

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
МОРСКОЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
к практическим занятиям студентов
09.02.07 Информационные системы и программирование
Общепрофессиональный цикл

Разработали: преподаватели:
Батыргареева С.Ф.,
Корепанова Н.Л.,
Лебедева М.А.,
Сушкова Ю.А.

2023г

Лабораторная работа №1

Создание и нормализация таблиц базы данных

Цель работы : научиться определять структуру таблицы и устанавливать свойства ее полей в режиме конструктора таблиц; определять простой и составной первичный ключ; приводить таблицы к нормальной форме; уметь устанавливать межтабличные связи.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предьявляется реализованная средствами MicrosoftAccess база данных для заданной предметной области

Краткие теоретические сведения:

Понятие нормализации схем баз данных неразрывно связано с понятием ограничений, накладываемых системой функциональных зависимостей, и во многом следует из этого понятия.

Исходной точкой любого проектирования базы данных является представление предметной области в виде одного или нескольких отношений, и на каждом шаге проектирования производится некоторый набор схем отношений, обладающих «улучшенными» свойствами. Таким образом, процесс проектирования представляет собой процесс нормализации схем отношений, причем каждая следующая нормальная форма обладает свойствами, в некотором смысле лучшими, чем предыдущая.

Каждой нормальной форме соответствует определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. Примером может служить ограничение первой нормальной формы – значения всех атрибутов отношения атомарны.

Нормальная форма — требование, предьявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц). Метод нормальных форм (НФ) состоит в сборе информации о объектах решения задачи в рамках одного отношения и последующей декомпозиции этого отношения на несколько взаимосвязанных отношений на основе процедур нормализации отношений.

Цель нормализации: исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении, редактировании и удалении кортежей(строк таблицы).

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- 1) первая нормальная форма (1 NF);
 - 2) вторая нормальная форма (2 NF);
 - 3) третья нормальная форма (3 NF);
 - 4) нормальная форма Бойса – Кодда (BCNF);
 - 5) четвертая нормальная форма (4 NF);
 - 6) пятая нормальная форма, или нормальная форма проекции-соединения (5 NF или PJ/NF).
- Основные свойства нормальных форм состоят в следующем:

- 1) каждая следующая нормальная форма в некотором смысле лучше предыдущей нормальной формы;
- 2) при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

Рассмотрим первые три нормальные формы

Первая нормальная форма

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

Вторая нормальная форма

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК).

Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

Третья нормальная форма

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Иными словами, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Порядок выполнения работы

Задание 1. Создайте базу данных о студентах, поступивших в ВУЗ, состоящую из двух таблиц: «Группа» и «Студент». Дана исходная таблица 1. Привести ее к нормальной форме.

Таблица 1.

Группа	Кол-во студенто в в группе	ФИО	Пол	Номер студен та	Год рождения	Адрес	Проходно й балл	Код специаль ности
11	25	Иванова	Ж	2	05.02.87		4,5	611
11	25	Сидоров	М	17	10.06.87		4,8	611
11	25	Петров	М	10	11.12.86		4,3	611
12	20	Мазуров	М	11	29.10.86		4,5	615
12	20	Бойко	Ж	1	06.01.87		5,0	615
12	20	Пашкова	Ж	9	15.12.86		4,3	615
13	20	Степанов	М	13	27.09.86		4,9	3514
13	20	Бородина	Ж	5	20.04.87		5,0	3514

Для создания структуры созданных таблиц необходимо определить основные характеристики полей таблиц, свойства полей, назначить первичный ключ. В таблицах 2 и 3 заполнить основные характеристики полей полученных таблиц. Заполненные таблицы 2 и 3 занести в отчет.

Таблица 2. Описание свойств полей таблицы «Группа»

Имя Поля	Ключ. поле	Тип данных	Размер, байт	Число десятич. знаков	Подпись поля	Условие на значение
						Сообщение об ошибке
		Текстовый	5			
		Текстовый	6			
		Числовой	Байт	0		
		Числовой	Одинарное с пл. точкой	2		

Таблица 3. Описание свойств полей таблицы «Студент»

Имя Поля	Ключ. поле	Тип данных	Размер, Байт	Число десятич. знаков	Подпись поля	Условие на значение
						Сообщение об ошибке
		Текстовый	5			
		Текстовый	2			
		Текстовый	15			
		Подстановкой (м, ж)	1			
		Дата/Время	Краткий форм. Д.			
		Текстовый	25			
		Числовой	Одинарное с пл. точкой	2		

Задание 2. Создание структуры базы данных

1. Запустите программу Microsoft Access.
2. В первом диалоговом окне Access установите переключатель «Новая база данных».
3. В следующем окне выберите папку и укажите в качестве имени файла новой базы данных **свою фамилию**. Щелкните на кнопке **Создать**.
4. В окне новой базы данных из списка объектов выберите объект **Таблицы**. Создайте структуру таблицы Группа в режиме Конструктора. При задании **Условия на значение** для полей пользуйтесь Построителем выражений.
5. Задайте ключевое поле.
6. Сохраните структуру первой таблицы под именем **Группа**.
7. Создайте структуру таблицы Студент. Условие задавайте Построителем выражений.
8. Задайте составной уникальный ключ, состоящий из двух полей: **НГ** (номер группы) и **НС** (номер студента в группе). Для выделения полей пользуйтесь клавишей CTRL.
9. Сохраните таблицу под именем **Студент**.

Задание 3. Ввод данных в таблицы

10. В окне базы данных откройте таблицу **Группа** двойным щелчком на ее значке.
11. Введите записи.
12. Закройте таблицу **Группа**.
13. Откройте таблицу **Студент**.
14. Введите записи.
15. Закройте таблицу **Студент**.

Задание 4. Установка связей между таблицами

16. Выполните команду **Работа с базами данных-Схема данных**.
17. В окне **Схемы данных** расположите все таблицы.
18. Установите между таблицами **Группа**• **Студент** связь по простому ключу **НГ**.
19. В открывшемся окне обратите внимание, что тип отношения установится *один-ко-многим*. Установите все флажки, обеспечивающие целостность во взаимосвязанных таблицах.
20. Установите все флажки, обеспечивающие целостность во взаимосвязанных таблицах.
21. Сохраните схему данных и закройте окно.
22. Откройте таблицы **Группа** и **Студент**. Проверьте, как влияют установленные параметры поддержания связной целостности:
 - в таблице **Группа** измените номер группы (что произошло в подчиненной таблице **Студент**? Ответ запишите в отчет);
 - в подчиненную таблицу **Студент** попытайтесь ввести запись с номером группы, не представленным в главной таблице **Группа**; Что произошло? Ответ запишите в отчет.
 - в главной таблице **Группа**, удалите запись, которая вводилась последней (выделите ее и выполните команду **Правка-Удалить**); что произошло в подчиненной таблице **Студент**? Ответ запишите в отчет.
23. Закройте таблицы **Группа** и **Студент**.

IV. Ввод и корректировка записей во взаимосвязанных таблицах

24. Откройте таблицу **Группа**.
25. Создайте заново удаленную запись.
26. Используя значки «+» в строке записи, введите несколько записей в подчиненную таблицу **Студент**. (если «плюсы» не отображены, выполните команду **Вставка-Подтаблица**).
27. Покажите работу преподавателю. Закройте окно **Базы данных** и **Access**.

Контрольные вопросы

1. Что такое нормализация базы данных?
2. Какие нормальные формы вы знаете? Дайте им определения.
3. Что такое декомпозиция?

Проектирование структуры базы данных.

Цель работы: Изучить этапы проектирования реляционной БД; изучить и практически освоить процесс создания новой БД средствами СУБД MS Access, включая разработку макета таблиц в режиме Конструктора и построение схемы БД.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предъявляется реализованная средствами Microsoft Access база данных для заданной предметной области

Краткие теоретические сведения:

Прежде чем приступить к созданию таких объектов базы данных, как таблицы, формы и отчеты, нужно разработать их проект. Главное назначение проекта—выработка четкого пути, по которому нужно следовать при его реализации. База данных—достаточно сложный объект, и время, затраченное на ее планирование, может значительно сократить сроки ее разработки. Отсутствие продуманной структуры базы данных приводит к необходимости постоянной переделки и перенастраиванию объектов базы данных, таких, как формы и таблицы.

Проектирование базы данных целесообразно начать с краткого описания отчетов, списков и других документов, которые необходимо получить с помощью БД. Далее следует разработать эскиз объектов, требуемых для получения необходимых результатов и определит связи между этими объектами.

При разработке эскиза необходимо ответить на следующие вопросы:

Какими данными мы располагаем?

Какие данные будут содержать таблицы?

Какой тип и какие свойства должны иметь данные в каждом поле таблицы?

Как эти таблицы будут связаны друг с другом?

Законченный план должен содержать подробное описание всех таблиц (имена полей, типы данных и их свойства), а также связей между ними.

Проектирование предусматривает этапы создания проекта базы данных от концепции до реального воплощения. Этапы проектирования базы данных:

1. Исследование предметной области и формулировка основных допущений (накладываемых условий). На этом этапе составляется список всех форм и отчетов, которые могут быть затребованы пользователями вашей БД.

2. Анализ данных. Составить перечень всех элементов данных, входящих в формы и отчеты и сгруппировать их в таблицы БД.

3. Установить, какие взаимосвязи существуют между элементами данных. Определить первичные и вторичные (внешние) ключи отношений. Организовать поля данных в таблицах, причем это необходимо сделать, следуя 4-м правилам нормализации:

Правило 1: Каждое поле таблицы должно представлять уникальный тип информации. Это правило означает, что необходимо избавиться от повторяющихся полей и разделить составные поля на отдельные элементы данных.

Правило 2: Каждая таблица должна иметь уникальный идентификатор или первичный ключ, который может состоять из одного или нескольких полей.

Правило 3: В таблице не должно быть данных, не относящихся к объекту, определяемому первичным ключом.

Правило 4: Независимость полей. Это правило означает возможность изменять значения любого поля (не входящего в первичный ключ) без воздействия на данные других полей.

Результатом 3 этапа должна явиться группа таблиц, удовлетворяющих правилам нормализации. На этом же этапе необходимо установить связи между таблицами.

Порядок выполнения работы

В качестве предметной области выбрана деятельность фирмы по продаже автомобилей марки Toyota. Поставлена задача: упорядочить информацию о клиентах фирмы, ассортименте продукции и сформированных заказах.

Этапы 1-3 проектирования БД изучить теоретически, 4-5 выполнить практически.

Оформить отчет о выполнении всех этапов задания

Ответить на контрольные вопросы

1-й этап. Определение цели проектирования БД.

Проектируемая реляционная БД должна содержать структурированную информацию о клиентах, продаваемых автомобилях и заказах.

Проектируемая БД должна поддерживать выполнение, как минимум, следующих основных функций: ввод и обновление информации, просмотр и удаление.

2-й этап. Разработка информационно-логической модели предметной области.

Вся информация о предметной области может быть логично разделена на 3 таблицы:

Клиенты, Автомобили, Заказы.

При этом выполняются основные требования к содержанию таблиц:

1. Каждая таблица содержит информацию только на одну тему.
2. Информация в таблицах не дублируется.
3. Для связи между таблицами заданы **первичные ключи**, однозначно определяющие каждую запись в таблице.

Содержание базовых таблиц приведено ниже:

Таблица Клиенты	Таблица Автомобили	Таблица Заказы
1. Код клиента (ключ)	1. Код модели (ключ)	1. Код заказа (ключ)
2. Фамилия	2. Модель	2. Код клиента
3. Имя	3. Мощность двигателя	3. Код Модели
4. Отчество	4. Цвет	4. Дата заказа
5. Адрес	5. Количество дверей	5. Скидка, %
6. Телефон	6. Заводская цена	6. Оплачено
	7. Издержки (транспортные, предпродажные)	
	8. Специальная модель	
	9. Дополнительное оснащение	

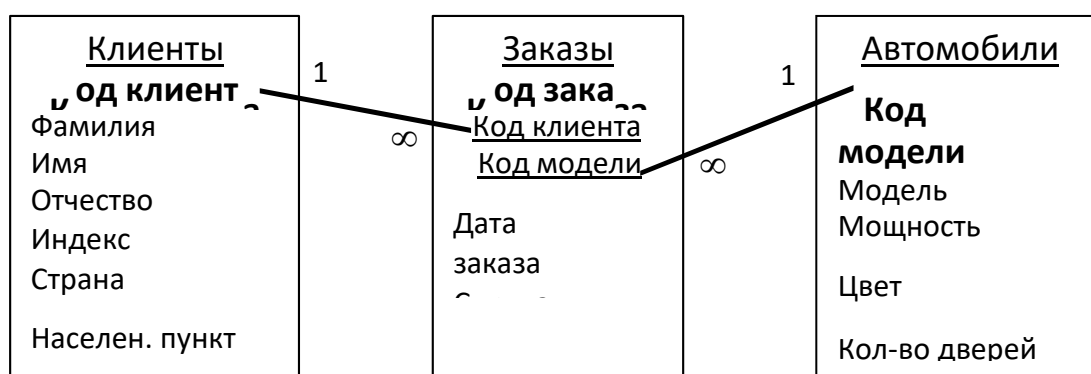
При разработке полей для каждой таблицы необходимо учитывать:

- Каждое поле должно быть связано с темой таблицы.
- Не включать в таблицу данные, которые являются результатом вычисления.
- Информацию следует разбивать на наименьшие логические единицы (Например, поля «Индекс», «Страна», «Населенный пункт», «Почтовый адрес», а не общее поле «Адрес»).

3-й этап. Определение отношений между таблицами.

Поскольку для проектируемой БД выполнены требования нормализации, между таблицами **Клиенты-Заказы** и **Автомобили-Заказы** могут быть установлены **одно-многочленные отношения** ($1 : \infty$), которые поддерживаются реляционной СУБД.

Связь между таблицами устанавливается с помощью **ключей** Код клиента и Код модели, которые в главных таблицах Клиенты и Автомобили являются первичными, а в таблице-связке Заказы - внешними.



4-й этап. Создание таблиц БД средствами СУБД MS Access.

4.1. Загрузить СУБД MS Access. Создать Новую базу данных.

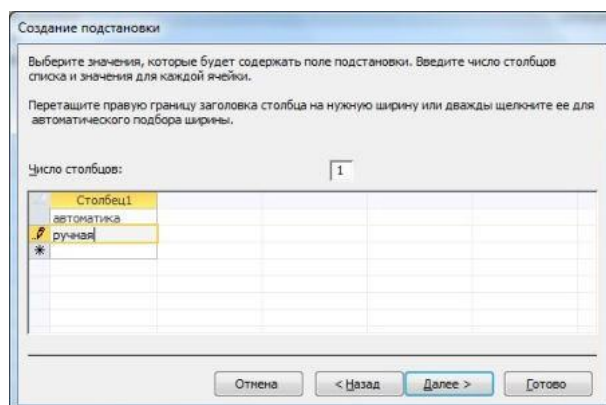
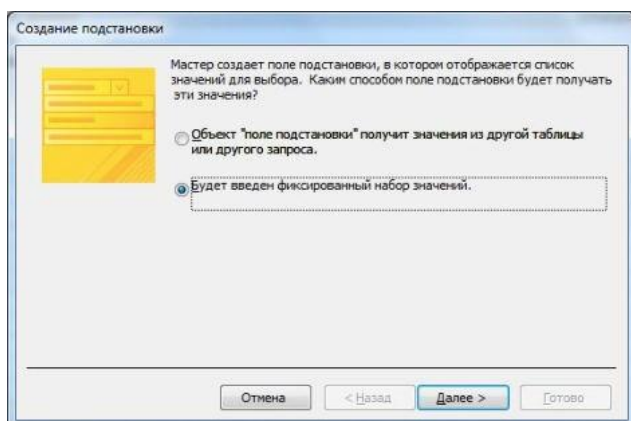
4.2. Создать макет таблицы Автомобили в режиме Конструктора, используя нижеприведенные данные об именах полей, их свойствах и типах данных.

*Все поля, за исключением поля Другое оснащение, должны быть обязательными для заполнения (Свойство Обязательное поле: Да).

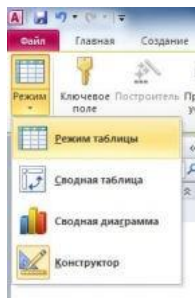
Имя поля	Тип данных	Описание	Свойства поля
Код модели	Числовой	Ключевое поле, код модели по заводскому каталогу	Индексированное поле: Да/Совпадения не допускаются Ключевое поле задается в меню Правка/Ключевое поле
Модель	Текст	Тип кузова	Размер поля: 20, Значение по умолчанию: Corolla Индексированное поле: Да/Совпадения допускаются (одна и та же модель может встречаться в БД многократно с различными вариантами оснащения)
Мощность	Текст	Мощность двигателя (кВт/л.с.)	Размер поля: 10 Индексированное поле: Нет
Цвет	Текст	Цвет кузова	Размер поля: 20 Индексированное поле: Нет
Количество дверей	Числовой	2 или 4	Размер поля: Байт Формат: Основной Число десятичных знаков: 0 Значение по умолчанию: 4 Условие на значение: 2 Or 4 Сообщение об ошибке: Данное поле может содержать

Имя поля	Тип данных	Описание	Свойства поля
			только 2 или 4 Индексированное поле: Нет
Коробка передач	*Мастер подстановок, фиксированный набор значений	Автоматика или ручная	Размер поля: 12 Значение по умолчанию: Ручная Условие на значение: “Ручная” От “Автоматика” Сообщение об ошибке: Допустимы только значения “Ручная” или “Автоматика” Индексированное поле: Нет
Обивка	*Мастер подстановок, фиксированный набор значений	Велюр, кожа, ткань	Размер поля: 10 Индексированное поле: Нет
Другое оснащение	Мето	Дополнительные аксессуары	Значение по умолчанию: Радио/плеер, раздвижная крыша
Заводская цена	Денежный	Заводская продажная нетто-цена	Формат: Денежный Число десятичных знаков: Auto Индексированное поле: Нет
Транспортные издержки	Денежный	Издержки на доставку	Формат: Денежный Число десятичных знаков: Auto Индексированное поле: Нет
Предпродажные издержки	Денежный	Издержки на предпродажную подготовку	Формат: Денежный Число десятичных знаков: Auto Значение по умолчанию: 105 Индексированное поле: Нет
Специальная модель	Логический	Спец. модель или стандартная	Формат: Да/Нет

*Мастер подстановок позволяет создать для указанных полей список, из которого выбирается значение.



4.3. Перейти в режим Таблицы, сохранив созданный макет таблицы под именем Автомобиля



4.4. Добавить в таблицу Автомобили 3 записи:

Код модели	12580	12653	12651
Модель	Corolla Liftback	Corolla CompactGT	Corolla CompactXL
Мощность	69/90	100/139	90/135
Цвет	Бутылочное стекло	Черный	Небесно-голубой
Количество дверей	4	2	2
Коробка передач	Автоматика	Ручная	Ручная
Обивка	Ткань	Кожа	Велюр
Другое оснащение	Радио/плеер, раздвижная крыша, лаковое покрытие “Металлик”	Радио/плеер, раздвижная крыша, алюмин. дворники	Электро-подъемник окон, раздвижная крыша
Заводская цена	39200	41100	37900
Транспортные издержки	1200	975	1050
Предпродажные издержки	105	105	105
Специальная модель	Нет	Да	Да

4.5. Создать макет таблицы Клиенты в режиме Конструктора.

*Обязательные поля Код клиента, Фамилия, Страна.

Имя поля	Тип данных	Описание	Свойства поля (определяют правила сохранения, отображения и обработки данных в поле)
Код клиента	Счетчик	Ключевое поле, уникальный номер клиента в БД	Индексированное поле: Да/Совпадения не допускаются <i>Ключевое поле задается в меню Правка/Ключевое поле</i>
Фамилия	Текст	Фамилия	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
Имя	Текст	Имя	Размер поля: 20, Индексированное поле: Да/Совпадения допускаются
Отчество	Текст	Отчество	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
Индекс	Числовой	Почтовый индекс	Размер поля: Длинное целое, Индексированное поле: Да/Совпадения допускаются
Страна	Текст	Название страны	Размер поля: 20, Индексированное поле: Да/Совпадения допускаются
Населенный пункт	Текст	Название населенного пункта	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются

Почтовый адрес	Текст	Почтовый адрес	Размер поля: 50, Индексированное поле: Нет
Телефон	Текст	Контактный телефон	Размер поля: 20, Индексированное поле: Нет

4.6. Добавить в таблицу Клиенты 3 записи. (Перейти в режим Таблицы, сохранив макет таблицы под именем Клиенты)

4.7. Создать в режиме Конструктора макет таблицы Заказы.

*Все поля, за исключением поля Скидка, являются обязательными для заполнения.

Имя поля	Тип данных	Описание	Свойства поля (определяют правила сохранения, отображения и обработки данных в поле)
Код заказа	Счетчик	Ключевое поле, уникальный номер заказа	Индексированное поле: Да/Совпадения не допускаются <i>Ключевое поле задается в меню Правка/Ключевое поле</i>
Код модели	Числовой, *Мастер подстановок	Внешний ключ, для связи с таблицей Автомобиля	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
Код клиента	Числовой, *Мастер подстановок	Внешний ключ, для связи с таблицей Клиенты	Размер поля: Длинное целое Индексированное поле: Да, допускаются совпадения
Дата заказа	Дата/время	Дата формирования заказа ДД.ММ.ГГ	Формат: Краткий формат даты Индексированное поле: Да/Совпадения допускаются
Скидка	Числовой	Размер скидки в %	Размер поля: Одинарное с плавающей точкой Формат: Процентный Условие на значение: Between 0 And 1

* Используя Мастер подстановок, сформировать для полей Код клиента и Код модели список выбора из таблиц Клиенты и Автомобиля (сбросить флажок Скрыть ключевое поле).

4.8. Добавить 5 записей в таблицу Заказы.

5-й этап. Создание схемы данных БД (связей между таблицами).

5.1. Выполнить опцию Схема данных из вкладки Работа с базами данных. В диалогом окне Добавление таблицы последовательно добавить все три таблицы. Закрыть диалоговое окно.

5.2. Установить связь между таблицами Клиенты-Заказы, Автомобиля-Заказы: выделить ключевое поле в главной таблице (Клиенты или Автомобиля) и перетащить его на соответствующее поле таблицы-связки Заказы. Обеспечить целостность данных.

5.3. Сохранить макет схемы данных.

Контрольные вопросы

1. Перечислите основные этапы проектирования реляционной БД. Кратко поясните содержание каждого этапа.
2. Какие требования предъявляют к содержанию таблиц реляционной БД?
3. Приведите порядок создания макета таблицы в режиме Конструктора.
4. Для чего в каждой таблице задается первичный ключ? В чем различие между первичным и внешним ключом?

