34551

3. Гальченко, Г. А. Информатика для колледжей: учебное пособие. Общеобразовательная подготовка / Г. А. Гальченко, О. Н. Дроздова. — Ростов-на-Дону: Феникс, 2021. — 382 с. — ISBN 978-5-222-27454-5. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROF образование: [сайт]. — URL: https://profspo.ru/books/59322

4. Цветкова, А. В. Информатика и информационные технологии: учебное пособие для СПО / А. В. Цветкова. — Саратов: Научная книга, 2021. — 190 с. — ISBN 978-5-9758-1891-1. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROF образование: [сайт]. — URL: https://profspo.ru/books/87074

3.2.2. Основные электронные издания

1. Федеральный центр информационно-образовательных ресурсов — URL: www.fcior.edu.ru

2. Единая коллекция цифровых образовательных ресурсов — URL: www.school-collection.edu.ru

3. Единое окно доступа к образовательным ресурсам Российской Федерации — URL: http://window.edu.ru/

3.2.3. Дополнительные источники

1. Лебедева, Т. Н. Информатика. Информационные технологии: учебно-методическое пособие для СПО / Т. Н. Лебедева, Л. С. Носова, П. В. Волков. — Саратов: Профобразование, 2021. — 128 с. — ISBN 978-5-4488-0339-0. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROF образование: [сайт]. — URL: https://profspo.ru/books/86070

2. Жилко, Е. П. Информатика. Часть 1: учебное пособие для СПО / Е. П. Жилко, Л. Н. Титова, Э. И. Дяминова. — Саратов, Москва: Профобразование, Ай Пи Ар Медиа, 2021. — 182 с. — ISBN 978-5-4488-0873-9, 978-5-4497-0637-9. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROF образование: [сайт]. — URL: https://profspo.ru/books/97411