5. Порядок
ок
формирова
ния
схемы
БД.

полнение
табличного
файла.
Модификация
структуры
табличного
файла.

Лабораторная
работа
№3

Цель работы: научиться добавлять данные в таблицу и редактировать имеющиеся;
научиться разбивать данные на несколько таблиц; менять структуру таблиц; обеспечивать целостность данных.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предьявляется реализованная средствами MicrosoftAccess база данных для заданной предметной области

Краткие теоретические сведения:

Сформировав таблицы и установив ключевое поле для каждой таблицы, между таблицами можно установить взаимосвязи, которые будут поддерживаться при создании форм, отчетов и запросов и задать условия целостности данных этих таблиц.

Существует 3 типа связей:

"один к одному" – каждой записи одной таблицы соответствует только одна запись в другой; "один ко многим" – каждой записи одной таблицы может соответствовать несколько записей в другой таблице или "многие к одному" – в таблице может быть несколько записей, соответствующих только одной записи в другой таблице;

"многие ко многим" – множеству записей одной таблицы соответствует множество записей другой таблицы.

При определении связи ключ в одной таблице содержит ссылки на конкретные записи в другой таблице. Поле, не являющееся ключевым для данной таблицы, но значения которого являются значениями первичного ключа другой таблицы, называют внешним ключом

Открытие,

редактирование

Содержимое поля внешнего ключа (значения свойства) должно совпадать с содержимым ключевого поля. Эти поля также могут иметь одинаковые имена.

позволяет более компактно хранить информацию. Данные в главной таблице индексируют и, используя эти индексы, связывают с подчиненными таблицами. В данной лабораторной работе мы будем приводить, созданную ранее базу данных, к такому виду.

Разделение данных на две таблицы. Пополнение табличного файла.

1. Откройте базу данных **Европа**
2. Проверьте содержимое таблицы и при необходимости внесите соответствующие коррективы. Для этого откройте таблицу **Страны Европы** в режиме **таблица**.

Албания

Площадь, 28 748 кв. кмСтолица ,Тирана

П
о
р
я
д
о
к
в
ы
п
о
л
н
е
н
и
я
р
а
б
о
т
ы

В процессе ввода данных очень часто возникает необходимость защитить оператора от ошибки и разграничить доступ к важной информации. Для этого данные разных категорий разделяют по разным таблицам, кроме этого, как правило, такое разделение

а, Лек

Ч
и
с
л
о
ж
и
т
е
л
е
й
,
3
1
4
9
0
0
0
О
с
н
о
в
н
а
я
р
е
л
и
г
и
я
,
А
т
е
и
з
м
Д
е
н
е
ж
н
а
я
е
д
и
н
и
ц

Гос. стройРеспублика

Андорра

Площадь, 468 кв. км

Столица Андорра-ла-Вьеха

Число жителей 51 400

Основная религияХристианство (католики)

Денежная единица Франк

Гос. стройКняжество

Бельгия

Площадь, 30 518 кв. км

Столица Брюссель

Число жителей 9 865 000

Основная религияХристианство (католики)

Денежная единица Франк

Гос. стройМонархия

Болгария

Площадь, 110 994 кв. км

Столица София

Число жителей 8 978 000

Основная религияХристианство (православные)

Денежная единица Лев

Гос. стройРеспублика

Дания

Площадь, 43 092 кв. км

Столица Копенгаген

Число жителей 5 130 000

Основная религияХристианство (протестанты)

Денежная единица Крона

Гос. стройМонархия

Финляндия

Площадь, 338 145 кв. км

Столица Хельсинки

Число жителей 4 952 000

Основная религияХристианство (протестанты)

Денежная единица Марки

Гос. стройРеспублика

Франция

Площадь, 543 965 кв. км

Столица Париж

Число жителей 55 860 000

Основная религияХристианство (католики)

Денежная единица Франк

Гос. стройРеспублика

Греция

Площадь, 131 957 кв. км

Столица Афины

Число жителей 10 055 000

Основная религияХристианство (православные)

Денежная единица Драхма

Гос. стройРеспублика

Ирландия

Площадь, 70 285 кв. км

Столица Дублин

Число жителей 3 553 000

Основная религияХристианство (католики)

Денежная единица Фунт

Гос. стройРеспублика

Исландия

Площадь, 103 000 кв. км

Столица Рейкьявик

Число жителей 248 000

Основная религияХристианство (протестанты)

Денежная единица Крона

Гос. стройРеспублика

Италия

Площадь, 301 277 кв. км

Столица Рим

Число жителей 57 401 000

Основная религияХристианство (католики)

Денежная единица Лира

Гос. стройРеспублика

Югославия

Площадь, 255 804 кв. км

Столица Белград

Число жителей 23 591 000

Основная религияХристианство (православные)

Денежная единица Динар

Гос. стройРеспублика

Лихтенштейн

Площадь, 160 кв. км

Столица Вадуц

Число жителей 27 840

Основная религияХристианство (католики)

Денежная единица Франк

Гос. стройМонархия

Люксембург

Площадь, 2 586 кв. км

Столица Люксембург

Число жителей 372 000

Основная религияХристианство (католики)

Денежная единица Франк

Гос. строй Герцогство

Мальта

Площадь, 316 кв. км

Столица Валлетта

Число жителей 347 000

Основная религия Христианство (католики)

Денежная единица Фунт

Гос. строй Республика

Монако

Площадь, 2 кв. км

Столица Монако

Число жителей 28 000

Основная религия Христианство (католики)

Денежная единица Франк

Гос. строй Княжество

Нидерланды

Площадь, 41 863 кв. км

Столица Амстердам

Число жителей 14 741 000

Основная религия Христианство (католики)

Денежная единица Флорин

Гос. строй Монархия

Норвегия

Площадь, 323 878 кв. км

Столица Осло

Число жителей 4 202 000

Основная религия Христианство (протестанты)

Денежная единица Крона

Гос. строй Монархия

Польша

Площадь, 312 683 кв. км

Столица Варшава

Число жителей 37 864 000

Основная религия Христианство (католики)

Денежная единица Злотый

Гос. строй Республика

Португалия

Площадь, 92 389 кв. км

Столица Лиссабон

Число жителей 10 349 000

Основная религия Христианство (католики)

Денежная единица Эскудо

Гос. строй Республика

Румыния

Площадь, 237 500 кв. км

Столица Бухарест

Число жителей 23 014 000

Основная религия Христианство (православные)

Денежная единица Леи

Гос. строй Республика

Сан-Марино

Площадь, 61 кв. км

Столица Сан-Марино

Число жителей 22 830

Основная религия Христианство (католики)

Денежная единица Лира

Гос. строй Республика

Швейцария

Площадь, 41 293 кв. км

Столица Берн

Число жителей 6 626 000

Основная религия Христианство (католики)

Денежная единица Франк

Гос. строй Республика

Испания

Площадь, 504 783 кв. км

Столица Мадрид

Число жителей 38 996 000

Основная религия Христианство (католики)

Денежная единица Песета

Гос. строй Монархия

Великобритания

Площадь, 244 110 кв. км

Столица Лондон

Число жителей 57 006 000

Основная религия Христианство (протестанты)

Денежная единица Фунт

Гос. строй Монархия

Швеция

Площадь, 449 964 кв. км

Столица Стокгольм

Число жителей 8 415 000

Основная религия Христианство (протестанты)

Денежная единица Крона

Гос. строй Монархия

Чехо-Словакия

Площадь, 127 900 кв. км

Столица Прага

Число жителей 15 604 000

Основная религия Христианство (католики)

Денежная единица Крона

Гос. строй Республика

Венгрия

Площадь, 93 031 кв. км

Столица Будапешт
Число жителей 10 591 000
Основная религияХристианство (католики)
Денежная единица Форинт
Гос. стройРеспублика

Германия

Площадь, 357 042 кв. км
Столица Бонн
Число жителей 77 370 000
Основная религияХристианство (протестанты)

Денежная единица Марка
Гос. стройРеспублика

Австрия

Площадь, 83 857 кв. км
Столица Вена
Число жителей 7 557 000
Основная религияХристианство (католики)
Денежная единица Шиллинг
Гос. стройРеспублика

3. Настройте ширину столбцов таблицы **Страны Европы** по ширине данных.

4. Создайте таблицу **Религия** с полями:

Название поля	Тип данных
Код_религии	Счетчик
Религия	Текстовый

5. Поле **Код_религии** сделайте ключевым

6. Создайте таблицу **Строй** с полями:

Название поля	Тип данных
Код_строя	Счетчик
Строй	Текстовый

7. Поле **Код_строя** сделайте ключевым

8. Заполните эти таблицы, выбрав необходимые данные из исходной таблицы (таким образом, напротив каждой религии и каждого строя будет стоять его код)

9. Откройте таблицу **Страны Европы** и замените названия в полях **Строй** и **Религия** на соответствующие им коды в ранее созданных таблицах (для автоматизации попробуйте использовать команду замены из меню правка)

10. Откройте таблицу **Страны Европы** в режиме конструктора

11. Для полей **Строй** и **Религия** измените тип данных на числовой

12. Одновременно установив им размер поля **Длинное целое** (так как связываемые поля, как правило, должны иметь одинаковый тип данных)

Установка связи между двумя таблицами

1. Выполните команду **Схема данных** из меню **Работа с базами данных**

2. В диалоговом окне добавления таблиц добавьте в схему все три таблицы

3. На поле **Код_религии** таблицы **Религия** нажмите левую клавишу мыши и удерживая ее перетащите на поле **Религия** таблицы **Страны Европы**

В появившемся диалоговом окне необходимо установить нужную связь: флажок напротив опции **обеспечение целостности данных** означает, что перед тем как занести данные в подчиненную таблицу, программа будет проверять их на соответствие главной. (Таблица **Страны Европы** является подчиненной для таблиц **Религия** и **Строй**). Флажок напротив опции **каскадное обновление связанных полей** означает, что изменения в главной таблице автоматически будут влиять на подчиненную. Флажок напротив опции **каскадное удаление связанных полей** означает, что поля удаленные в главной таблице будут удалены и в подчиненной.

4. Установите все эти флажки

5. Аналогичную операцию проделайте с таблицами **Страны Европы** и **Строй** (Если связь не устанавливается, еще раз проверьте типы данных связываемых полей)

6. Закройте схему с сохранением

7. Попробуйте в таблице **Страны Европы** в полях **Строй** и **Религия** поменять коды на несуществующие в главных таблицах. Получилось? Почему?

Контрольные вопросы

1. Чем поле отличается от записи?

2. Какие структуры ACCESS вы знаете?

3. Для чего служит структура "таблица"?

4. Для чего данным задаются различные типы?
5. Что можно настроить в свойствах поля таблицы?
6. Для чего данные разбивают на несколько таблиц?
7. Какое необходимое условие для создания связи с обеспечением целостности данных?
8. Каким образом при разбиении достигается компактность хранения информации?

Лабораторная работа №4

Индексирование и сортировка таблиц

Цель работы: знакомство с технологией создания и использования простых и составных индексов и первичных ключей. Знакомство с использованием ключевых полей для связывания таблиц. Изучение типов связей.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предьявляется реализованная средствами MicrosoftAccess база данных для заданной предметной области

Порядок выполнения работы

Задание 1. Создание индексов и ключей

1. Создать базу данных База.№3 используя базу БАЗА_ИСХ
База содержит таблицы:

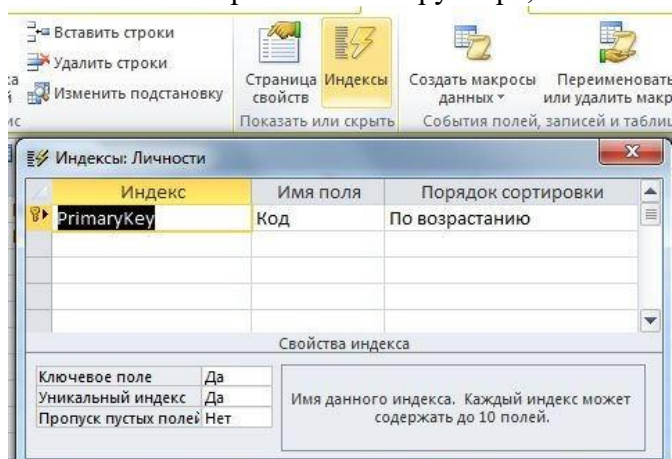
Личности	Адреса	Предметы	Экзамены
ЛичнКод	ЛичнКод	КодПредм	КодПредмета
Фамилия	ПочтКод	Наименование	КодСтудента
Имя	Город	Семестр	Балл
Рожд	Улица	Часы	ДатаЭкз
Пол	Дом	Контроль	
Рост	Корпус		
Адрес	Квартира		
Телефон	Телефон		

Заполнить таблицы экспериментальным содержимым (5-7 записей) используя мастера подстановки

2. Создать первичные ключи в ранее сконструированных таблицах **Личности** и **Адреса**.

2.1. Открыть таблицу *Личности* в режиме *Конструктора таблиц*. Пользуясь контекстным меню поля *ЛичнКод* отметить его как ключевое поле.

2.1.1. Находясь в окне *Личности* в режиме конструктора, вызвать меню Индексы”.



2.2. Открыть таблицу *Адреса* в режиме *Конструктора таблиц*. Пользуясь окном *Индексы*, создать первичный ключ для данной таблицы на основе поля *ЛичнКод*.

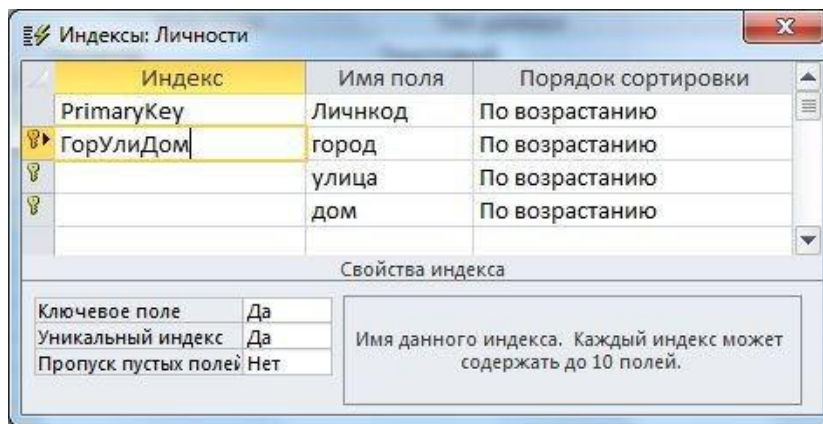
3. Создать составные индексы.

3.1. Открыть для таблицы *Адреса*, находящейся в режиме *Конструктора таблиц*, окно *Индексы* и просмотреть его содержимое.

3.2. Создать новый индекс *Пол*, предназначенный для ускорения процесса поиска и выполнения запросов по данному полю.

3.3. Создать новый составной индекс *ГорУлиДом*, включив в его состав поля *Город*, *Улица*, *Дом*.

3.4. Находясь в окне *Индексы* и переходя от строки *PrimaryKey* к *ГорУлиДом*, просмотреть расположенные ниже *Свойства поля* и дать объяснения увиденным свойствам индексов. Окно *Индексы* для таблицы *Адреса* будет выглядеть примерно так, как показано на рисунке ниже:



3.4.1. Для получения сведений о свойствах индексов прочитать пояснения, размещенные в области *Свойства индекса* окна *Индексы*, а также раздел подсказки (F1), посвященный этому вопросу.

3.5. Создать составной индекс *ФамИмяРожд*, включающий поля *Фамилия*, *Имя*, *Рожд*.

3.6. Открыть для таблицы *Личности*, находящейся в режиме *Конструктора таблиц*, окно *Индексы* и просмотреть его содержимое.

3.6.1. Добавить в составной индекс *ГорУлиДом* поле *Квартира*.

3.6.2. Сделать составной индекс *ГорУлиДом* уникальным. Данный индекс необходим для обеспечения логической достоверности данных (предполагается, что не может быть двух студентов, проживающих в одной квартире).

3.6.3. Проверить работоспособность созданного индекса, введя два одинаковых адреса.

3.7. В таблице **Предметы** для последующего создания индексов сделать следующие коррективы:

3.7.1. Ввести новое числовое поле *Семестр*, имеющее минимальный размер, допускающее ввод только следующих значений [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] и заполнить его данными.

3.7.2. Создать составной индекс *КодНауСем*, включающий поля *КодПредм*, *Наименов*, *Семестр*.

Заметьте, что *КодПредм* не будет первичным ключом таблицы, каким, например, в таблицах **Личности** и **Адреса** является поле *ЛичнКод*, так как в нескольких семестрах может читаться один и тот же предмет с одним кодом и разной продолжительностью.

3.7.3. Установить такие свойства индекса *КодНауСем*, чтобы этот составной индекс стал ключом таблицы и не допускал повторяющихся значений ключа (комбинаций значений полей, составляющих этот ключ).

3.8. Перейти в *режим таблицы* и ввести в таблицу **Предметы** несколько повторяющихся значений, например: **П-005 ИНФОРМАТИКА П-005 ИНФОРМАТИКА**

3.9. Для таблицы **Экзамены** создать составной индекс *КонКосДат* и придать ему свойства ключа таблицы. В его состав должны войти поля *КодПредм*, *КодСтуд*, *ДатаЭкз*.

3.9.1. Для проверки правильности созданного ключа ввести в таблицу **Экзамены** данные о студенте, дважды сдававшем (пересдававшем) в один день один экзамен и получившем разные оценки. Убедитесь в невозможности ввести две различные по всему множеству полей записи с одинаковыми значениями составных ключей.

3.9.2. Ввести в таблицу **Экзамены** 5 корректных записей о студентах, упомянутых в таблице **Личности**.

4. Ввести в каждую из имеющихся таблиц по 5 новых записей с различными (допустимыми - 3 записи и недопустимыми - 2 записи) значениями уникальных составных ключей и убедиться в том, что правильно определенные ключи:

- не позволяют вводить логически неверные данные (например, о студенте, дважды сдававшем в один день один экзамен, что не допускается правилами сдачи экзаменов);
- позволяют ввести данные, о двух полных однофамильцах (тезках), родившихся в один день;
- не позволяет присвоить двум людям (хоть и имеющим одинаковые данные) один личный учетный код.

4.1. Проверить правильность работы всех таблиц базы данных, просмотреть их содержимое и сохранить все таблицы в базе данных **База№3**.

Примечание. Созданные в данной работе индексы, особенно составные, создавались с целью демонстрации возможности использования механизма индексации для обеспечения логической достоверности данных.

Основная цель использования индексов состоит в существенном ускорении поиска и выборки данных при использовании индексированных полей в запросах. В данном задании выборка не выполняется, поскольку на базе данных **База№3** в силу ее чрезвычайно малого объема невозможно заметить разницу в скорости выполнения выборки с использованием индексов и без их использования.

Приведенные варианты индексов служат демонстрационным целям, и в следующих заданиях в структуру ключевых полей и индексов будут внесены существенные

изменения, создающие новую систему ключевых полей, предназначенную в первую очередь для организации связей между таблицами с целью их совместного использования.

Задание 2. Связывание таблиц

1. Открыть базу данных *База№3*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

1.2. Удалить все индексы, созданные ранее для всех таблиц базы данных *База№3*. Отменить все признаки *первичных ключей* для всех таблиц базы данных.

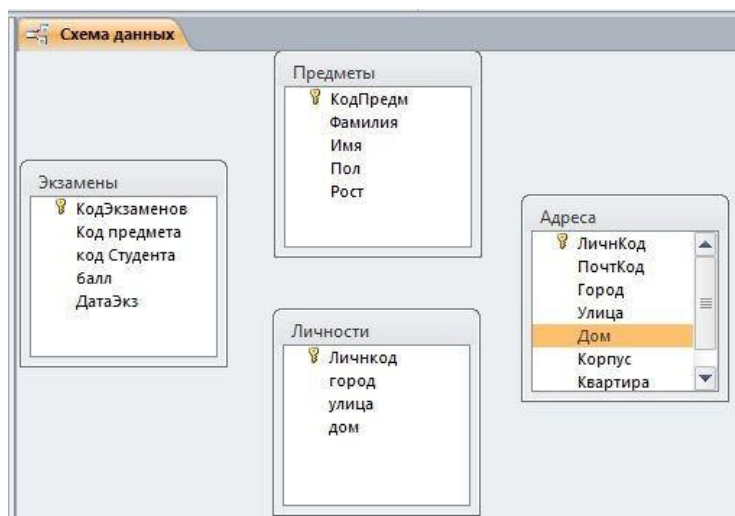
2. Создать новую систему первичных ключей для таблиц по следующему образцу

Таблица	Первичный ключ
<i>Личности</i>	<i>ЛичнКод</i>
<i>Адреса</i>	<i>ЛичнКод</i>
<i>Предметы</i>	<i>КодПредм</i>
<i>Экзамены</i>	<i>КодЭкзамена</i>

2.1. В тех полях таблиц, которые становятся ключевыми, удалить повторяющиеся записи.

2.3. В таблице *Экзамены* для поля *КодЭкзамена*, которое будет уникально определять каждую запись таблицы и использоваться как *первичный ключ*, установить тип поля - *Счетчик*.

2.4. Открыть окно *Схема данных* и разместить таблицы так, как показано на рисунке ниже:



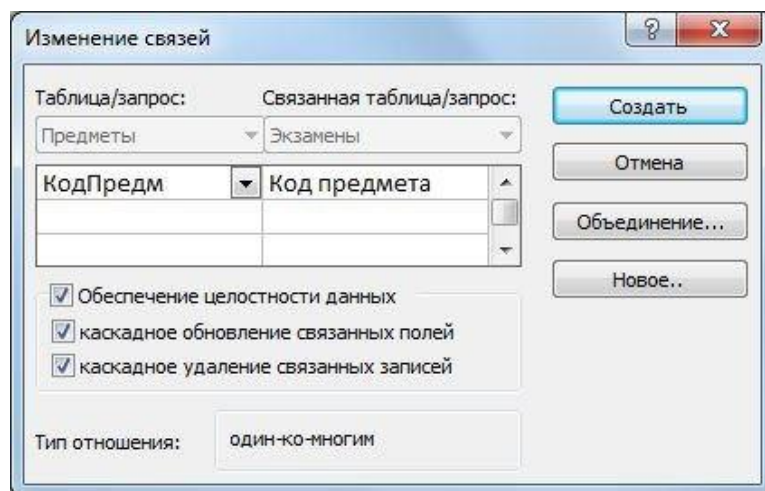
3. Установить связи между таблицами *Личности*, *Адреса*, *Предметы*, *Экзамены*.

3.1. Внимательно прочитать все подсказки, комментарии и пояснения, касающиеся связывания таблиц и параметров объединения.

3.2. Выбрав таблицу *Экзамены*, с помощью мыши “перетащить” поле *КодПредмета* на поле *КодПредм* таблицы *Предметы*, указав тем самым необходимость установления связи между таблицами по этим полям.

3.3. В открывшемся окне *Связи* определить тип и параметры связи между таблицами *Экзамены* и *Предметы*.

3.3.1. Между полями *Предметы.КодПредм* и *Экзамены.КодПредм* должна быть установлена связь типа *Один-ко-Многим*.



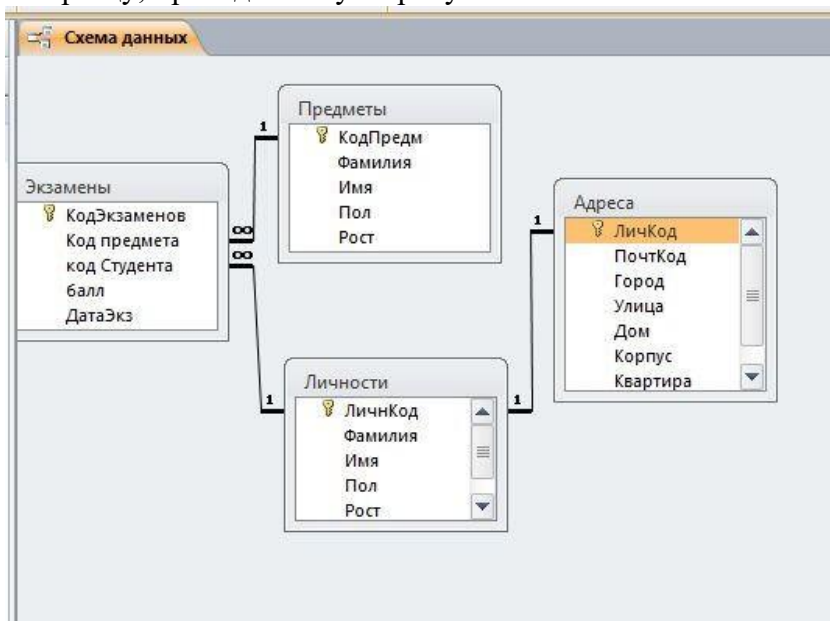
3.3.2. Установить режим *Обеспечения целостности данных: Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*. Для установки этих режимов необходимо активизировать соответствующие переключатели в группе *Обеспечение целостности* окна *Связи*.

3.3.3. Нажав кнопку *Объединение*, открыть окно *Параметры объединения* и выбрать такой режим объединения данных различных таблиц, при котором из связанных таблиц будут выбираться только те записи, для которых совпадают значения полей связи.

3.4. Установить связь между таблицами *Личности* и *Экзамены* через поля *КодСтудента* и *ЛичнКод* соответственно. Тип отношения между этими таблицами - *Один-ко-Многим*, где стороной *Много* выступает таблица *Экзамены*. Обеспечить целостность данных при обновлении и удалении записей. Установить объединение таблиц - по первому типу (когда из связанных таблиц выбираются только записи, для которых совпадают значения полей связи).

3.5. Установить связь типа *Один-к-Одному* между соответствующими полями таблиц *Личности* и *Адреса* с обеспечением целостности данных при обновлении и удалении записей.

3.6. Активизировать и просмотреть окно *Схема данных*. Его внешний вид должен соответствовать образцу, приведенному на рисунке ниже:



4. Используя подтаблицы просмотреть связанные данные для таблицы *Личности* (значок «+» слева от каждой записи в таблице).

5. Сохранить все таблицы и базу данных *База№3*.

Контрольные вопросы

1. Назначение индексов в базе данных?
2. Как создать составной индекс?
3. Назначение и особенности задания связей между таблицами.

Лабораторная работа №5

Поиск данных в таблице. Установка даты и вывод записей на экран

Цель работы: освоить приемы редактирования проекта БД и макета таблиц, поиска и сортировки данных в СУБД ACCESS; освоить принципы создания запросов выборки.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предьявляется реализованная средствами MicrosoftAccess база данных для заданной предметной области

Краткие теоретические сведения:

Использование Мастера подстановок при вводе данных в таблицы

Мастер подстановок позволяет формировать для нужного поля список значений, который может содержать данные другой таблицы или запроса, либо состоять из фиксированного набора значений. В обоих случаях Мастер подстановок облегчает ввод данных и обеспечивает их достоверность.

Например, в базе данных **Студенты-Экзамены** поле **Номерзачетки** является общим для обеих таблиц. При заполнении этого поля в неключевой таблице **Экзамены** важно, чтобы его значения для каждого студента точно соответствовали значениям в ключевой таблице. Для этого значения поля **Номер зачетки** в таблице **Экзамены** лучше всего представить в виде списка. Это избавит от необходимости печатать данные при заполнении этой таблицы, предотвратит появление ошибок при наборе и сделает ввод данных более наглядным (рис.4).

Комбинированный список для поля **Номер зачетки** формируется на основе данных связанной таблицы **Студенты**. Для создания комбинированного списка нужно войти в режим конструктора таблицы **Экзамены**, перейти в колонку **Тип данных** для поля **Номер зачетки**. Из списка доступных типов полей выбрать элемент **Мастер подстановок**.

Первое диалоговое окно Мастера подстановок предлагает выбрать источник формирования списка: на основе данных таблицы\запроса или фиксированный набор значений. В данном случае нужно выбрать первый вариант.

В следующем окне из приведенного списка таблиц\запросов следует выбрать таблицу\запрос, являющуюся источником данных для списка. В нашем примере такой таблицей является таблица **Студенты**, т.к. она послужит источником данных для списка номеров зачетов.

В третьем окне Мастера подстановок из списка **Доступные поля** нужно выбрать поля, значения которых фигурируют в списке. В данном случае можно выбрать поля **Фамилия** и **Имя**, которые сделают список более информативным. После этого можно

нажать кнопку **Готово** для завершения процесса проектирования комбинированного списка. Сохранив проект таблицы, можно открыть таблицу **Экзамены** и просмотреть результат работы.

■ Экзамены : таблица					
	Номер зачетк	Предмет	Итог	Оценка	Дата
	12522	История	Дифзачет	5	10.09.98
	12522	Боярино Андрейка	Дифзачет	3	06.09.98
	12578	Сидоров Петр	экзамен	3	09.09.98
	12579	Петров Иван	зачет	0	25.09.98
	12536	Каледин Сергей	зачет	0	24.09.98
	12578	химия	дифзачет	4	10.10.98

Рис.4 Использование комбинированного списка

Другая возможность использования Мастера подстановок - это создание списков с фиксированными значениями. Он используется в тех случаях, когда набор значений в поле строго ограничен. Например, поле **Факультет** в таблице **Студенты** может иметь ограниченный набор значений (все факультеты, имеющиеся в данном ВУЗе). Никаких других значений в этом поле быть не может. Чтобы ввести единообразие в записях и избежать ошибок при вводе, это поле также можно представить в виде списка, формируемого Мастером подстановок. Процесс создания списка похож на описанный выше, но на первом шаге Мастера подстановок следует выбрать опцию **Фиксированный набор значений**. На следующем шаге нужно создать список этих значений (в данном примере записать в список все факультеты данного ВУЗа). После чего можно завершить процесс создания списка и сохранить изменения в структуре. Теперь при вводе данных в поле **Факультет** нет необходимости печатать значения, а достаточно просто выбрать из предлагаемого списка нужное.

Порядок выполнения работы

Создать новую базу данных **База№4**.

Задание 1.

Для создания структуры ключевой таблицы **Студенты** рекомендуется использовать режим конструктора.

Внимание! В бланке **Свойства** обязательно указать длину текстовых полей, формат числовых полей и дат. Поле **Номер зачетки** в таблице **Студенты** объявить ключевым и индексированным со значением **Совпадения не допускаются**.

Структура таблицы **Студенты** должна быть следующей:

Имя поля	Тип поля
Номер зачетки	Числовой
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Факультет	Текстовый
Курс	Числовой
Группа	Числовой
Дата рождения	Дата\Время
Стипендия	Числовой

Заполнить таблицу содержимым (10-15 записей).

Структура таблицы **Экзамены** может быть следующей:

Имя поля	Тип поля
Номер зачетки	Мастер подстановок..

Предмет	Текстовый
Оценка	Числовой
Дата сдачи	Дата\Время

Обязательно определить нужные формат и длину полей в бланке **Свойства**.

Тип поля **Номер зачетки** определяется Мастером подстановок, используя для подстановки данные из таблицы **Студенты**. В качестве доступных при подстановке полей выбрать **Фамилию** и **Имя**.

Поле **Номер зачетки** в таблице **Экзамены** объявить индексированным со значением **Совпадения допускаются**.

Внимание!При сохранении структуры неключевой таблицы Access может предупредить об отсутствии ключевого поля и предложит создать это поле сейчас. В данном случае следует отказаться от этого.

После определения структур обеих таблиц вызвать окно **Схема данных** и добавить в схему данных обе таблицы (**Студенты** и **Экзамены**). Установить в окне схемы данных связь между таблицами по полю **Номер зачетки**. В окне **Связи** включить переключатель **Определение целостности данных**. После этого нужно указать тип связи: **Один-ко-многим** и включить опции **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**. После этого закрыть окно **Связи**.

В режиме таблицы ввести данные в таблицу **Экзамены**, используя созданный с помощью Мастера подстановок список в поле **Номер зачетки**.

Для проверки соблюдения целостности данных при работе с таблицами нужно:

***изменить значение ключевого поля (**Номер зачетки**) для одной из записей в таблице **Студенты**. Перейти в таблицу **Экзамены** и проверить, изменилось ли в ней значение общего поля для соответствующих записей;

***удалить одну из записей в таблице **Студенты**. Перейти в таблицу **Экзамены** и проверить, удалены ли в ней соответствующие записи.

Внимание!После редактирования таблицы **Студенты** нужно сначала сохранить в ней изменения, а затем переходить в неключевую таблицу.

Задание 2.

В режиме Конструктора таблицы **Студенты** произвести в ее структуре следующие изменения:

* добавить поля **Город**, **Адрес**, **Телефон**;

* определить тип поля **Факультет** с помощью Мастера подстановок, взяв в качестве источника данных фиксированный набор значений (список всех факультетов ВУЗа).

* для поля **Город** в свойстве **Значение по умолчанию** задать значение: **Уфа**.

* для поля **Курс** ввести условие на значение: **>0, <=5** и задать соответствующее сообщение об ошибке.

Данные в поля **Город**, **Адрес**, **Телефон** вводить в режиме таблицы.

Отредактировать значения в поле **Факультет**, используя список значений, созданный Мастером подстановок.

Для нескольких записей использовать значение по умолчанию в поле **Город**.

В одной из записей попробовать внести в поле **Курс** значение, большее 5.

Вызвать окно для изменения схемы данных. Скрыть одну из таблиц, включенных в схему данных (например, **Экзамены**). Затем отобразить все прямые связи.

Изменить макет таблицы **Студенты**:

- зафиксировать столбцы **Фамилия** и **Номер зачетки**.
- поле **Город** поставить после поля **Отчество**;
- скрыть столбцы **Адрес**, **Телефон** и **Стипендия**;
- оставить для столбцов только вертикальную сетку;

- установить произвольно цвет фона для записей;
- изменить шрифт для записей таблицы на курсив.

Отсортировать таблицу **Студенты** по следующим признакам:

- возрастанию в поле **Фамилия**;
- убыванию в поле **Стипендия**;
- возрастанию в поле **Факультет** и убыванию в полях **Курс** и **Группа**.

Найти в таблице **Студенты** все записи, удовлетворяющие следующим условиям:

- студенты, чьи фамилии начинаются с определенной буквы;
- студенты, обучающиеся на одном курсе определенного факультета.

Найти записи для студентов определенного факультета и заменить для них название этого факультета. Например, **эконом.на экономический**.

Задание 3.

Создать простой запрос - выбрать несколько произвольных полей из таблицы **Студенты**.

С помощью **Конструктора** создать запросы, удовлетворяющие условиям:

- единственное значение факультета;
- два различных факультета;
- фамилии студентов, начинающиеся с определенной буквы (использовать шаблоны);
- фамилии студентов, заканчивающиеся на "ов";
- фамилии студентов одного факультета и одного курса;
- фамилии и имена студентов, проживающие в одном из городов или обучающиеся на одном из факультетов;
- фамилии студентов, у которых стипендия больше 400 рублей;
- фамилии студентов, занимающиеся не в 1-ой группе и стипендия которых в пределах от 200 до 500 р.

Примечание:

В запрос должны быть включены поля *Фамилия, Имя, Отчество* и те поля, где вводятся критерии.

После того как был задан критерий для запроса, запрос нужно выполнить и сохранить под именем, подходящим по смыслу.

Для запросов с полем типа **Дата/время** добавить поле *Дата рождения* и выбрать записи, удовлетворяющие условиям:

- дата больше 1.1.80;
- дата в интервале значений и задан факультет;
- фамилии и имена студентов, родившихся в 80-х годах;
- вычислить возраст студентов;
- фамилии и имена студентов, родившихся в первой половине месяца;

Создать итоговый запрос:

- оставить в запросе поля *Факультет, Стипендия, Номер зачетки*, вычислить максимальное значение стипендии для каждого факультета и подсчитать количество студентов на каждом факультете (используя Count).

Запрос с вычисляемыми полями:

- включить в запрос вычисляемое поле, которое является результатом сцепления текстовых полей *Фамилия, Имя, Отчество*. Назвать поле *Ф. И. О. студента*.
- используя построитель выражений, подсчитать надбавку студентам, равную 15% от стипендии;

Создать запрос, в котором используются поля из двух ранее созданных и связанных таблиц, задав ему имя **Запрос для 2-х таблиц**

—убрать несколько полей таблицы **Студенты** и добавить поля *Предмет* и *Оценка* из таблицы **Успеваемость**;

–выбрать поле *Фамилия, предмет и Оценка*, вычислить минимальное значение по полю *Оценка*;

–сгруппировать по номеру зачетки и вычислить среднюю оценку для каждого студента.

Контрольные вопросы

1. Как отсортировать записи по одному полю? по нескольким полям?
2. Как произвести поиск и замену данных в Access?
3. Как используется Мастер подстановок?
4. Что такое запрос ?
5. Что такое бланк QBE ?
6. Где записываются критерии условия выбора для запроса?
7. Перечислите основные операторы, используемые в запросе.
8. В чем различие между операторами OR и AND ?
9. Назначение итоговых запросов.
10. Назначение построителя выражений.
11. Как вычислить сумму значений заданного поля?
12. Как осуществить сортировку записей в запросе?

Лабораторная работа №6

Установление взаимосвязей между таблицами.

Цель работы: знакомство с основными понятиями присоединения и импорта данных. Создание учебной базы данных. Знакомство с основными приемами работы по установке связей между файлом БД и файлами электронных таблиц, текстовых файлов.

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предьявляется реализованная средствами MicrosoftAccess база данных для заданной предметной области

Краткие теоретические сведения:

Импорт и присоединение таблиц из текстовых файлов

Access позволяет импортировать в базы данных табличные данные из обычных текстовых файлов. Эти файлы могут быть созданы либо одним из текстовых редакторов (например, приложением "Блокнот" в Windows), либо путем экспорта из другого приложения. Такая возможность особенно удобна в тех случаях, когда вам необходимы данные из приложения, которое не позволяет экспортировать данные из внутреннего представления в форматы DBF, WKS или XLS.

Существуют два основных формата текстовых файлов, в которых содержатся табличные данные:

- с полями фиксированной ширины — строки состоят из полей фиксиро-

ванной ширины, пустые позиции при этом заполняются пробелами;

5. с полями переменной ширины — со специальными символами — разделителями полей.

Для файлов с фиксированной шириной полей начало каждого поля определяется по его позиции относительно начала строки. Строки имеют одинаковую длину и разделяются символами новой строки (обычно это пара символов — возврат каретки и перевод строки).

Для файлов с разделителями полей существует несколько стандартных вариантов разделения полей. Они приведены в табл. 3.4.

Таблица 1.1. Форматы текстовых файлов, поддерживаемые Access

Формат	Описание
Записи в одну строку, поля разделены запятыми	Разделителем записей является символ новой строки. Некоторые приложения включают все значения полей в двойные кавычки, другие приложения ограничиваются заключением в кавычки лишь текстовых значений, чтобы отличить их от числовых
Записи в одну строку, поля разделены знаками табуляции	Разделителем записей является символ новой строки. Предполагается, что все значения полей текстовые, а сами поля разделены знаками табуляции. Этот формат поддерживается большей частью текстовых процессоров
Записи в одну строку, поля разделены пробелами	Access позволяет использовать пробел в качестве символа разделителя полей, однако использование такого разделителя очень неудобно для файлов, содержащих текстовые данные (в которые также может входить пробел)

Процесс импорта данных, хранящихся в текстовом файле, выполняется с помощью Мастера импорта текстов. Выберите команду **Файл, Внешние данные, Импорт** и следуйте указаниям Мастера. Если файл содержит разделители полей, то укажите символ разделителя на втором шаге Мастера импорта текстов или выберите переключатель **фиксированная ширина полей**. Если в первой строке импортируемого файла содержатся заголовки полей, установите флажок **Первая строка содержит имена полей**

Порядок выполнения работы

Задание 1. Создайте следующие таблицы. Таблица Покупатели – текстовый файл **таблица1.txt** (в качестве разделителя можно выбрать знак табуляции), таблица Заказы – электронная таблица **Заказы**. Импортируйте созданные таблицы в базу данных **База5** создайте в ней таблицу **Продавцы**. Задайте ключевые поля (если они не заданы), установите связи, обеспечивающие целостность данных.

Таблица 2.1 Продавцы

Номер	Имя	Город	Комиссионные
1001	Дугинов	Липецк	0,12
1002	Ольшевский	Саратов	0,13
1004	Помякушина	Липецк	0,11
1007	Шикин	Бронницы	0,15
1003	Борескова	Новгород	0,1

Таблица 2.2 Покупатели

Номер	Имя	Город	Рейтинг	Номер_продавца
2001	Голубев	Липецк	100	1001
2002	Дмитриева	Архангельск	200	1003
2003	Кустов	Саратов	200	1002
2004	Кузьминов	Подольск	300	1002
2006	Кречко	Липецк	100	1001

2008	Фролов	Саратов	300	1007
2007	Каратыгин	Архангельск	100	1004

Таблица 2.3 Заказы

Номер	Стоимость	Дата_покупки	Номер_покупателя	Номер_продавца
3001	18,69	10/03/1996	2008	1007
3003	767,19	10/03/1996	2001	1001
3002	1900,10	10/03/1996	2007	1004
3005	5160,45	10/03/1996	2003	1002
3006	1098,16	10/03/1996	2008	1007
3009	1713,23	10/04/1996	2002	1003
3007	75,75	10/04/1996	2004	1002
3008	4723,00	10/05/1996	2006	1001
3010	1309,95	10/06/1996	2004	1002
3011	9891,88	10/06/1996	2006	1001

Задание 2.

Создайте еще три произвольных таблицы в Access, Excel и текстовом файле и присоедините их к существующей базе используя связь с таблицами.

Контрольные вопросы:

1. Особенности форматов текстовых файлов при присоединении.
2. Особенности присоединения и импорта таблиц баз данных Access.
3. Что будет отображено в таблицах после присоединения электронной таблица содержащей формулы.
4. Как происходит определение типов полей импортируемой таблицы.

Лабораторная работа №7

Использование элементов программирования для работы с базами данных

Цель работы: знакомство с созданием простых макросов и их использованием для автоматизации работы базы данных. Использование групп макросов и макросов с условиями для автоматизации работы базы данных. Научиться создавать автоформы и автоотчеты. Знакомство с создание кода VBA

Содержание отчета:

1. Цель работы
2. Этапы создания таблиц со скриншотами страниц
3. Этапы создания схемы данных со скриншотом схемы данных.
4. Выводы

Также, с отчетом предъявляется реализованная средствами MicrosoftAccess база данных для заданной предметной области

Краткие теоретические сведения:

Что такое Visual Basic for Applications.

Visual Basic for Applications представляет собой полуфункциональный язык программирования, являющийся неотъемлемой составной частью Access. Этот язык используется для разработки приложений, предназначенных для манипулирования БД и

для настройки пользовательского интерфейса. VBA - это структурированный язык программирования высокого уровня. В нем, как и в других языках, есть операторы проверки условий, циклического выполнения повторяющихся операций, а также обмена данными с памятью и дисками. В языке VBA реализованы общие принципы объектно-ориентированного программирования. Это означает, что пользовательская среда, управляемая приложением, не подвергается изменениям путем выполнения последовательности процедур и операторов, но реагирует на события, связанные с различными объектами: полями ввода, кнопками, разделами форм и отчетов. В языке VBA программный код привязан непосредственно к объектам и срабатывает тогда, когда случается определенное событие. Все программирование в Windows основано именно на отклике на то или иное событие в системе.

Кроме того для автоматизации работы с базой данных используется система встроенных макрокоманд.

В **MICROSOFT ACCESS** имеется свыше 40 макрокоманд, которые вы можете включать в макросы. Макрокоманды выполняют такие действия, как открытие таблиц и форм, выполнение запросов, запуск других макросов, выбор опций из меню, изменение размеров открытых окон и т.п.

Для выполнения, создания или редактирования макроса щелкните по корешку **Макрос**, находясь в окне вашей базы данных. В появившемся меню снизу вы увидите список уже созданных макросов, сверху – кнопки: **Создать** (для создания нового макроса), **Выполнить** (для выполнения действия в макросе, выбранном в окне базы данных, причем при запуске макроса выполняется его первая макрокоманда), **Конструктор** (для создания и изменения макросов).

Создание простого макроса

В столбце **Макрокоманда** вы можете задать одну из предоставляемых макрокоманд, в столбце **Примечание** можете писать комментарии. При выборе макрокоманды в правой нижней части экрана появятся аргументы макрокоманды, которые вы можете установить нужным вам образом. Перед запуском макроса его следует сохранить. Для проверки работы макроса рекомендуется выполнить команды макроса в пошаговом режиме, для этого в окне создания или редактирования макроса щелкните по кнопке **По шагам** на панели инструментов, либо выполните команду **Макрос>По шагам**.

Порядок выполнения работы

Задание 1. Создание автоформ, автоотчетов, макросов.

1. Создать базу данных *База№6*, которая содержит следующие таблицы:

1.1. **Студенты** (поля: *КодСтуд*, *ФамСтуд*, *ИмяСтуд*, *ПолСтуд*, *РождСтуд*).

1.2. **Родители** (поля: *КодСтуд*, *ФамРод*, *ИмяРод*, *ОтчРод*, *ПолРод*, *ТелРод*).

1.3. **Оценки** (поля: *КодСтуд*, *БаллХим*, *БаллИнф*, *БаллМат*, *Семестр*).

Заполнить таблицы экспериментальным содержанием (5-7 записей).

Создайте для каждой таблицы форму, а для таблиц *Студенты* и *Оценки* – отчеты.

1. Выбрать таблицу и выбрать кнопку **Форма** на вкладке **Создание** стандартной панели инструментов. Для таблицы будет создана форма по умолчанию.

2. Сохранить созданную форму, выполнив щелчок на кнопке закрытия окна, а затем на запрос о сохранении ответить утвердительно и ввести имя созданной формы или оставить предложенное для сохранения имя формы.

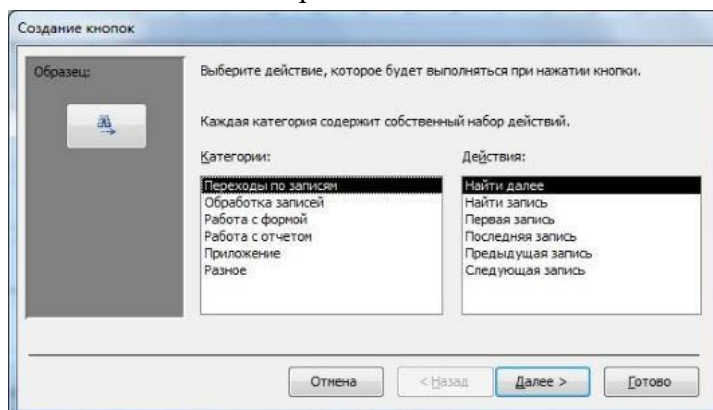
3. Используя команду **Отчет** во вкладке **Создание** создать указанные отчеты.

Создать в одной из созданных Форм кнопки:

1. Выбрать созданную форму.

2. Перейти в режим конструктора формы.

3. Выполнить щелчок на элементе Кнопка.
4. Выполнить щелчок на свободном месте формы, чтобы вставить в нее кнопку. Откроется диалоговое окно мастера кнопок:



5. В списке Категории выбрать пункт Переходы по записям, а затем в списке Действия -> Найти запись. После щелчка на кнопке Далее в будет предложен выбор – добавления кнопки или текста на форму.
6. Выполнить щелчок на элементе Кнопка.
7. Вставить кнопку на свободном месте формы ниже предыдущей, созданной ранее. Снова откроется окно кнопок.
8. В списке Категории выбрать пункт Работа с формой, а в списке Действия - пункт Закрытие формы. После щелчка по кнопке Готово в форму будет добавлена кнопка, выполняющая закрытие формы.
9. Сохранить форму. После этого можно запустить форму и проверить работу добавленных кнопок.
10. Переключиться в режим Конструктора, выполнив щелчок по кнопке Конструктор.
11. Выбрать команду  Просмотреть код или выполнить щелчок по кнопке Программа на панели инструментов. Откроется окно редактора VisualBasicforApplications, представленное на рисунке :

```
Option Compare Database
Option Explicit

Private Sub Кнопка13_Click()
On Error GoTo Err_Кнопка13_Click

Screen.PreviousControl.SetFocus
DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70

Exit_Кнопка13_Click:
Exit Sub

Err_Кнопка13_Click:
MsgBox Err.Description
Resume Exit_Кнопка13_Click

End Sub

Private Sub Кнопка14_Click()
On Error GoTo Err_Кнопка14_Click

DoCmd.Close

Exit_Кнопка14_Click:
Exit Sub
```

В этом окне можно видеть три фрагмента программы на языке VBA: раздел объявлений (в верхней части окна), который содержит код, относящийся ко всей форме в целом, а также два раздела кода, привязанных к двум кнопкам.

Первый раздел начинается с заголовка процедуры:

PrivateSub Кнопка13_Click()

Заканчивается процедура оператором: EndSub. Операторы, заключенные между ними, выполняются после щелчка по кнопке Найти. Основная часть работы выполняется следующим оператором:

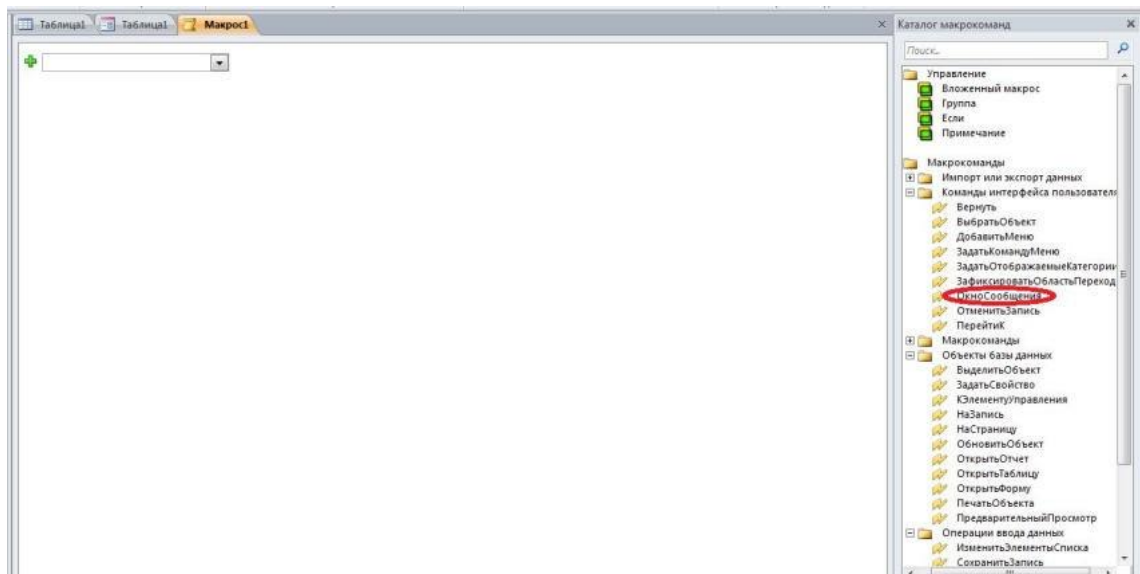
DoCmd.DoMenuItemacFormBar, AcEditMenu, 10, AcMenuVer70

Этот оператор выполняет те же действия, что и команда меню Найти, т.е. выводит на экран диалоговое окно Поиск в поле, с помощью которого можно выполнить поиск.

Создать три запроса: студенты женского пола; студенты, имеющие 2 по информатике; родители, имеющие двух детей – студентов с помощью кнопки Макрос во вкладке Создание.

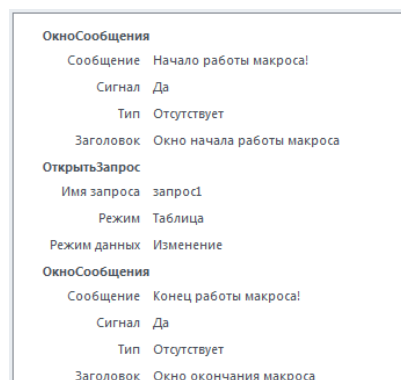
2. Создать макрос *Макрос1*, обеспечивающий выполнение следующих действий:

- выдачу сообщения “НАЧАЛО РАБОТЫ МАКРОСА”. Окно, в котором выводится текст, озаглавить как “Окно начала макроса”;



- выполнение запроса *Запрос1*, созданного в Задании 1. В качестве первого аргумента - *Имени запроса* - выбрать *Запрос 1*. В качестве режима просмотра выбрать режим *Таблица*;
- выдачу сообщения “КОНЕЦ РАБОТЫ МАКРОСА”. Окно, в котором выводится текст, озаглавить как “Окно окончания макроса”.

Примечание. Для команды *Сообщение* обязательными параметрами, задаваемыми в нижней части диалогового окна (рис. 1) являются аргумент *Сообщение*, в окне которого записывается выводимый текст и аргумент *Заголовок*, в окне которого записывается текст заголовка окна.



3. Выполнить макрос *Макрос1*. Убедиться в правильности выполнения.

4. Установить с помощью кнопки на панели макросов режим пошагового выполнения и, выполнив макрос по шагам, пронаблюдать сообщения, выдаваемые на каждом шаге.

5. Создать макрос *Макрос2*.

5.1. Добавить в макрос *Макрос1*, после второй команды еще две команды:

- для выдачи информационного сообщения *ПАУЗА* после появления таблицы с результатом запроса;
- для закрытия окна с результатом выполнения запроса.

5.2. Сохранить макрос под новым именем *Макрос2*.

6. Выполнить “Макрос2”. Убедиться в правильности выполнения.

7. Создать макросы *Родители* и *Оценки*, обеспечивающие открытие соответствующих форм.

- a. На вкладке Создание выбрать команду Макросы.
- b. Перетащить из вкладки *Формы* окна базы данных форму *Родители* в область Добавления нового макроса и проверить заполнение аргументов макрокоманды.
- c. Сохранить макрос под именем *Родители*.
- d. Проверить работоспособность макроса.
- e. Повторить п. а. – d. для формы *Оценки*.
- f. Создать командные кнопки с надписями “Родители” и “Оценки”, к которым присоединить соответствующие макросы.
- g. Проверить работоспособность кнопок.

8. Сохранить базу данных на диске для дальнейшего использования.

Задание 2. Макросы с условиями

Для базы данных *База№6*, создать макрос *МакАдр_1*, который выполняет следующие действия:

- выдает информационное сообщение о начале работы;
- открывает форму *Студенты*;
- выдает об этом информационное сообщение и приостанавливает работу до нажатия *Кнопки*;
- открывает отчет *Оценки*;
- выдает информационное сообщение об окончании работы.

1.1. Дать этому макросу имя *МакАдр_1* и сохранить его.

- Выполнить макрос *МакАдр_1* несколько раз и попытаться проследить последовательность действий. Обратит внимание на то, что форма *Студенты* и отчет *Оценки* остаются открытыми после работы макроса.

2. Создать макрос *МакАдр_2*, выполняющий следующие действия:

- выдачу сообщения о начале работы;
- открытие формы *Родители*;
- выдачу *информационного* сообщения о предстоящем закрытии формы *Родители*;
- закрытие формы *Родители*;
- открытие отчета *Оценки*;
- выдачу *предупреждающего* сообщения о предстоящем закрытии отчета;
- закрытие отчета *Математика*.

2.1. Дать этому макросу имя *МакАдр_2* и сохранить его.

2.2. Выполнить макрос *МакАдр_2* в пошаговом и в автоматическом режимах.

Контрольные вопросы:

1. Определение, назначение макросов.
2. Последовательность создания макроса.
3. Отличие макросов с командами безусловного и условного переходов.
4. Как создать код VBA с помощью мастеров элементов управления.

Практическая работа № 1

Тема: «Нормативно-правовые документы и стандарты в области защиты информации и информационной безопасности»

Цель работы: Получить практические навыки работы со стандартами в области защиты информации и информационной безопасности.

Ход работы:

1. Найти в сети Internet серию стандартов, регламентирующих информационные технологии. Номер серии записать в тетрадь.
2. Выписать в тетрадь 10 стандартов и их реквизиты (номер, название, дата введения в действие, кем издан)
3. В виде таблицы или схемы выписать из стандарта стадии и этапы создания АС.
4. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется в соответствии с ходом работы)

Практическая работа № 2

Тема: «Системы менеджмента качества»

Цель работы: Ознакомиться и изучить содержания международных стандартов в области ИТ: ISO/IEC 9126, ISO/IEC 14598 и ИСО/МЭК 9126-1
Получить практические навыки работы с данными стандартами.

Ход работы:

1. Найти в сети Internet серию стандартов, регламентирующих информационные технологии. Номер серии записать в тетрадь.
2. Найти определения следующих понятий;
 - качество;
 - надежность;
 - совместимость;
 - взаимозаменяемость;

- безопасность;
- соответствие;
- дефект.

Для сравнения дайте определения этим же понятиям, принятым в отечественной практике и по результатам работы заполните таблицу.

Пример таблицы

№ п/п	Понятие	Определения международных стандартов	Определения, принятые в отечественной практике

3. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется, в соответствии с ходом работы)

Практическая работа № 3

Тема: «Стандарты и спецификации в области информационной безопасности»

Цель работы: Изучить международные и национальные стандарты и спецификации в области ИБ — от "Оранжевой книги" до ISO 15408.

Получить навыки определения сильных и слабых стороны этих документов.

Ход работы:

1. Ознакомиться со стандартами и спецификациями в области информационной безопасности.

2. Разработать интерфейс пользователя «Оранжевая книга» как оценочный стандарт».

3. Ответить на контрольные вопросы.

Контрольные вопросы:

1. Уровень безопасности C, согласно "Оранжевой книге", характеризуется:
 - произвольным управлением доступом
 - принудительным управлением доступом
 - верифицируемой безопасностью

2. Согласно рекомендациям X.800, аутентификация может быть реализована на:

- сетевом уровне
- транспортном уровне
- прикладном уровне

3. "Общие критерии" содержат следующие виды требований:

- функциональные
- доверия безопасности
- экономической целесообразности

4. Уровень безопасности В, согласно "Оранжевой книге", характеризуется:

- произвольным управлением доступом
- принудительным управлением доступом
- верифицируемой безопасностью

5. Согласно рекомендациям X.800, целостность с восстановлением может быть реализована на:

- сетевом уровне
- транспортном уровне
- прикладном уровне

6. В число классов функциональных требований "Общих критериев" входят:

- анонимность
- приватность
- связь

7. Уровень безопасности А, согласно "Оранжевой книге", характеризуется:

- произвольным управлением доступом
- принудительным управлением доступом
- верифицируемой безопасностью

8. Согласно рекомендациям X.800, неотказуемость может быть реализована на:

- сетевом уровне
- транспортном уровне
- прикладном уровне

9. В число классов требований доверия безопасности "Общих критериев"

входят:

- разработка
- оценка профиля защиты
- сертификация

Практическая работа № 4

Тема: «Основные виды технической и технологической документации».

Цель работы: Познакомиться с технической и технологической документацией и получить практические навыки составления маршрутной технологической карты.

Ход работы:

1. Записать в тетрадь определение технологической документации.
2. Найти в сети Internet и изучить виды технологической документации. Записать их в тетрадь с кратким описанием.
3. Составить маршрутную технологическую карту на процесс производства компьютера.
4. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется обучающимся, в соответствии с ходом работы)

Практическая работа № 5

Тема: «Изготовление и оформление технической документации. Экспертиза ценности научно-технических документов».

Цель работы: Познакомиться с правилами изготовления и оформления технической документации.

Ход работы:

1. Найти в сети Internet и ознакомиться с правилами изготовления и оформления технической документации. Записать их в тетрадь с кратким описанием.
2. Выписать, как создаётся экспертная комиссия по работе с научно-техническими

документами.

3. Сделать вывод о проделанной работе.

Отчёт о работе:

(отчёт составляется обучающимся, в соответствии с ходом работы)

ОП.10 Численные методы

Практическая работа №1.

Тема: Вычисление погрешностей результатов арифметических действий над приближенными числами.

Приближение числа. Погрешности приближённых значений чисел.

Пусть x_0 – некоторая величина, истинное значение которой неизвестно и равно x . Число x_0 , которое можно принять за значение величины x , мы и будем называть приближенным числом.

Определение: Абсолютной погрешностью e_x приближенного значения числа X называется модуль разности между точным числом X его приближенным значением x , т.е. $e_x = |X - x|$.

Определение: Число x называется приближённым значением точного числа X с точностью до Δx , если абсолютная погрешность приближённого значения a не превышает Δx , т.е. $|X - x| \leq \Delta x$ (1).

Определение: Число Δx называется границей абсолютной погрешности приближённого значения числа x . $\Delta x = x - x_0 \Rightarrow x = x_0 + \Delta x$

Число Δx на практике стараются подобрать как можно меньше и простое по записи.

Из неравенства (1) найдём границы, в которых заключено точное значение числа X :

$$x - \Delta x \leq X \leq x + \Delta x.$$

$НГ_x = x - \Delta x$ - нижняя граница приближения величины X . $ВГ_x = x + \Delta x$ - верхняя граница приближения величины X .

Определение: Относительной погрешностью δx приближенного числа x числа X называется отношение абсолютной погрешности Δx этого приближения к числу x , т.е.

$$\delta x = \frac{\Delta x}{|x|}$$

Если первая значащая цифра в относительной погрешности δx меньше 5, то граница относительной погрешности определяется из неравенства $\delta x \leq \frac{1}{2} \cdot 10^{-n}$, где n - количество верных цифр.

Определение: Число x_0 называется приближенным по недостатку, если оно меньше истинного значения ($x_0 < x$), и по избытку, если оно больше истинного значения ($x_0 > x$).

Например, число 3,14 является приближенным значением числа π по недостатку, а 2,72 – приближенным значением числа e по избытку.

Пример: $\pi = 3,14$ $x = 3,1416$; $x_0 = 3,14$ $\Delta x = 0,0016$

$$\delta x = 0,00051 \quad \delta x = 0,051 \%$$

Относительная погрешность выражается в %.

Действия над приближенными числами.

Погрешность результата может быть выражена через погрешности первоначальных данных при помощи следующих теорем:

Теорема 1. Абсолютная погрешность алгебраической суммы равна сумме алгебраической погрешности слагаемых.

Теорема 2. Относительная погрешность алгебраической суммы заключена между наибольшим и наименьшим с относительной погрешностью слагаемых.

Теорема 3. Относительная погрешность произведения или частного равна сумме или разности относительных погрешностей соответственно делимого и делителя.

Абсолютная погрешность:

$$A + B = A_1 + B_1 + \Delta A + \Delta B \Rightarrow \text{СУММА}$$

$$\Delta(A + B) = \Delta A + \Delta B$$

$$A \cdot B = A_0 B_0 + A_0 \Delta B + B_0 \Delta A + \Delta A \Delta B$$

2) $\Rightarrow$ УМНОЖЕНИЕ

$$\Delta(AB) = A_0 \Delta B + B_0 \Delta A$$

$$3) \frac{A}{B} = \frac{A_0 + \Delta A}{B_0 + \Delta B} \Rightarrow \text{ЧАСТНОЕ}$$

$$\frac{\Delta A}{B} = \frac{\Delta A}{B_0} - \frac{A_0 \Delta B}{B_0^2}$$

Относительная погрешность:

$$1. \delta(A+B) = \frac{\delta A + \delta B}{A_0 + B_0}$$

$$2. \delta AB = \frac{\delta A}{A_0} + \frac{\delta B}{B_0} = \delta A + \delta B$$

$$3. \delta \frac{A}{B} = \frac{\delta A}{A_0} - \frac{\delta B}{B_0} = \delta A - \delta B$$

Пример.

1) Сложить приближенные числа 117,55 и 24,71

$$A+B=A_0+B_0+\Delta A+\Delta B$$

$$A=117,55 B=24,71$$

$$A_0=117 \quad \Delta A=0,55$$

$$B_0=24 \quad \Delta B=0,71$$

$$A+B=117+24+0,55+0,71=141+1,26=142,26$$

$$2) A \cdot B = (A_0 + \Delta A)(B_0 + \Delta B) = A_0 B_0 + A_0 \Delta B + \Delta A B_0 + \Delta A \Delta B$$

$$4,26 \text{ и } 7,12$$

$$\Rightarrow 4 \cdot 7 + 4 \cdot 0,12 + 7 \cdot 0,26 + 0,26 \cdot 0,12 = 28 + 0,48 + 1,82 = 30,33$$

18,88; 2,40 и 17,50 – сложить

$$A+B+C+D+E=A_0+B_0+C_0+D_0+E_0+\Delta A+\Delta B+\Delta C+\Delta D+\Delta E=117+24+18+2+17+0,55+0,71+0,88+0,4+0,5=141+20=17+3,04=181,04$$

$$4) -0,906-0,330=1,23$$

$$5) 1,84 \cdot 0,672 = 1 \cdot 0 + 1 \cdot (+0,672) + 0,84 \cdot 0 + 0,84 \cdot (+0,672) = 0 + (+0,672) + 0 + (+0,564) = +1,236$$

$$A \cdot B = 8,46 \cdot 5,3 = 8 \cdot 5 + 8 \cdot 0,3 + 0,46 \cdot 5 + 0,46 \cdot 0,3 = 40 + 2,4 + 2,3 + 0,138 = 44,8$$

$$6) 5,73; 4,104; 2,60; 1,894$$

$$A+B+C+D=A_0+B_0+C_0+D_0+\Delta A+\Delta B+\Delta C+\Delta D=5+4+2+1+0,73+0,104+0,6+0,894=12+1,598=14,33$$

$$7) 3,5 \cdot 51,2 + 8,25 \cdot 12,7$$

Пример.

$$1) \text{ Умножить } 3,5 \cdot 51,2 \Rightarrow A_0=3 B_0=51; \Delta A=0,5 \Delta B=0,2.$$

$$3 \cdot 51 + 3 \cdot 0,2 + 51 \cdot 0,5 + 0,5 \cdot 0,2 = 153 + 0,6 + 25,5 + 0,1 = 179,2$$

$$2) 8,25 \cdot 12,7 \Rightarrow A_0=8 B_0=12; \Delta A=0,25 \Delta B=0,7$$

$$8 \cdot 12 + 8 \cdot 0,7 + 12 \cdot 0,25 + 0,25 \cdot 0,7 = 96 + 5,6 + 3 + 0,175 = 104,77$$

$$A_0=179 \Delta A=0,2 B_0=104 \Delta B=0,77$$

$$A+B=283,97$$

$$8) (19,55+1,87) \cdot 0,42$$

$$A+B=19+1+0,55+0,87=21,42$$

$$A \cdot B = 21 \cdot 0 + 21 \cdot 0,42 + 0,42 + 0,42 \cdot 0 = 8,82 + 0,17 = 8,9$$

Значащие цифры. С помощью абсолютной погрешности определяется так называемое верное значение цифры.

Опр. Значащая цифра приближенного значения числа x , находящейся в разряде, в котором выполняется условие: абсолютная погрешность Δx не превосходит половину единицы этого разряда, называется *верной*.

Опр. Значащие цифры разрядов, где данное условие не выполняется, называется сомнительными. Все значащие цифры, расположены слева от верной, также будут верными и справа от сомнительной будут сомнительными.

Пример: 1) Для приближенного числа $x = 72,356$ известна абсолютная погрешность $\Delta x = 0,04$. Определить погрешность верно значащей цифры.

7	$\frac{10}{2} = 5 \geq 0,04$	верное число
2	$\frac{1}{2} = 0,5 \geq 0,04$	
3	$\frac{0,1}{2} = 0,05 \geq 0,04$	
5	$\frac{0,01}{2} = 0,005 <$	сомнительное число

$$\Rightarrow x \approx 72,4$$

2) $x = 3,73$; $\Delta x = 0,056$ $x \approx 3,7$

3	$\frac{1}{2} = 0,5 \geq 0,056$ – верное
7	$\frac{0,1}{2} = 0,05 \leq 0,056$ – неверное

Задания.

Определить абсолютную погрешность и указать верные цифры:

- 1) $1,936 \pm 12,29$;
- 2) $3,73 \pm 0,056$;
- 3) $3,672 \pm 0,0008$;
- 4) $4,732 \pm 0,06$;
- 5) 561274 ± 500 ;

Практическая работа № 2.

Тема: Решение алгебраических и трансцендентных уравнений методом половинного деления и методом итераций.

Определение. Трансцендентная функция – аналитическая функция, не являющаяся алгебраической. Простейшими примерами трансцендентных функций служат показательная функция, тригонометрические функции, логарифмическая функция, степенная функция.

Общая постановка задачи. Найти корни уравнения $f(x)=0$, где $f(x)$ -это алгебраическая или трансцендентная функция. Решение проводится в следующей последовательности:

- 1) отделение (локализация) корня;
- 2) приближенное вычисления корня до заданной точности.

Отделение действительного корня уравнения $f(x)=0$ – это нахождения отрезка $[a,b]$, которому принадлежит этот корень.

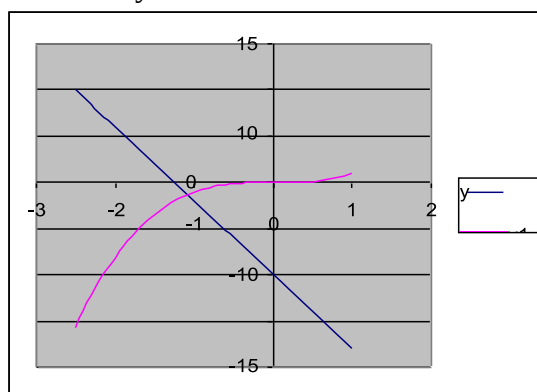
Определение. Отрезок $[a,b]$ называется отрезком локализации корня или изоляцией. Наиболее удобным методом определения отрезка является *графический* метод: строится функция $f(x)$, абсциссы точек пересечения графика функции с осью ОХ есть корни уравнения $f(x)=0$. Если функция $f(x)$ -сложная, то ее можно представить в следующем виде $f(x)=\varphi_1(x)-\varphi_2(x)$, так как $f(x)=0$, то $\varphi_1(x)=\varphi_2(x)$. Строим графики функций $y=\varphi_1(x)$ и $y=\varphi_2(x)$, абсциссы точек пересечения графиков этих функций будут корнями уравнения $f(x)=0$.

Пример. Найти корни уравнения:

$$y = x^3 + 8x + 10x^3 + 8x + 10 = 0$$

$$y_1 = x^3$$

$$y_2 = -8x - 10$$



Абсцисса точки пересечения графика функций $y_1 = x^3$ и $y_2 = -8x - 10$

$$x \in [-2; -1].$$

Уточнение корня производится следующими методами:

- 1) метод половинного деления (бисекций);
- 2) метод итераций;
- 3) метод хорд(секущих);
- 4) метод касательных (Ньютона);
- 5) комбинированный метод.

Метод половинного деления.

При решении уравнения методом половинного деления задается интервал $[a, b]$, на котором существует только одно решение и желаемая точность ε .

Такой метод можно применить, если функция $y=f(x)$ непрерывна на отрезке

$[a, b]$

и принимает разные значения на концах этого отрезка, то есть выполняется условие:

$$f(a) \cdot f(b) < 0$$

(*)

Разделим отрезок $[a, b]$ пополам точкой

$$C_1 = \frac{a+b}{2} \text{ и получим два отрезка } [a, c_1] \text{ и } [c_1, b].$$

Из этих двух отрезков выбираем тот, для которого выполняется условие (*). Пусть это отрезок $[c_1, b]$ и пусть для него выполняется условие:

$$f(c_1) \cdot f(b) < 0$$

Снова делим отрезок $[c_1, b]$ пополам, находим

$$C_2 = \frac{C_1 + b}{2}. \text{ И продолжаем до тех}$$

пор, пока не достигнется заданная точность ε , то есть пока не выполнится условие $|c_{i+1} - c_i| < \varepsilon$.

Пример: Найти корни уравнения $y = x^3 + 8x + 10$, отрезок изоляции $[-2; -1]$. Проверим условие (1):

$$f(-2) = (-2)^3 + 8(-2) + 10 = -8 - 16 + 10 = -14$$

$$f(-1) = (-1)^3 + 8(-1) + 10 = -1 - 8 + 10 = 1$$

$$f(-2) \cdot f(-1) = -14 \cdot 1 = -14 < 0$$

$$C_1 = \frac{-2 + (-1)}{2} = -1,5 \Rightarrow [-2; -1,5] \quad \text{и} \quad [-1,5; -1]$$

Проверим условие (*) для отрезка $[-1,5; -1]$

$$f(-1,5) = (-1,5)^3 + 8(-1,5) + 10 < 0, \quad f(-1,5) \cdot f(-1) < 0$$

$$[-1,5; -1]: \quad C_2 = \frac{-1,5 + (-1)}{2} = -1,25$$

$$\varepsilon = 0,001$$

$$|-1,25 + 1,5| = 0,25 > 0,001 \Rightarrow \text{условие не выполняется} \quad C_{10} = -1,0889$$

$$C_{11} = -1,0884$$

$$|-1,0884 + 1,0889| = 0,0005 < 0,001 \Rightarrow \text{заданная точность достигнута и корень уравнения}$$

$$x = -1,0884 \approx -1,088$$

Метод итераций.

При решении нелинейных уравнений методом итераций запишем уравнение в виде $x = f(x)$. Задаются начальное значение аргумента x_0 и точность ε . Первое приближение решения x_1 , находим из выражения $x_1 = f(x_0)$; второе $x_2 = f(x_1)$ и т.д. В общем случае $(i+1)$ приближения найдем по формуле: $x_{i+1} = f(x_i)$. Указанную процедуру повторяем пока $|f(x_i)| > \varepsilon$. Условие сходимости $|f(x)| < 1$.

Задания

Решить уравнения методом половинного деления и методом итераций:

1. $x^3 + 2x - 7 = 0$; $x \in [1; 2]$; $\varepsilon = 0,01$

2. $x^3 + 8x + 10 = 0$; $x \in [-2; -1]$; $\varepsilon = 0,001$

3. $x^3 - 12x + 7 = 0$; $x \in [0; 1]$; $\varepsilon = 0,01$

4. $x^3 - 2x - 5 = 0$; $x \in [2; 3]$; $\varepsilon = 0,01$

5. $x^3 + 7x - 3 = 0$; $x \in [0; 1]$; $\varepsilon = 0,1$

6. $x^3 - x + 8 = 0$; $x \in [-3; -2]$; $\varepsilon = 0,1$

Практическая работа № 3.

Тема: Решение алгебраических и трансцендентных уравнений методами хорд и касательных.

Метод Хорд.

При решении уравнений методом Хорд, задаются интервал $[a; b]$, на котором существует только одно решение и точность, затем через две точки с координатами $(a, F(a))$ и $(b, F(b))$ проводим отрезок по прямой линии с осью ОХ (точка с).

Если при этом $F(a) \cdot F(c) < 0$, то правую границу интервала переносим в точку с ($b=c$). Если указанное условие не выполняется, то в точку с переносится левая граница интервала ($a=c$). Поиск решения прекращается при достижении заданной точности $|F(c)| < \varepsilon$.

Этот метод применяется для решения уравнений вида $f(x)=0$ с отрезком изоляции $[a; b]$, то есть $x \in [a; b]$ и выполняется условие:

1) $f(a) \cdot f(b) < 0$

2) $f'(x)$ сохраняет знак на отрезке $[a, b]$, то есть функция возрастает или убывает на этом отрезке.

Первое приближение корня находится по формуле:

$$x_1 = a - \frac{(b-a) \cdot f(a)}{f(b) - f(a)}$$

и из отрезка $[a, x_1]$ и $[x_1, b]$ выбираем тот, на котором выполняется условие (1). Второе приближение корня:

$$x_2 = a - \frac{(x_1 - a) \cdot f(a)}{f(x_1) - f(a)}, \text{ если } x \in [a; x_1] \quad \text{и по формуле}$$

$$x_2 = x_1 - \frac{(b - x_1) \cdot f(x_1)}{f(b) - f(x_1)}, \text{ если } x \in [x_1; b]$$

Так продолжается до тех пор, пока не достигнет заданной точности.

Метод касательных.

При решении уравнений методом касательных, задается начальным значением аргумента X_0 и точность ε . Затем в точке $(x_1, F(x_0))$ проводим касательную к графику $f(x)$, и определяем точку пересечения касательной с осью абсцисс x_1 .

В точке $(x_1, F(x))$ снова строим касательную; находим следовательно приближенную искомого решение и т.д. Указанную процедуру повторим пока $|F(x_i)| > \varepsilon$.

Метод касательных применяется для уравнений вида $f(x)=0$, если корень x отделенный, $x \in [a; b]$, и выполняется условия:

1) $f(a) \cdot f(b) < 0$;

2) f' и f'' сохраняет свои знаки на отрезке $[a, b]$, то есть функция либо возрастает, либо убывает на отрезке и сохраняет на нем направление выпуклости.

На отрезке $[a, b]$ выбирается такое число x_0 , для которого добивается условие

$$f'(x_0) \cdot f''(x_0) > 0$$

Таким образом, выбирается точка с абсциссой x_0 , которая касается графика функции $f(x)$, пересекает ось ОХ на отрезке $[a, b]$, выбираем один из отрезков $[a, b] \Rightarrow$ выбираем x_0

Первое приближения корня находится по формуле:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$
$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

вычисляется до тех пор, пока не выполнится условие $|x_n - x_{n-1}| < \varepsilon$.

Комбинированный метод хорд и касательных.

Этот метод применяется, если выполняется два условия:

- 1) $f(a) \cdot f(b) < 0$;
- 2) $f'(x)$ и $f''(x)$ сохраняет знак на отрезке $[a, b]$, то есть график функции возрастает, либо убывает на отрезке $[a, b]$ и сохраняет направление выпуклости.

Схема нахождения корня уравнения по комбинированному методу:

1. Найти знак $f(a)$ и $f(b)$;
2. Проверить выполнения условия (1);
3. Найти производные $f'(x)$ и $f''(x)$;
4. Проверить постоянство знака производной на отрезке $[a; b]$;
5. Найти приближения корня для метода касательных. За x_0 выбираем

тот из концов отрезка $[a; b]$, для которого выполняется условие $f(x_0) \cdot f''(x_0) > 0$;

6. а) по методу касательных $x_{11} = x_0 - \frac{f(x_0)}{f'(x_0)}$ - первое приближение корня;

$$x_{11} = x_0 - \frac{f(x_0)}{f'(x_0)}$$

б) по методу хорд
$$x_{12} = a - \frac{(b-a) \cdot f(a)}{f(b) - f(a)}$$

7. $\xi = \frac{x_{11} + x_{12}}{2}$;

8. Проверим, выполняется ли условие $|\xi - x_{11}| < \varepsilon$, где ε - заданная точность. Если условие не выполняется, то продолжаем применять метод по схеме 1-8.

Задания

Решить уравнения методом хорд, методом касательных и комбинированным методом:

1. $x^3 + 2x - 7 = 0$; $x \in [1; 2]$; $\varepsilon = 0,01$
2. $x^3 + 8x + 10 = 0$; $x \in [-2; -1]$; $\varepsilon = 0,001$
3. $x^3 - 12x + 7 = 0$; $x \in [0; 1]$; $\varepsilon = 0,01$
4. $x^3 - 2x - 5 = 0$; $x \in [2; 3]$; $\varepsilon = 0,01$
5. $x^3 + 7x - 3 = 0$; $x \in [0; 1]$; $\varepsilon = 0,1$
6. $x^3 - x + 8 = 0$; $x \in [-3; -2]$; $\varepsilon = 0,1$

Практическая работа № 4.

Тема: Решение систем линейных уравнений приближенными методами.Метод простой итерации

Как отмечалось ранее, итерационные методы используются для решения уравнений и систем любой природы. Рассмотрим, как это делается применительно к системам линейных алгебраических уравнений.

Приведём систему линейных алгебраических уравнений (2.7)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 \dots + a_{2n}x_n = b_2; \\ \\ a_{n1}x_1 + a_{n2}x_2 \dots + a_{nn}x_n = b_n. \end{cases}$$

к равносильной ей системе вида $x = Ax$:

$$\begin{cases} x_1 = \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n + \beta_1, \\ x_2 = \alpha_{21}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n + \beta_2, \\ \\ x_n = \alpha_{n1}x_1 + \alpha_{n2}x_2 + ... + \alpha_{nn}x_n + \beta_n \end{cases} \quad (2.10)$$

В сокращенной форме:

$$x_i = \sum_{j=1}^n \alpha_{ij} x_j + \beta_i, \quad i = \overline{1, n}$$

О системе (2.10) говорят, что она «приведена к нормальному виду».

$$F: y_i = \sum_{j=1}^n \alpha_{ij} x_j + \beta_i \quad (2.11),$$

Правая часть системы определяет отображение

переводящее точку $\bar{x}(x_1, x_2, \dots, x_n)$ в точку $\bar{y}(y_1, y_2, \dots, y_n)$. Используя отображения (2.11) и

выбрав начальную точку $\bar{x}^0(x_1^0, x_2^0, \dots, x_n^0)$ можно построить итерационную последовательность точек.

Если отображение F является сжимающим, то эта последовательность сходится и её предел является решением системы (2.10) а, следовательно, и решением исходной системы (2.7).

Замечание: Отображение является *сжимающим*, если расстояние между образами меньше, чем расстояние между исходными точками.

Для отображения (2.11) необходимым и достаточным условием сжимаемости

$$\max_{1 \leq j \leq n} \sum_{i=1}^n |\alpha_{ij}| < 1$$

является следующее: $\sum_{1 \leq j \leq n} |a_{ij}| \leq 1$, (2.12), т.е. максимальная из сумм модулей коэффициентов при неизвестных в правой части системы (2.10), взятых по столбцам, должен быть меньше 1.

Практическая схема решения с.л.у. методом простой итерации

С.л.у. (2.7) необходимо привести к нормальному виду (2.10).

Для обеспечения сходимости итерационной последовательности необходимо,

чтобы коэффициенты α_{ij} при неизвестных в правой части системы были существенно меньше 1. Этого можно достичь, если исходную систему (2.7) с помощью равносильных преобразований привести к системе, у которой абсолютная величина коэффициентов, стоящих на главной диагонали, больше абсолютных величин каждого из других

коэффициентов, стоящих при неизвестных в соответствующих уровнях (такую систему называют *системой с преобладающими диагональными коэффициентами*). Если теперь разделить все уравнения на соответствующие диагональные коэффициенты и выразить из каждого уравнения неизвестное с коэффициентом, равным 1, будет получена система (2.10), у которой все $|\alpha_{ij}| < 1$.

Для проверки точности решения используем условие (2.12).

Пример: Решить систему линейных уравнений

$$\begin{cases} 2,34x_1 - 4,21x_2 - 11,61x_3 = 14,41, \\ 8,04x_1 + 5,22x_2 + 0,27x_3 = -6,44, \\ 3,92x_1 - 7,99x_2 + 8,37x_3 = 55,56. \end{cases}$$

методом простой итерации с точностью $\varepsilon = 1 \cdot 10^{-4}$.

Решение:

Построим систему с преобладающими диагональными коэффициентами.

В качестве 1-ого уравнения возьмем 2-ое, в качестве 3-его уравнения – 1-ое, в качестве 2-ого уравнения – сумму 1-го и 2-го уравнений:

$$\begin{cases} 8,04x_1 + 5,22x_2 + 0,27x_3 = -6,44, \\ 6,26x_1 - 12,2x_2 - 3,24x_3 = 69,97, \\ 2,34x_1 - 4,21x_2 - 11,61x_3 = 14,41. \end{cases}$$

Разделим каждое из полученных уравнений на диагональный коэффициент и выразим из каждого уравнения диагональные элементы:

$$\begin{cases} x_1 = -0,649x_2 - 0,034x_3 - 0,801, \\ x_2 = 0,573x_1 - 0,266x_3 - 5,735, \\ x_3 = 0,202x_1 - 0,363x_2 - 1,241. \end{cases}$$

Метод Зейделя

Будем снова рассматривать систему линейных уравнений (2.7) и эквивалентную ей систему (2.10).

При решении системы (2.10) методом простой итерации каждый шаг итерационного процесса состоит в переходе от уже имеющегося приближения значений неизвестных к новому (очередному) приближению.

Обозначим элементы имеющегося приближения через $x_1, x_2, \dots, x_n$, а элементы очередного (вычисляемого) приближения через $y_1, y_2, \dots, y_n$.

Вычислительные формулы имеют вид:

$$y_i = \sum_{j=1}^n \alpha_{ij} x_j + \beta_i, \quad i = 1, 2, \dots, n$$

Основная идея метода Зейделя состоит в том, что на каждом шаге итерационного процесса при вычислении значения y_i учитываются уже полученные значения $y_1, y_2, \dots, y_{i-1}$. Выпишем соответствующие вычислительные формулы:

$$y_1 = \sum_{j=1}^n \alpha_{1j} x_j + \beta_1,$$

$$y_2 = \alpha_{21} y_1 + \sum_{j=2}^n \alpha_{2j} x_j + \beta_2,$$

$$y_i = \sum_{j=1}^{i-1} \alpha_{ij} y_j + \sum_{j=i}^n \alpha_{ij} x_j + \beta_i$$

$$y_n = \sum_{j=1}^{n-1} \alpha_{nj} y_j + \alpha_{nn} x_n + \beta_n$$

Справедливо следующее **утверждение**:

Если для матрицы коэффициентов системы (2.10) выполняется условие (2.12), то итерационный процесс метода Зейделя сходится к решению системы при любом выборе начального приближения $x^{(0)}$.

Преимущество этого метода состоит в том, что он обеспечивает более быструю сходимость, чем метод простой итерации.

Пример: Решить систему линейных уравнений

$$10x_1 + x_2 + x_3 = 12$$

$$2x_1 + 10x_2 + x_3 = 13$$

$$2x_1 + 2x_2 + 10x_3 = 14$$

$$x_1 = 1,2 - 0,1x_2 - 0,1x_3 \quad x_2 = 1,3 - 0,2x_1 -$$

$$0,1x_3 \quad x_3 = 1,4 - 0,2x_1 - 0,2x_2$$

$$x^{(0)} = 1,2; x^{(0)} = 0; x^{(0)} = 0$$

$$x^{(1)} = 1,2 - 0,1 \cdot 0 - 0,1 \cdot 0$$

$$\begin{cases} x^{(1)} = 1,3 - 0,2 \cdot 1,2 - 0,1 \cdot 0 \\ x^{(1)} = 1,4 - 0,2 \cdot 1,2 - 0,2 \cdot 0 \end{cases}$$

$$x^{(1)} = 1,2; x^{(1)} = 1,06; x^{(1)} = 1,16$$

$$i = 2(i - \text{приближение})$$

$$x^{(2)} = 1,2 - 0,1 \cdot 1,06 - 0,1 \cdot 1,16$$

$$\begin{cases} x^{(2)} = 1,3 - 0,2 \cdot 1,2 - 0,1 \cdot 1,16 \\ x^{(2)} = 1,4 - 0,2 \cdot 1,2 - 0,2 \cdot 1,06 \end{cases}$$

$$x^{(2)} = 0,978; x^{(2)} = 0,944; x^{(2)} = 0,948$$

Задания

Методом Зейделя решить с точностью 0,001 систему линейных уравнений, приведя ее к виду, удобному для итераций.

$$\begin{cases} 1,7x_1 + 2,8x_2 + 1,9x_3 = 0,7; \\ 2,1x_1 + 3,4x_2 + 1,8x_3 = 1,1; \\ 4,2x_1 - 1,7x_2 + 1,3x_3 = 2,8. \end{cases}$$

1.

$$\begin{aligned} 2. \quad & \begin{cases} 2,7x_1 + 3,3x_2 + 1,3x_3 = 2,1; \\ 3,5x_1 - 1,7x_2 + 2,8x_3 = 1,7; \\ 4,1x_1 + 5,8x_2 - 1,7x_3 = 0,8. \end{cases} \end{aligned}$$

$$\begin{aligned} 3. \quad & \begin{cases} 3,1x_1 + 2,8x_2 + 1,9x_3 = 0,2; \\ 1,9x_1 + 3,1x_2 + 2,1x_3 = 2,1; \\ 7,5x_1 + 3,8x_2 + 4,8x_3 = 5,6. \end{cases} \end{aligned}$$

$$\begin{aligned} 4. \quad & \begin{cases} 9,1x_1 + 5,6x_2 + 7,8x_3 = 9,8; \\ 3,8x_1 + 5,1x_2 + 2,8x_3 = 6,7; \\ 4,1x_1 + 5,7x_2 + 1,2x_3 = 5,8. \end{cases} \end{aligned}$$

$$\begin{aligned} 5. \quad & \begin{cases} 3,3x_1 + 2,1x_2 + 2,8x_3 = 0,8; \\ 4,1x_1 + 3,7x_2 + 4,8x_3 = 5,7; \\ 2,7x_1 + 1,8x_2 + 1, x_3 = 3,2. \end{cases} \end{aligned}$$

$$\begin{aligned} 6. \quad & \begin{cases} 7,6x_1 + 5,8x_2 + 4,7x_3 = 10,1; \\ 3,8x_1 + 4,1x_2 + 2,7x_3 = 9,7; \\ 2,9x_1 + 2,1x_2 + 3,8x_3 = 7,8. \end{cases} \end{aligned}$$

Практическая работа № 5.

Тема: Составление интерполяционных формул Лагранжа, Ньютона.

Постановка задачи аппроксимации функций

В вычислительной математике нередки случаи, когда одну функцию приходится заменять другой, более другой и удобной для дальнейшей работы. Такую задачу называют **аппроксимацией** функций. Поводом для аппроксимации функции может послужить, в частности, табличный способ её задания. Предположим, что результате некоторого эксперимента для конечного набора значений величины x из отрезка $[a;b]$: $a = x_0 < x_1 < \dots < x_n = b$

получен набор значений y_i величины y

Таблица (3.1)

x	x_0	x_1	$\dots$	x_i	$\dots$	x_n
$F(x)$	y_0	y_1	$\dots$	y_i	$\dots$	y_n

Допустим, существует функциональная зависимость $y=F(x)$. Необходимо задать $F(x)$ аналитически.

Точки $x_0, \dots, x_n$ называют **узлами аппроксимации**. Классический подход к численному решению подобных задач заключается в том, чтобы, опираясь на информацию о функции F , по некоторому алгоритму подобрать аппроксимирующую функцию G , в определенном смысле «близкую» к F .

Чаще всего задача аппроксимации решается с помощью многочленов. Вычисления значений многочлена легко автоматизировать, производная и интеграл от многочлена, в свою очередь, также являются многочленами.

Для оценки «близости» функций выбирают тот или иной **критерий согласия**.

Для функций, заданных таблично, достаточно распространенным является **критерий Чебышева**, который определяет расстояние ρ между аппроксимируемой и аппроксимирующей функциями как максимум величины отклонения

$$\rho = \max_i |F(x_i) - G(x_i)| \quad (3.1)$$

между этими функциями в узлах:

Если $\rho = 0$, т.е. $F(x_i) = G(x_i) = y_i$ (в узлах значения совпадают), то соответствующий способ аппроксимации называют **интерполяцией**, а процедуру вычисления значений $F(x)$ с помощью $G(x)$ в точках, не являющихся узлами сетки, **-интерполированием**.

Часто процедура аппроксимации связана с другим критерием согласия:

$$\rho = \min_i \sum_{i=0}^n (F(x_i) - G(x_i))^2$$

Применяемый на его основе способ аппроксимации называется **методом наименьших квадратов**.

Существование и единственность интерполяционного многочлена

Пусть известны значения некоторой функции $F(x)$:

x	x_0	x_1	$\dots$	x_i	$\dots$	x_n
$F(x)$	y_0	y_1	$\dots$	y_i	$\dots$	y_n

Будем решать задачу интерполирования этой функции с помощью построения интерполяционного многочлена n -ой степени.

$G(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ (3.2), который в узлах принимает значения y_i

По формуле (3.9) получаем:

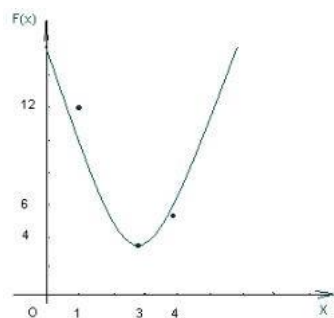
$$L_2(x) = 12 \frac{(x-3)(x-4)}{(1-3)(1-4)} + 4 \frac{(x-1)(x-4)}{(3-1)(3-4)} + 6 \frac{(x-1)(x-3)}{(4-1)(4-3)} =$$

$$= 12 \frac{(x^2 - 7x + 12)}{-2(-3)} + 4 \frac{(x^2 - 5x + 4)}{2(-1)} + 6 \frac{(x^2 - 4x + 3)}{3 \cdot 1} =$$

$$= 2(x^2 - 7x + 12) - 2(x^2 - 5x + 4) + 2(x^2 - 4x + 3) = 2x^2 - 12x + 22.$$

Таким образом, интерполяционный многочлен для заданной функции имеет вид $L_2(x) = 2x^2 - 12x + 22$.

Построим график $L_2(x)$ и точки в одной координатной плоскости.



Задания

Задание 1. По заданной таблице значений функции

x	x_0	x_1	x_2	x_3
y	y_0	y_1	y_2	y_3

составить формулу интерполяционного многочлена Лагранжа. Построить его график и отметить на нем узловые точки.

Задание 2. Вычислить с помощью калькулятора одно значение заданной функции для промежуточного значения аргумента с помощью интерполяционного многочлена Лагранжа и оценить погрешность интерполяции.

	<i>Вариант</i>									
	1	2	3	4	5	6	7	8	9	10
x_0	-1	2	0	7	-3	1	-1	2	-4	-1
x_1	0	3	2	9	-1	2	-1	4	-2	1,5
x_2	3	5	3	13	3	4	2	5	0	3
x_3	4	6	5	15	5	7	4	7	3	5
y_0	-3	4	-1	2	7	-3	4	9	2	4
y_1	5	1	-4	-2	-1	-7	9	-3	8	-7
y_2	2	7	2	3	4	2	1	6	5	1
y_3	-6	2	-8	-4	-6	8	6	-2	10	-8
x	3,8	3,5	0,5	4,8	4,1	3,9	3,3	4	2,9	5,3

Практическая работа № 6.

Тема: Нахождение интерполяционных многочленов сплайнами.

Интерполяция сплайнами

При большом количестве узлов интерполяции сильно возрастает степень интерполяционных многочленов, что делает их неудобными для вычислений.

Высокой степени многочлена можно избежать, разбив отрезок интерполяции на несколько частей, с последующим построением на каждой части самостоятельного интерполяционного многочлена.

Однако такое интерполирование наталкивается на существенный недостаток: в точках стыка разных интерполяционных многочленов бывает разрывной их первая производная.

В этом случае удобно пользоваться особым видом кусочно-полиномиальной интерполяции - интерполяции *сплайнами*.

Суть этого подхода заключается в следующем:

Определение: Функция $S_m(x)$ называется *интерполяционным сплайном* порядка m для функции $f(x)$, заданной таблицей:

x	x_0	x_1	$\dots$	x_i	$\dots$	x_n
y	y_0	y_1	$\dots$	y_i	$\dots$	y_n
	0	1		i		

если:

1. на каждом отрезке $[x_i; x_{i+1}]$ ($i=0, \dots, n-1$) $S(x)$ является многочленом порядка m ;
2. $S(x)$ и её производная до $(m-1)$ -го порядка включительно непрерывны на $[x_0; x_n]$;
3. $S(x_i)=y_i$ ($i=0, \dots, n$) - непосредственно условие интерполяции.

Остановимся на построении наиболее популярных в практике аппроксимации функций *кубических сплайнов*.

По определению кубический сплайн $S(x)$ можно представить в виде

$$S(x) = \begin{cases} P_1(x), & x \in [x_0; x_1]; \\ P_2(x), & x \in [x_1; x_2]; \\ \\ P_n(x), & x \in [x_{n-1}; x_n]. \end{cases} \quad (3.17)$$

Где каждый из $P_i(x)$ - многочлен третьей степени:

$$P_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, i = \overline{1, n} \quad (3.18)$$

Коэффициенты a_i найдем из условия: $y_i = S(x_i) = P_i(x_i) = a_i$, т.е. $a_i = y_i$ (3.19)

Условие непрерывности $S(x)$ в каждом узле приводит к равенствам:

$$P_i(x_i) = P_{i+1}(x_i), \quad i = \overline{1, n-1}$$

В развернутом виде с учетом формулы (3.18) эти равенства примут вид:

$$a_i = a_{i+1} + b_{i+1}(x_i - x_{i+1}) + C_{i+1}(x_i - x_{i+1})^2 + d_{i+1}(x_i - x_{i+1})^3 \quad (3.20)$$

Введем обозначения: $h_i = x_i - x_{i-1}$. Понижая в равенстве (3.20) индекс на единицу (меняем i на $i-1$) и,

учитывая (3.19), получим: $h_i b_i - h_i^2 c_i + h_i^3 d_i = y_i - y_{i-1}$ (3.21)

Условие непрерывности первой производной кубического сплайна сводится к

требованию $P_i'(x_i) = P_{i+1}'(x_i), \quad i = \overline{1, n-1}$.

используя, введем обозначения, получим:

Из условия непрерывности второй производной:

Тогда дифференцируя формулу (3.18) и

$$b_{i-1} - b_i + 2h_i c_i - 3h_i^2 d_i = 0 \quad (i = \overline{2, n}) \quad (3.22)$$

$$P_i''(x_i) = P_{i+1}''(x_i) \text{ получим:}$$

) (3.23). Составим систему из равенств (3.21)-(3.23) и, решив её, найдем коэффициенты b_i, C_i, d_i .

Однако, для однозначной ее разрешимости добавим условия непрерывности на концах отрезка:

$$S''(x_0) = S''(x_n) = 0, \quad P_1''(x_0) = 0, \quad P_n''(x_n) = 0 \quad \text{т.е.}$$

$$\begin{cases} C_1 - 3h_1 d_1 = 0 \\ a_n = 0 \end{cases} \quad (3.24)$$

В результате получаем систему уравнений:

$$\begin{cases} C_1 - 3h_1 d_1 = 0; \\ a_n = 0; \\ h_i b_i - h_i^2 c_i + h_i^3 d_i = y_i - y_{i-1}; \\ b_{i-1} - b_i + 2h_i c_i - 3h_i^2 d_i = 0; \\ C_{i-1} - C_i + 3h_i d_i = 0. \end{cases}$$

Последовательно, исключая переменные получим

$$h_{i+1} C_{i+1} + 2(h_i + h_{i+1}) C_i + h_i C_{i-1} = 3 \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right) \quad (3.25)$$

(это уравнение содержит лишь C_i).

$$d_i = \frac{C_i - C_{i-1}}{3h_i} \quad (3.26)$$

(это уравнение содержит лишь неизвестные d_i).

$$b_i = \frac{y_i - y_{i-1}}{h_i} + h_i C_i - h_i^2 d_i \quad (3.27)$$

(это уравнение содержит лишь неизвестные b_i).

Построив кубический сплайн, найдем оценку погрешности интерполяции:

$$|f(x) - S(x)| \leq \max_{[a;b]} |f^{(4)}(x)|, \quad \text{где } [a;b] \text{ - промежуток интерполяции.}$$

Пример: Построить кубический сплайн для функции $y=f(x)$, заданной таблицей:

x_i	-1	0	1	2
y_i	1/2	1	2	4

с дополнительным условием: $S''(-1) = S''(2) = 0$. Найти с помощью $S(x)$ значения функции при $x=0,3$. (Заметим, что в основу таблицы положена функция $y=2^x$).

Решение: $C_0 = 0$ (т.к. не используется в функциях) и $C_3 = 0$ (т.к. из условия

(3.24): $C_n = 0$).

Шаг таблицы $h_i = 1$.

из (3.25) получаем:

$$\begin{cases} 1 \cdot C_2 + 2(1+1)C_1 + 1 \cdot C_0 = 3 \left(\frac{y_2 - y_1}{1} - \frac{y_1 - y_0}{1} \right), \text{ при } i=1, \\ 1 \cdot C_3 + 2(1+1)C_2 + 1 \cdot C_1 = 3 \left(\frac{y_3 - y_2}{1} - \frac{y_2 - y_1}{1} \right), \text{ при } i=2. \end{cases}$$

$$\begin{cases} C_2 + 4C_1 = 3 \left(\frac{2-1}{1} - \frac{1-\frac{1}{2}}{1} \right) = 3 \left(1 - \frac{1}{2} \right) = \frac{3}{2}, \\ D + 4C_2 + C_1 = 3 \left(\frac{4-2}{1} - \frac{2-1}{1} \right) = 3(2-1) = 3. \end{cases}$$

$$\begin{cases} C_2 + 4C_1 = \frac{3}{2}, \\ 4C_2 + C_1 = 3. \end{cases}$$

$$15C_1 = 6 - 3 = 3,$$

$$C_1 = \frac{1}{5}, \quad C_2 = \frac{3}{2} - \frac{4}{5} = \frac{7}{10}.$$

Из (3.26) имеем:

$$d_1 = \frac{C_1 - C_0}{3h} = \frac{\frac{1}{5} - 0}{3 \cdot 1} = \frac{1}{15},$$

$$d_2 = \frac{C_2 - C_1}{3h} = \frac{\frac{7}{10} - \frac{1}{5}}{3} = \frac{1}{6},$$

$$d_3 = \frac{C_3 - C_2}{3h} = \frac{0 - \frac{7}{10}}{3} = -\frac{7}{30}.$$

Из (3.27) имеем:

$$b_1 = \frac{y_1 - y_0}{h} + hC_1 - h^2d_1 = \frac{1 - \frac{1}{2}}{1} + 1 \cdot \frac{1}{5} - 1^2 \cdot \frac{1}{15} = \frac{1}{2} + \frac{1}{5} - \frac{1}{15} = \frac{19}{30}$$

$$b_2 = \frac{y_2 - y_1}{h} + hC_2 - h^2d_2 = \frac{2-1}{1} + 1 \cdot \frac{7}{10} - 1 \cdot \frac{1}{6} = 1 + \frac{7}{10} - \frac{1}{6} = \frac{23}{15},$$

$$b_3 = \frac{y_3 - y_2}{h} + hC_3 - h^2d_3 = \frac{4-2}{1} + 1 \cdot 0 - 1 \cdot \left(-\frac{7}{30} \right) = 2 + \frac{7}{30} = \frac{67}{30}.$$

из формулы (3.28) получаем:

$$P_1(x) = a_1 + b_1(x - x_1) + C_1(x - x_1)^2 + d_1(x - x_1)^3, x \in [x_0; x_1]$$

$$P_1(x) = 1 + \frac{19}{30}(x - 0) + \frac{1}{5}(x - 0)^2 + \frac{1}{15}(x - 0)^3, \text{ т.е. } P_1(x) = 1 + \frac{19}{30}x + \frac{1}{5}x^2 + \frac{1}{15}x^3, x \in [-1; 0]$$

$$P_2(x) = a_2 + b_2(x - x_2) + C_2(x - x_2)^2 + d_2(x - x_2)^3, x \in [x_1; x_2],$$

$$P_2(x) = 2 + \frac{23}{15}(x - 1) + \frac{7}{10}(x - 1)^2 + \frac{1}{6}(x - 1)^3, x \in [x_1; x_2]$$

$$P_3(x) = a_3 + b_3(x - x_3) + C_3(x - x_3)^2 + d_3(x - x_3)^3, x \in [x_2; x_3],$$

$$P_3(x) = 4 + \frac{67}{30}(x - 2) - \frac{7}{30}(x - 2)^3, x \in [1; 2]$$

Следовательно, сплайн $S(x)$ построен:

$$S(x) = \begin{cases} P_1(x) = 1 + \frac{19}{30}x + \frac{1}{5}x^2 + \frac{1}{15}x^3, x \in [-1; 0]; \\ P_2(x) = 2 + \frac{23}{15}(x - 1) + \frac{7}{10}(x - 1)^2 + \frac{1}{6}(x - 1)^3, x \in [0; 1]; \\ P_3(x) = 4 + \frac{67}{30}(x - 2) - \frac{7}{30}(x - 2)^3, x \in [1; 2]. \end{cases}$$

Найдем его значение при $x=0,3$:

: Заметим, что $0,3 \in [0;1]$, поэтому используем многочлен $P_2(x)$

$$P(0.3) = 2 + \frac{23}{15}(0,3-1) + \frac{7}{10}(0,3-1)^2 + \frac{1}{6}(0,3-1)^3 = 1,2125$$

Отметим для сопоставления с той же точностью значение функции, положенной в основу данного примера: $f(x) = 2^x$; $f(0.3) = 2^{0,3} = 1,2311$.

Интерполяция сплайнами сопряжена с немалым объемом вычислительной работы. Весьма необычна и форма окончательного результата, ибо сплайн имеет различные представления на различных частичных отрезках интерполяции. Это осложняет доступ к значениям сплайна в каждой конкретной точке, так как предполагает, прежде всего, поиск параметров, определяющих соответствующую форму сплайна. Эти трудности легко предотвратимы при использовании компьютера, так как упорядоченное хранение всех необходимых параметров организовать нетрудно, а выполнение однотипных процедур по вычислению параметров сплайна и его значений может быть обеспечено специальными процедурами.

Задания.

Задание 1. По заданной таблице значений функции

x	x_0	x_1	x_2	x_3
y	y_0	y_1	y_2	y_3

вычислить коэффициенты и составить формулы кубического сплайна.

Задание 2. Результат интерполирования проверить путем вычисления значений сплайна узловых точках.

	Вариант									
	1	2	3	4	5	6	7	8	9	10
x_0	-1	2	0	7	-3	1	-1	2	-4	-1
x_1	0	3	2	9	-1	2	1	4	-2	0
x_2	1	4	4	11	1	3	3	6	0	1
x_3	2	5	6	13	3	4	5	7	2	2
y_0	-3	4	-1	2	7	-3	4	9	2	4
y_1	5	1	-4	-2	-1	-7	9	-3	8	-7
y_2	2	7	2	3	4	2	1	6	5	1
y_3	-6	2	-8	-4	-6	8	6	-2	10	-8
x	3,8	3,5	0,5	4,8	4,1	3,9	3,3	4	2,9	5,3

ОП.11 Компьютерные сети

Практическая работа №1 Монтаж кабельных сетей технологий Ethernet. Построение одноранговой сети. Построение схемы компьютерной сети.

Преимущества и недостатки проводных сетей

Структурированная кабельная система (СКС) – законченная совокупность кабелей связи и коммутационного оборудования, отвечающая требованиям соответствующих нормативных документов.

СКС включает набор кабелей и коммутационных элементов, и методику их совместного использования, позволяющую создавать регулярные расширяемые структуры связей в локальных сетях различного назначения. СКС – физическая основа инфраструктуры здания, позволяющая свести в единую систему множество сетевых информационных сервисов разного назначения: локальные вычислительные сети и телефонные сети, системы безопасности, видеонаблюдения и т. д.

СКС представляет собой иерархическую кабельную систему, смонтированную в здании или в группе зданий, состоящую из структурных подсистем. В состав СКС входят такие элементы, как главный кросс (МС), кабель магистральной подсистемы первого и второго уровня, промежуточные кроссы (IC), горизонтальные кроссы (НС) и кабели горизонтальной подсистемы, а также консолидационные точки (СР), многопользовательские телекоммуникационные розетки (МуТОА или МуТО) и телекоммуникационные розетки (ТО) и другие. Система может быть построена на основе медных или оптических кабелей, все элементы СКС интегрируются в единый комплекс (систему) и эксплуатируются согласно определенным правилам.

Преимущества:

- при относительно высокой начальной стоимости оправдывает капиталовложения за счет длительной и безотказной эксплуатации;
- доступная оптическая и медная УТР-технология;
- обеспечивает модульность и возможность наращивания и легкого внесения изменений;
- способность к передаче на высокой скорости и полосе пропускания;
- открытая архитектура допускает одновременное использование различных протоколов и продуктов различных пользователей;
- не зависит от изменений информационной технологии;
- допускает использование уже существующего активного оборудования;
- использует стандартные компоненты и материалы;
- допускает обслуживание минимальным количеством персонала пользователя;
- соответствует всем существующим стандартам по кабельной проводке здания в настоящее время и в будущем.

Недостатки:

- высокая стоимость проектирования и инсталляции.

Простые (неструктурированные) кабельные сети представляют из себя обычные кабельные системы на основе витой пары, проложенные в кабельных каналах. Для передачи данных и офисной телефонии используются разные кабельные системы. Часто

представляют из себя очень печальное зрелище: провода запутаны, просто валяются на полу, под столами.

Преимущества:

- низкая стоимость монтажа по сравнению с СКС;
- сравнительно высокая скорость монтажа;
- высокая надежность.

Недостатки:

- небольшая гарантия на систему;
- сложность расширения системы, дополнительные затраты на расширение.

Монтаж локальной сети

При монтаже проводных локальных сетей нет ничего запредельно сложного или требующего каких-то специальных навыков или знаний. Достаточно запомнить несколько простых правил и следовать им.

Основным носителем в подавляющем большинстве проводных сетей Ethernet сейчас является **витая пара**. Этот кабель делится по категориям. Наиболее распространен кабель маркировкой UTP-8 Cat 5 или 5e, состоящий из 4-х пар (8 проводников) (рис. 1).

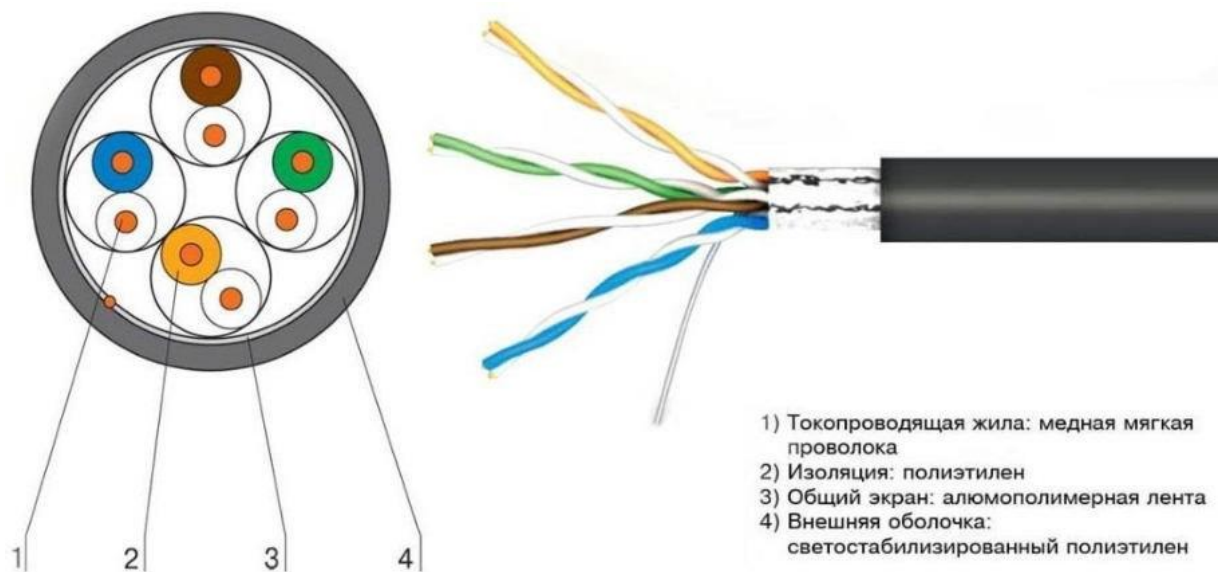


Рис. 1. Витая пара

Каждая пара скручена со своим определённым шагом, что служит защитой от помех. Проводники в парах имеют цветную маркировку: оранжевый+бело-оранжевый, зелёный+бело-зелёный, синий+бело-синий, коричневый+бело-коричневый. Маркировка двухцветных проводов может быть выполнена одним из двух способов: «колечками» — когда дополнительный цвет нанесён на белый проводник через определённое расстояние небольшими колечками, либо «полосками» — когда проводник поделён вдоль пополам, одна половина белого цвета, вторая — дополнительного. Предпочесть следует кабель второго типа по одной простой причине: после зачистки кабеля от изоляции на одном из двухцветных проводов ближайшее колечко может остаться под изоляцией, и мы будем иметь проводник неопределённого белого цвета.

Максимальная длина отрезка кабеля, используемого между двумя устройствами, по стандарту не может быть больше 100 метров. Розетки, переходники и прочие пассивные элементы устройствами в данном контексте не считаются. Устройство принимает сигнал и отправляет дальше, предварительно усилив его. В небольших сетях это концентраторы (hub), коммутаторы (switch hub), мосты (bridge) и маршрутизаторы (router). Ну и, конечно же, сами компьютеры с сетевыми картами. Давайте остановимся на сетевых устройствах чуть более подробно, ведь, в конечном счёте, именно от них в большей степени зависит качество работы и удобство использования нашей будущей сети.

Концентратор – это многопортовый повторитель. То есть он принимает сигнал на один из своих портов (порт – это, проще говоря, гнездо, в которое вставляется сетевой кабель), и, усилив его, передаёт его на все остальные свои порты. Это один из самых дешёвых способов объединить компьютеры в локальную сеть, но у него есть один существенный недостаток. Допустим, в сеть соединены четыре компьютера. Так вот, если с первого компьютера пользователь передаёт файл на второй и, в это же время, с третьего компьютера пользователь передаёт файл на четвёртый, то скорость передачи падает более чем вдвое по сравнению с максимальной. Это происходит из-за того, что сигналы каждого из компьютеров передаются на все порты. Возникают так называемые «коллизии». Для решения этой проблемы были созданы коммутаторы. Внешне они ничем, как правило, не отличаются от концентраторов (да и по стоимости не очень заметно), но работают они принципиально по-другому. При включении компьютера в сеть коммутатор запоминает его физический адрес (MAC-адрес). Таким образом, если с первого компьютера файл передаётся на второй, а с третьего на четвёртый, то коммутатор сигнал, пришедший на первый порт, усиливает и передаёт только на второй порт, а сигнал, пришедший на третий

– только на четвёртый. Таким образом, за счёт этого две передачи файлов происходят одновременно, причём на максимально возможной скорости и безо всяких коллизий.

Мосты используются для связи в одно целое двух отдельных сегментов сети. То есть, допустим, у нас в одном кабинете 4 компьютера, объединённые в локальную сеть, и в другом кабинете 3 компьютера, также объединённые в локальную сеть. Расстояние между кабинетами равно 150 метрам. Нам необходимо объединить эти две сети в одну. Просто протянуть витую пару у нас не получится (как мы помним, её максимальная длина – 100 метров). Значит, надо либо где-то на посторонней территории устанавливать ретранслятор (усилитель сигнала), а это не всегда возможно, либо использовать, например, коаксиальный кабель (его длину стандарт ограничивает уже 200 метрами). Либо, даже при гораздо меньшем расстоянии, но при невозможности проложить вообще какой-либо кабель, использовать беспроводную связь (Wi-Fi). Так вот, мост как раз и даёт возможность соединять сети, построенные на одном носителе (в нашем случае, витой паре), с помощью другого носителя (коаксиального кабеля или радиоволн) в одну сеть. Либо просто объединять в одну сеть два сегмента, построенные на разных носителях (допустим, все стационарные компьютеры подключены к сети с помощью кабеля, а ноутбуки и планшеты – по Wi-Fi).

Маршрутизаторы или роутеры. Сейчас для небольших домашних и офисных сетей это наиболее распространённые устройства. Недостаток коммутатора состоит, прежде всего, в том, что для того, чтобы обеспечить всем выход в Интернет по одной линии (одному подключению), необходимо для этого настроить один из компьютеров, который для этого будет подключён и к локальной сети

и к Интернету. Более того, для обеспечения выхода остальных пользователей в Интернет этот компьютер должен быть постоянно включён даже тогда, когда его пользователь на нём не работает. То же самое относится и к другим общим устройствам, например, принтерам или сканерам (если они используются по сети совместно).

Роутер объединяет в себе функции коммутатора, моста и такого компьютера. То есть он подключается к Интернету, а все остальные подключены к нему как к коммутатору (по кабелю или по Wi-Fi). Наиболее продвинутые модели также позволяют подключать к себе принтеры, сканеры и внешние жёсткие диски при помощи USB. Так же они содержат программное обеспечение, которое может постоянно работать без включения компьютера (например, торрент-клиент). При выборе роутера необходимо учесть все потребности и откинуть всё лишнее (так, для дома, вероятнее всего, вряд ли понадобится 12-и портовый роутер со встроенным принт-сервером. А вот в небольшом офисе это будет практически идеальный вариант. Также выбор модели зависит от способа вашего подключения к провайдеру (наиболее распространены сейчас, по понятным причинам, Ethernet-роутеры и ADSL-роутеры).

Ниже рассмотрим, как создать небольшую локальную сеть, центральным узлом которой будет являться роутер. Для начала надо определиться с его месторасположением. Оно должно отвечать следующим требованиям:

1. Роутер должен быть легко доступен, и к нему без затруднений должен подключаться кабель провайдера Интернет.
2. В непосредственной близости от места размещения роутера (не дальше 1 метра) должна быть электророзетка.
3. Общая длина кабеля, идущего на подключение компьютеров к роутеру, должна быть минимальной.

Кабели обычно проложены либо методом скрытой проводки (в стене), либо в кабель-каналах и под фальшпотолком. На их конце монтируются розетки под разъёмы RJ-45 (рис. 2), а компьютеры в эти розетки подключаются с помощью так называемых патч-кордов. **Патч-корд** – это отрезок гибкой витой пары, длиной от 75 см до 1 м, на обоих концах которого находятся разъёмы RJ-45. Их можно приобрести в любом магазине, торгующем компьютерным оборудованием либо изготовить самостоятельно.

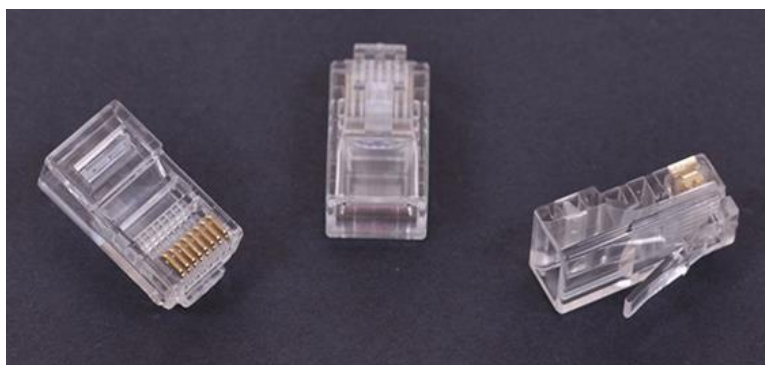


Рис. 2. Разъёмы RJ-45

Для присоединения кабеля «витая пара» к розеткам и для присоединения к кабелю разъёмов RJ-45 нам понадобятся два инструмента:

- **обжимной инструмент RJ-45** (обжимные клещи или кримпер, рис. 3) служит для закрепления на концах кабеля разъёмов

- **устройство для зачистки RJ-45** (куттер, рис. 4) – служит для закрепления кабеля в розетке.



Рис. 4. Устройство для зачистки RJ-45

У разъёма RJ-45 восемь контактов. Нумеруются они следующим образом: если взять разъём и повернуть его фиксирующим язычком

вниз, чтобы металлические контакты были направлены от себя, то контакты слева направо нумеруются с 1 по 8.

Для фиксации разъёма на кабеле («обжима»), необходимо сделать следующие операции (рис. 5):

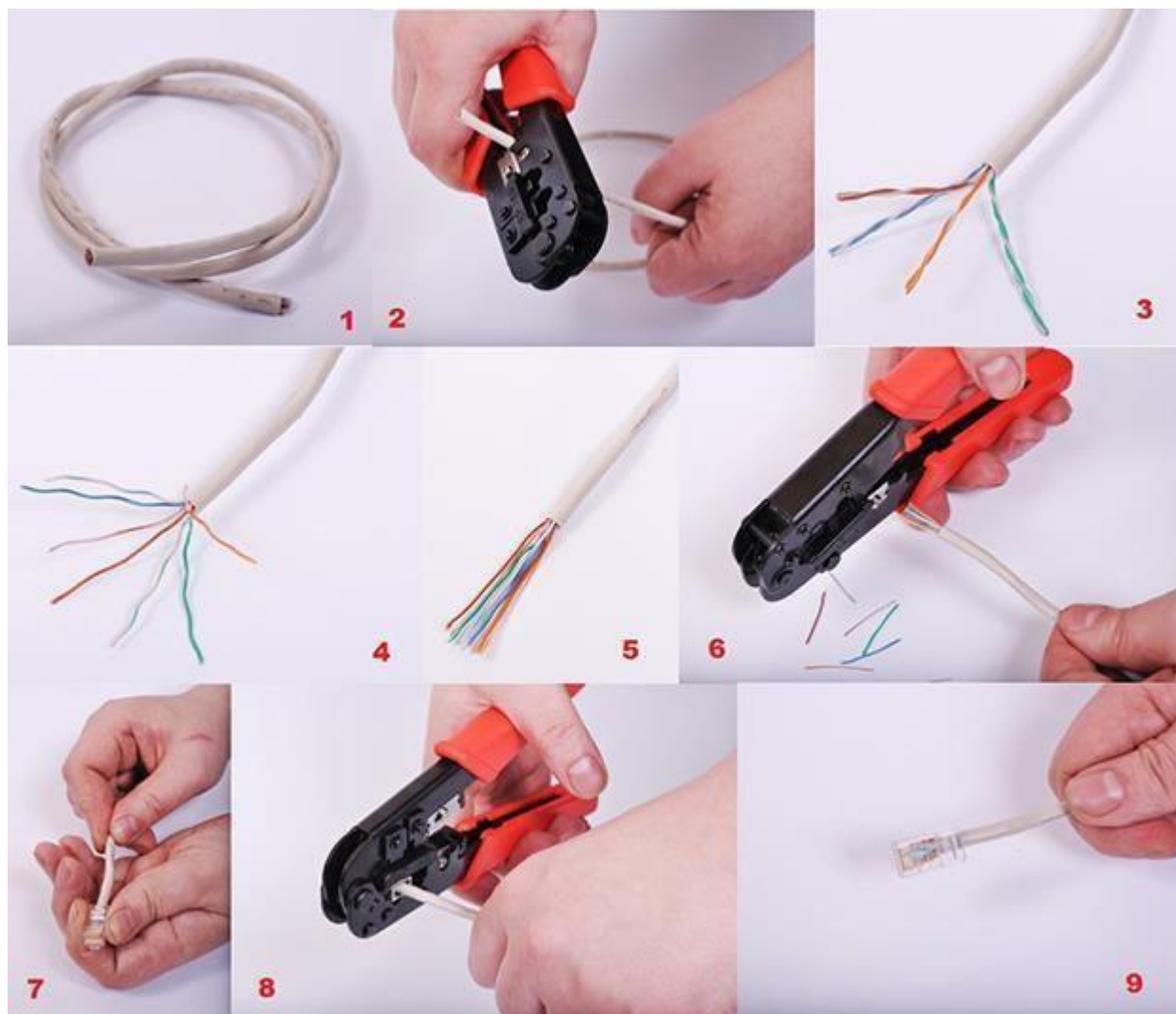


Рис. 5. Операции «обжима» витой пары

- зачистить конец кабеля от изоляции (рис. 5 (1,2));
- раскрутить витые пары и разложить провода в линейку согласно цветовой схемы (рис. 5 (3,4,5));
- срезать лишние провода до необходимой длины (чтобы провода помещались в разъём, а изоляция кабеля входила в него на несколько миллиметров) (рис. 5 (6));
- вставить провода в разъём, внимательно следя за тем, чтобы последовательность проводов не была нарушена (рис. 5 (7));
- вставить разъём в кримпер и сдавливанием рукояток зафиксировать разъём на кабеле (рис. 5 (8));
- извлечь разъём из клещей (рис. 5 (9)).

В настоящее время применяются две схемы обжима: TIA/EIA-568A и TIA/EIA-568B, который используется гораздо чаще. Расположение проводников в них следующее:

№ контакта	TIA/EIA-568A	TIA/EIA-568B	Crossover 1000 Mbit/s
1	Бело-зелёный	Бело-оранжевый	Бело-зелёный
2	Зелёный	Оранжевый	Зелёный
3	Бело-оранжевый	Бело-зелёный	Бело-оранжевый
4	Синий	Синий	Бело-коричневый
5	Бело-синий	Бело-синий	Коричневый
6	Оранжевый	Зелёный	Оранжевый
7	Бело-коричневый	Бело-коричневый	Синий
8	Коричневый	Коричневый	Бело-синий

При соединении компьютера с роутером оба конца кабеля обжимаются одинаково. Для соединения же однотипного оборудования (компьютер соединяется напрямую с компьютером либо, например, два концентратора соединяются между собой), используется так называемая «перекрёстная» схема обжима (crossover). Она также существует в двух вариантах. Первый вариант (который применяется на скоростях до 100 мбит/с и поэтому является наиболее распространённым), один конец кабеля обжимается по схеме «А», а второй – по схеме «В». Для скорости же до 1000 мбит/с применяются следующие схемы обжима: один конец кабеля обжимается по схеме «В», а второй – по схеме «Crossover 1000 Mbit/s».

Обжим розеток производить ещё проще. На розетках уже нанесены цветовые маркировки. Так что, в зависимости от выбранной схемы обжима «А» или «В» вам надо произвести следующие действия:

- с помощью куттера зачистить конец кабеля от изоляции;
- раскрутить пары проводов;
- наложить каждый провод на своё гнездо и с помощью носика куттера сдвинуть их вниз по лезвиям гнезда до упора;
- закрепить розетку в подрозетнике.

После этого необходимо с помощью патч-кордов соединить компьютеры с розетками локальной сети.

Настройка локальной сети

После того, как все компоненты локальной сети физически соединены в одно целое, необходимо их настроить. Первым делом настраиваем центральное устройство нашей сети, то есть роутер. Узнать в подробностях, как это делается, Вы можете здесь. С компьютерами же ещё проще. Обычно все компьютеры под управлением любой версии Windows по-умолчанию настроены как клиенты DHCP.

DHCP – это сервер, который занимается тем, что каждому подключающемуся к сети компьютеру либо другому устройству он выдаёт по заданным правилам все необходимые сетевые настройки, получив которые, компьютер подключается к сети и

может в ней работать. После удачного входа в сеть этот ключ компьютер запоминает, и при последующих подключениях необходимость в его вводе отсутствует.

Но иногда (правда, довольно редко), возникает необходимость в настройке сетевых свойств вручную. Для этого нам необходимо будет зайти в «Центр управления сетями и общим доступом». Проще всего это сделать, кликнув по значку подключения сети в трее и выбрав там «Центр управления сетями и общим доступом» (рис. 6).

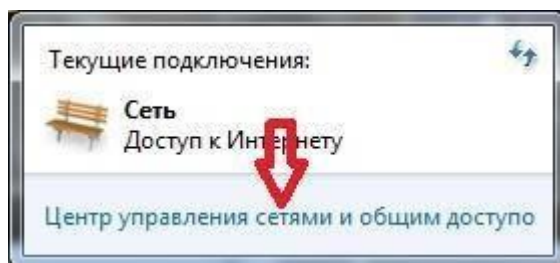


Рис. 6. Центр управления сетями и общим доступом

Либо зайти в меню «Пуск», далее в «Панель управления», в ней войти в раздел «Сеть и Интернет» и там выбрать «Центр управления сетями и общим доступом».

Там, в зависимости от того, какое подключение Вы настраиваете, Вам необходимо будет выбрать либо «Подключение по локальной сети» (1), либо «Беспроводное сетевое соединение» (2) (рис. 7).

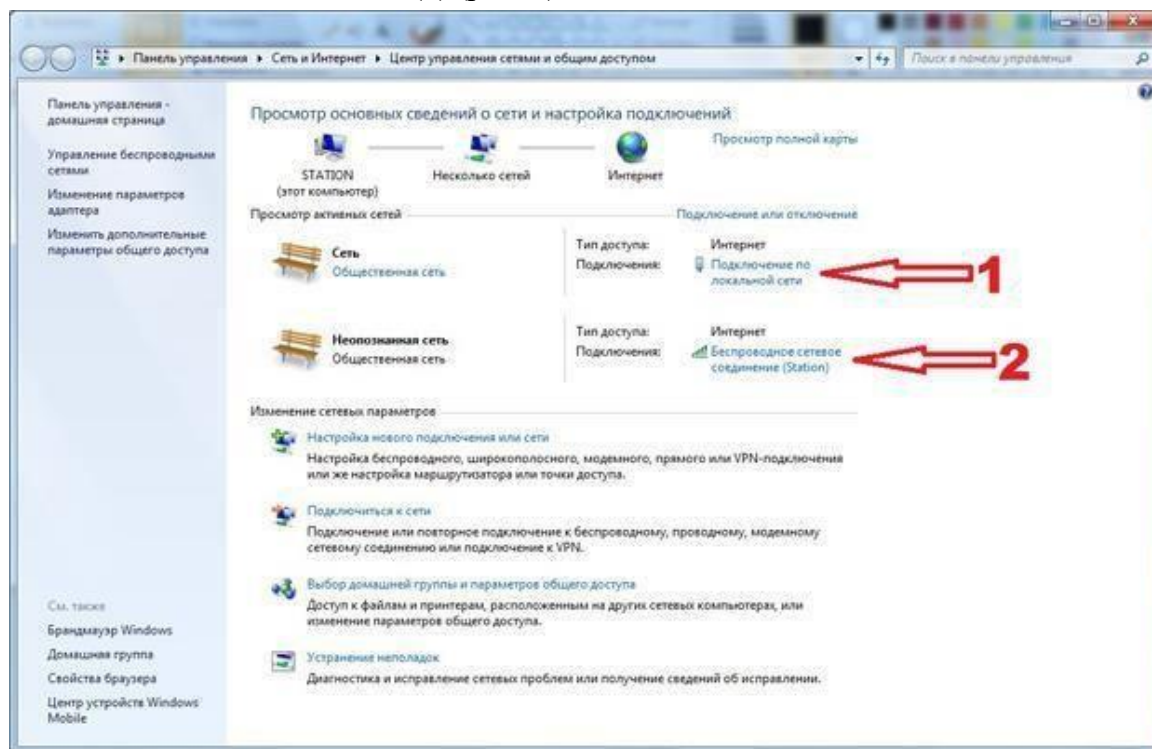


Рис. 7. Настройка сетевых подключений

Ручная настройка сети, вне зависимости от типа подключения, производится следующим образом.

В окне подключения кликнуть по кнопке «Свойства» (рис. 8).

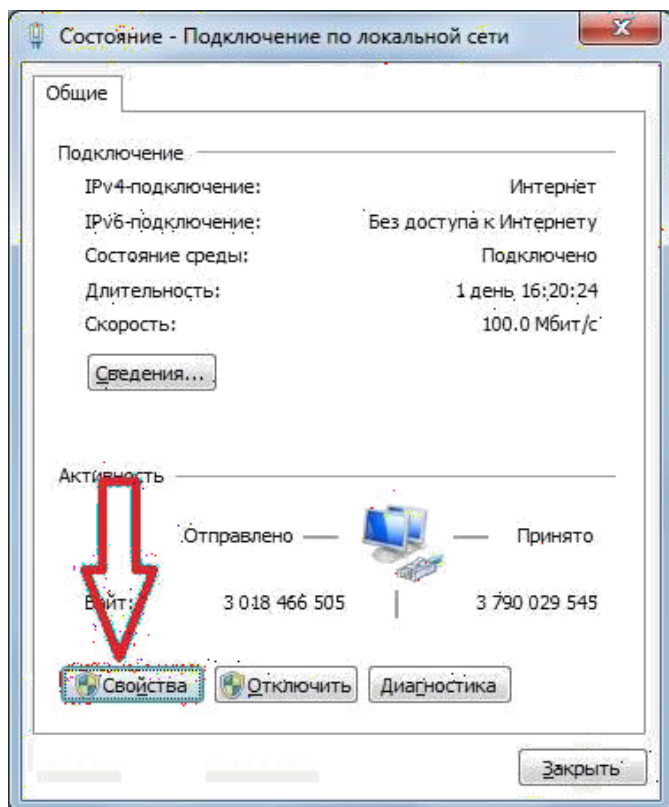


Рис. 8. Окно состояния подключения по локальной сети

Далее необходимо выбрать «Протокол Интернета 4 (TCP/IPv4)» и нажать «Свойства» (рис. 9).

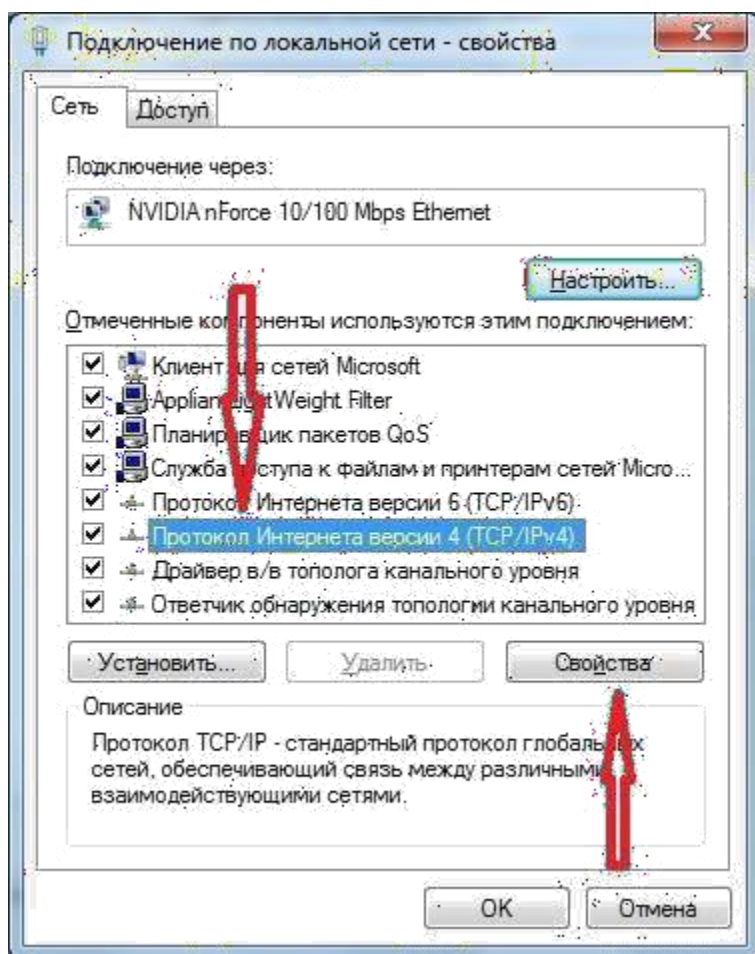


Рис. 9. Настройка параметров подключения по локальной сети

В свойствах необходимо будет ввести IP-адрес Вашего компьютера, маску подсети, шлюз по умолчанию и один либо два адреса (основной и резервный) сервера доменных имён DNS (рис. 10).

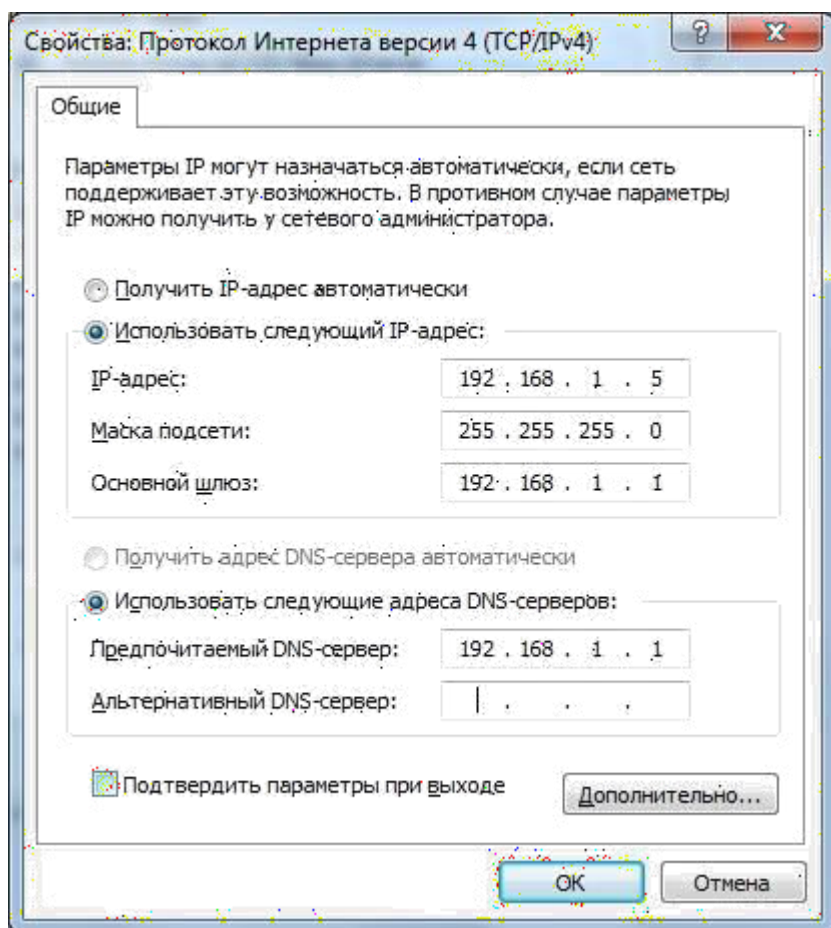


Рис. 10. Настройка параметров протокола TCP/IP

Сервер DNS занимается тем, что по символьному адресу URL, который Вы вводите в браузере, например, www.fixitbook.ru, выдаёт сопоставленный ему IP-адрес, с помощью которого Ваш компьютер может подключиться к нужному и показать Вам тот сайт, который Вам необходим. Сделана эта служба была потому, что названия из слов подавляющее большинство людей запоминает гораздо лучше и проще, нежели последовательность случайных цифр, которую представляет из себя IP-адрес.

Практическая работа №2 «Настройка протокола TCP/IP в операционных системах»Цель: обобщение и систематизация знаний по теме «Межсетевое взаимодействие»

Средства для выполнения работы:

- аппаратные: компьютер, с установленной ОС Windows XP;
- программные: виртуальные машины: VM-1;
- информационные: IP-адрес; маска подсети; основной шлюз; предпочитаемый DNS.

Задания к работе

1. Проверьте работоспособность стека протоколов TCP/IP.

Запустите виртуальную машину VM-1 и загрузите ОС Windows.

Запустите консоль (*Пуск/Программы/Стандартные/Командная строка*).

В командной строке введите `ipconfig /all / more`.

Используя приведенную ниже информацию, создайте в своей папке текстовый документ со следующими данными:

- имя компьютера;
- основной DNS-суффикс;
- описание DNS-суффикса для подключения;
- физический адрес;
- DHCP включен;
- автоконфигурация включена;
- IP-адрес автоконфигурации;
- маска подсети;
- шлюз по умолчанию.

Убедитесь в работоспособности стека *TCP/IP*, отправив эхо-запросы на IP-адреса. Для этого воспользуйтесь командой `ping`:

- отправьте эхо-запросы на локальный адрес компьютера (loopback) `ping 127.0.0.1` (на экране должны появиться сообщения о полученном ответе от узла 127.0.0.1);
- отправьте эхо-запрос по другому IP-адресу, например 172.21.5.1.

2. Настройте стек протоколов TCP/IP для использования статического IP-адреса.

Откройте окно Сетевые подключения (*Пуск/Панель управления/Сетевые подключения*).

Вызовите свойства подключения по локальной сети. Для этого можно воспользоваться контекстным меню.

В появившемся диалоговом окне на вкладке Общие откройте свойства Протокол Интернета TCP/IP.

Щелкните переключатель *Использовать следующий IP-адрес* и введите в соответствующие поля данные: IP_адрес; Маску подсети; Основной шлюз; Предпочитаемый DNS.

Примените параметры кнопкой *ОК*.

Закройте окно свойств подключения кнопкой *ОК* (если потребуется, то согласитесь на перезагрузку компьютера).

Проверьте работоспособность стека протоколов *TCP/IP*.

3. Настройте TCP/IP для автоматического получения IP-адреса.

Откройте окно Сетевые подключения.

Вызовите свойства Подключения по локальной сети.

Откройте свойства Протокол Интернета TCP/IP.

Установите переключатель *Получить IP-адрес автоматически*.

Закройте диалоговое окно Свойства: Протокол Интернета TCP/IP кнопкой *ОК*.

Примените параметры кнопкой *ОК*.

Проверьте настройку стека протоколов *TCP/IP*.

Получите другой адрес для своего компьютера. Для этого:

- запустите консоль (командную строку);
- введите команду для сброса назначенных адресов - `ipconfig /release`;
- введите команду для получения нового адреса `ipconfig / renew`;

Проверьте работоспособность стека протоколов *TCP/IP*.

Контрольные вопросы:

1. Опишите параметры, используемые при настройке статического адреса TCP/IP.
2. Какие преимущества дает применение стека протоколов TCP/IP.
3. Дайте определение понятию стек протоколов TCP/IP.

Практическая работа №3 «Работа с диагностическими утилитами протокола TCP/IP»Цель: обобщение и систематизация знаний по теме «Межсетевое взаимодействие»

Задания к работе

Задание 1. Получение справочной информации по командам.

Выведите на экран справочную информацию по всем рассмотренным утилитам (см. таблицу п.1). Для этого в командной строке введите имя утилиты без параметров и дополните /?.

Сохраните справочную информацию в отдельном файле. Изучите

ключи, используемые при запуске утилит.

Задание 2. Получение имени хоста.

Выведите на экран имя локального хоста с помощью команды hostname. Сохраните результат в отдельном файле.

Задание 3. Изучение утилиты ipconfig.

Проверьте конфигурацию TCP/IP с помощью утилиты ipconfig. Заполните таблицу:

Имя хоста	
IP-адрес	
Маска подсети	
Основной шлюз	
Используется ли DHCP (адрес DHCP-сервера)	
Описание адаптера	
Физический адрес сетевого адаптера	
Адрес DNS-сервера	
Адрес WINS-сервера	

Задание 4. Тестирование связи с помощью утилиты ping.

1. Проверьте правильность установки и конфигурирования TCP/IP на локальном компьютере.
2. Проверьте функционирование основного шлюза, послав 5 эхо-пакетов длиной 64 байта.
3. Проверьте возможность установления соединения с удаленным хостом.
4. С помощью команды ping проверьте адреса (взять из списка локальных ресурсов на сайте aspu.ru) и для каждого из них отметьте время отклика. Попробуйте изменить параметры команды ping таким образом, чтобы увеличилось время отклика. Определите IP-адреса узлов.

Задание 5. Определение пути IP-пакета.

С помощью команды tracert проверьте для перечисленных ниже адресов, через какие промежуточные узлы идет сигнал. Изучите ключи команды.

- a) aspu.ru
- b) mathmod.aspu.ru
- c) yarus.aspu.ru

Задание 6: Просмотр ARP-кэша.

С помощью утилиты `arp` просмотрите ARP-таблицу локального компьютера. Внести в кэш локального компьютера любую статическую запись.

Задание 7: Просмотр локальной таблицы маршрутизации.

С помощью утилиты `route` просмотреть локальную таблицу маршрутизации.

Задание 8. Получение информации о текущих сетевых соединениях и протоколах стека TCP/IP.

С помощью утилиты `netstat` выведите перечень сетевых соединений и статистическую информацию для протоколов UDP, TCP, ICMP, IP.

Контрольные вопросы:

1. Раскрыть термины: хост, шлюз, хоп, время жизни пакета, маршрут, маска сети, авторитетный/неавторитетный (компетентный) DNS-сервер, порт TCP, петля обратной связи, время отклика.
2. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP?
3. Каким образом команда `ping` проверяет соединение с удаленным хостом?
4. Каково назначение протокола ARP?
5. Как утилита `ping` разрешает имена узлов в `ip`-адреса (и наоборот)?
6. Какие могут быть причины неудачного завершения `ping` и `tracert`? (превышен интервал ожидания для запроса, сеть недоступна, превышен срок жизни при передаче пакета).
7. Всегда ли можно узнать символьное имя узла по его `ip`-адресу?
8. Какой тип записи запрашивает у DNS-сервера простейшая форма `nslookup`?

Практическая работа №4 «Решение проблем с TCP/IP» Цель: обобщение и

систематизация знаний по теме «Межсетевое взаимодействие»

Задания к работе

1. Открыть окно командной строки, ввести команду `ping` с IP адресом машины, при взаимодействии с которой возникают проблемы. Определить, использует ли проблемная машина конфигурацию статичного или динамичного IP адреса. Для этого откройте панель управления и выберите опцию Сетевые подключения. Теперь правой клавишей нажмите на подключении, которое собираетесь диагностировать, затем выберите опцию Свойства в появившемся меню быстрого доступа.
2. Перейдите по спискам элементов, используемых подключением, пока не дойдете до TCP/IP протокола (выбран на рисунке 3). Выберите этот протокол, нажмите на кнопке Свойства, чтобы открыть страницу свойств для Internet Protocol (TCP/IP).
3. Запишите IP конфигурацию машины. Особенно важно сделать заметки следующих элементов:
 - а. Использует ли машина статичную или динамичную конфигурацию?
 - б. Если используется статичная конфигурация, запишите значение IP адреса, маски подсети и основного шлюза?
 - в. Получает ли машина адрес DNS сервера автоматически?
 - г. Если адрес DNS сервера вводится вручную, то какой адрес используется?
4. Если на компьютере установлено несколько сетевых адаптеров, то в панели управления будут перечислены несколько сетевых подключений.
5. Проверьте тип адаптера.
6. Определите, принимает ли Windows такую конфигурацию. Для этого откройте окно командной строки и введите следующую команду: `IPCONFIG /ALL`.
7. Определите правильный сетевой адаптер. В этом случае определение нужного адаптера довольно простое, поскольку в списке есть всего лишь один адаптер.
8. Отправьте `ping` запрос на адрес локального узла. Существует два различных способа того, как это сделать. Одним способом является ввод команды: `PING LOCALHOST`.
9. Введите команду `Nslookup`, за которой должно идти полное доменное имя удаленного узла. Команда `Nslookup` должна суметь разрешить полное доменное имя в IP адрес.
11. Необходимо просканировать клиентскую машину на предмет вредоносного ПО. Если на машине не обнаружено вредоносного ПО, сбросьте DNS кэш путем ввода следующей команды: `IPCONFIG /FLUSHDNS`.

Контрольные вопросы

1. Поясните, что может означать, если время TTL закончилось до получения ответа.
2. Как подтвердить наличие сетевого соединения?
3. Что показывает команда `IPCONFIG /ALL`?
4. Что означает наличие IP адрес со значением 0.0.0.0.?
5. С помощью какой команды можно проверить то, что конфигурация IP адреса работает корректно, и что отсутствуют проблемы с стеком локального протокола TCP/IP?
6. Как производится опрос основного шлюза?
7. Как производится опрос DNS сервера?

Практическая работа №5 «Преобразование форматов IP-адресов. Расчет IP-адреса и маски подсети»

Цель работы: определение класса и расчет IP-адреса и маски подсети.

Общие положения

IP-адрес представляет собой 32-разрядное двоичное число, разделенное на группы по 8 бит, называемых *октетами*.

Наиболее распространенной формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в *десятичной форме* и разделенных точками, например: 128.10.2.30

Этот же адрес может быть представлен в *двоичном формате*: 10000000 00001010 00000010 00011110.

А также в *шестнадцатеричном формате*: 80.0A.02.1D

Следует заметить, что максимальное значение октета равно 11111111 (двоичная система счисления), что соответствует в десятичной системе 255.

Поэтому IP-адреса, в которых хотя бы один октет превышает это число, являются недействительными. Пример: 172.16.123.1 – действительный адрес, 172.16.123.256 – несуществующий адрес, поскольку 256 выходит за пределы допустимого диапазона.

IP-адрес состоит из двух логических частей – **номера подсети (ID подсети)** и **номера узла (ID хоста)** в этой подсети. При передаче пакета из одной подсети в другую используется ID подсети. Когда пакет попал в подсеть назначения, ID хоста указывает на конкретный узел в рамках этой подсети.

Чтобы записать ID подсети, в поле номера узла в IP-адресе ставят нули. Чтобы записать ID хоста, в поле номера подсети ставят нули. Например, если в IP-адресе 172.16.123.1 первые два байта отводятся под номер подсети, остальные два байта – под номер узла, то номера записываются следующим образом:

ID подсети: 172.16.0.0.

ID хоста: 0.0.123.1.

По числу разрядов, отводимых для представления номера узла (или номера подсети), можно определить общее количество узлов (или подсетей) по простому правилу: если число разрядов для представления номера узла равно N , то общее количество узлов равно $2^N - 2$. Два узла вычитаются

вследствие того, что адреса со всеми разрядами, равными нулям или единицам, являются особыми и используются в специальных целях.

Например, если под номер узла в некоторой подсети отводится два байта (16 бит), то общее количество узлов в такой подсети равно $2^{16} - 2 = 65534$ узлов.

Для определения того, какая часть IP-адреса отвечает за ID подсети, а какая за ID хоста, применяются два способа:

- 1) с помощью классов
- 2) с помощью масок.

Общее правило: под ID подсети отводятся *первые* несколько бит IP-адреса, оставшиеся биты обозначают ID хоста.

Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса.

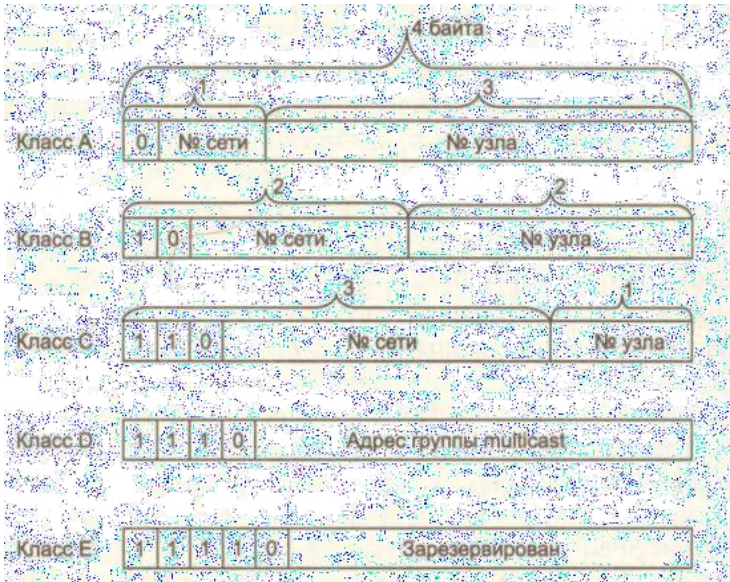


Таблица - Классы IP-адресов

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Количество сетей	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	126	$2^{24} - 2 = 16777214$
B	10	128.0.0.0	191.255.0.0	16384	$2^{16} - 2 = 65534$
C	110	192.0.1.0	223.255.255.0	2097152	$2^8 - 2 = 254$
D	1110	224.0.0.0	239.255.255.255	Групповой адрес	
E	11110	240.0.0.0	247.255.255.255	Зарезервирован	

Адреса **класса А** предназначены для использования в больших сетях общего пользования.

Они допускают большое количество номеров узлов.

Адреса **класса В** используются в сетях среднего размера, например, сетях университетов и крупных компаний.

Адреса класса С используются в сетях с небольшим числом компьютеров.

Адреса класса D используются при обращениях к группам машин.

Адреса класса E зарезервированы на будущее.

Некоторые IP-адреса являются особыми, они не должны применяться для идентификации обычных сетей:

- Если все биты IP-адреса равны нулю, адрес обозначает узел-отправитель и используется в некоторых сообщениях ICMP.
- Если все биты ID сети равны 1, адрес называется *ограниченным широковещательным (limited broadcast)*, пакеты, направленные по такому адресу, рассылаются всем узлам той подсети, в которой находится отправитель пакета.
- Если все биты ID хоста равны 1, адрес называется *широковещательным (broadcast)*, пакеты, имеющие широковещательный адрес, доставляются всем узлам подсети назначения.
- Если все биты ID хоста равны 0, адрес считается идентификатором подсети (subnet ID).

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является *внутренним адресом стека протоколов* компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети. В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со значения 127. Когда программа посылает данные по IP-адресу 127.x.x.x, то данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера, как только что принятые. Маршрут перемещения данных образует «петлю», поэтому этот адрес называется *адресом обратной петли (loopback)*.

Форма *группового IP-адреса - multicast* - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Групповой адрес не делится на номера сети и узла и обрабатывается маршрутизатором особым образом. Основное назначение групповых адресов - распространение информации по схеме «один ко многим». Основное назначение multicast-адресов - распространение информации по схеме «один-ко-многим». Хост, который хочет передавать одну и ту же информацию многим абонентам,

с помощью специального протокола IGMP (Internet Group Manageme Protocol) сообщает о создании в сети новой мультивещательной группы с определенным адресом. Маршрутизаторы, поддерживающие мультивещательность, распространяют информацию о создании новой группы во всех, подключенных к портам этого маршрутизатора. Хосты, которые хотят присоединиться к вновь создаваемой мультивещательной группе, сообщают об этом своим локальным маршрутизаторам и те передают эту информацию хосту, инициатору создания новой группы. Групповая адресация предназначена для экономичного распространения в Internet или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей.

Маска - число, которое служит для выделения частей IP-адреса, чтобы TCP/IP мог отличать номер сети от номера хоста. Используя маску подсети, TCP/IP-хосты могут связаться и определить, где находится хост назначения: в локальной или удаленной сети. Пример маски подсети: 255.255.255.0.

Биты IP-адреса, определяющие номер IP-сети, в маске подсети должны быть равны 1, а биты, определяющие номер узла, в маске подсети должны быть равны 0. Для стандартных классов сетей маски имеют следующие значения:

-класс А - 11111111. 00000000. 00000000. 00000000 (255.0.0.0); -
класс В - 11111111.11111111. 00000000. 00000000 (255.255.0.0);
-класс С-11111111.11111111.11111111. 00000000 (255.255.255.0).

Маски подсетей могут использоваться для маскирования тех частей адреса, которые согласно структуре класса, определяются как адреса сети. На практике разделение на подсети применяется в случае, когда конкретное сетевое адресное пространство разбивается дальше на отдельные подсети.

Подсети являются удобным средством структуризации сетей в рамках одной организации, когда все адресное пространство сети internet может быть разделено на непересекающиеся подпространства - "*подсети*", с каждой из которых можно работать как с обычной сетью TCP/IP. Таким образом единая IP-сеть организации может строиться как объединение подсетей. При этом организация должна получить один сетевой номер.

Выполнение работыЗадание 1. Изучить теоретические основы

IP-адресации

1. Сколько октетов в IP — адресе?
2. Сколько битов в октете?
3. Сколько бит в маске подсети?

Задание 2. Определить IP адрес вашего ПК

Узнайте собственный *IP адрес* компьютера и определите, к какому классу он относится.

Узнать свой собственный *IP адрес* вы можете, если запустите в ОС Windows XP на выполнение команду **Пуск – Программы – Стандартные – Командная Строка** и наберете в ней **ipconfig**.

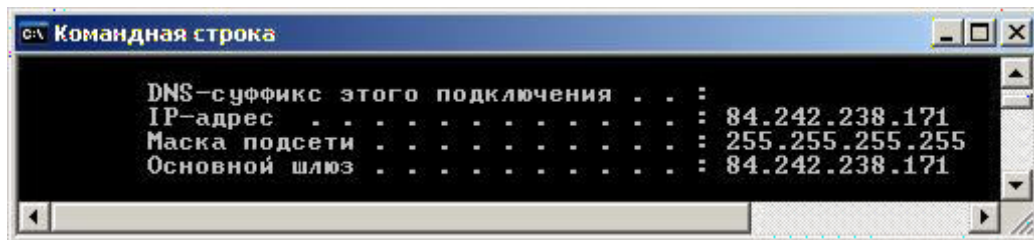


Рис. IP адрес вашего ПК в десятичной системе счисления

Задание 3. Переведите следующие двоичные числа в десятичные, а десятичные в двоичные.

Двоичное значение	Десятичное значение	Десятичное значение	Двоичное значение
10101100.00101000.00000000.00000000		127.1.1.1	
01011110.01110111.10011111.00000000		109.128.255.254	
10010001.0110000.10000000.00011001		131.107.2.89	
01111111.00000000.00000000.00000001		129.46.78.0	

Задание 4. Определение частей IP- адресов.

Заполнить таблицу об идентификации различных классов IP-адресов.

IP- адреса хостов	Класс адреса	Адрес сети	Адреса хостов	Широковещательный (broadcast) адрес	Маска подсети по умолчанию
216.14.55.137					
123.1.1.15					
150.127.221.244					
194.125.35.199					
175.12.239.244					

Задание 5. Дан IP- адрес 142.226.0.15

1. Чему равен двоичный эквивалент второго октета?

2. Какому классу принадлежит этот адрес?
3. Чему равен адрес сети, в которой находится хост с этим адресом?
4. Является ли этот адрес хоста допустимым в классической схеме адресации?

Задание 6

Найти адрес сети, минимальный IP, максимальный IP и число хостов по IP-адресу и маске сети:

IP-адрес: 192.168.215.89

Маска: 255.255.255.0

Задание 7

Найти маску сети, минимальный IP, максимальный IP по IP-адресу и адресу сети:

IP-адрес: 124.165.101.45

Сеть: 124.128.0.0

Задание 8

Найти минимальный IP, максимальный IP по адресу сети и маске:

Маска: 255.255.192.0

Сеть: 92.151.0.0

Задание 9. Определите, какие IP-адреса не могут быть назначены узлам. Объясните, почему такие IP-адреса не являются корректными.

1. 131.107.256.80
2. 222.222.255.222
3. 31.200.1.1
4. 126.1.0.0
5. 190.7.2.0
6. 127.1.1.1
7. 198.121.254.255
8. 255.255.255.255