

**ГАОУ ВО «ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
НАРОДНОГО ХОЗЯЙСТВА»  
БИЗНЕС-КОЛЛЕДЖ**

*Утвержден решением  
Ученого совета ДГУНХ,  
протокол № 13 от 06 июля 2020 г.*

**КАФЕДРА «ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАЦИОННЫЕ  
ТЕХНОЛОГИИ»**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ  
«РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ ОТРАСЛЕВОЙ НАПРАВЛЕННОСТИ»**

СПЕЦИАЛЬНОСТЬ – 09.02.05 ПРИКЛАДНАЯ ИНФОРМАТИКА  
(по отраслям)  
УРОВЕНЬ ОБРАЗОВАНИЯ – СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ

Квалификация – техник-программист

**Махачкала – 2020**

УДК 004.056  
ББК 32.973

**Составитель** – Саидов Шамил Рабазанович, старший преподаватель кафедры «Прикладная математика и информационные технологии» ДГУНХ.

**Внутренний рецензент** – Раджабов Карахан Якубович, кандидат экономических наук, доцент кафедры «Прикладная математика и информационные технологии», начальник департамента факультета информационных технологий ДГУНХ.

**Внешний рецензент** – Меджидов Зияудин Гаджиевич, кандидат физико-математических наук, старший научный сотрудник Отдела математики и информатики Дагестанского научного центра Российской Академии Наук

**Представитель работодателя** – Мухидинов Юнус Гудович, операционный директор ООО «Крон».

*Фонд оценочных средств по междисциплинарному курсу «Разработка, внедрение и адаптация программного обеспечения отраслевой направленности» разработана в соответствии с требованиями федерального государственного образовательного стандарта СПО 09.02.05 Прикладная информатика (по отраслям), утвержденного приказом Министерства образования и науки Российской Федерации от 13 августа 2014 г. № 1001*

Фонд оценочных средств по междисциплинарному курсу «Разработка, внедрение и адаптация программного обеспечения отраслевой направленности» размещена на официальном сайте [www.dgunh.ru](http://www.dgunh.ru)

Саидов Ш.Р. Фонд оценочных средств по междисциплинарному курсу «Разработка, внедрение и адаптация программного обеспечения отраслевой направленности» для СПО 09.02.05 Прикладная информатика (по отраслям)– Махачкала: ДГУНХ, 2020. – 63 с.

Рекомендован к утверждению Учебно-методическим советом ДГУНХ 03 июля 2020 г.

Рекомендован к утверждению руководителем образовательной программы СПО – программы подготовки специалистов среднего звена по специальности 09.02.05 Прикладная информатика (по отраслям), к.э.н. Гереевой Т.Р.

Одобен на заседании кафедры «Прикладная математика и информационные технологии» 30 июня 2020 г., протокол № 11.

## Назначение фонда оценочных средств

Фонд оценочных средств (ФОС) составляется в соответствии с требованиями ФГОС СПО для проведения промежуточной аттестации обучающихся по междисциплинарному курсу «Разработка, внедрение и адаптация программного обеспечения отраслевой направленности» на соответствие их учебных достижений поэтапным требованиям соответствующей Программы подготовки специалистов среднего звена (ППССЗ).

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по междисциплинарному курсу «Разработка, внедрение и адаптация программного обеспечения отраслевой направленности» включает в себя: перечень компетенций с указанием этапов их формирования в процессе освоения ППССЗ; описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания; типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения ППССЗ; методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Фонд оценочных средств сформирован на основе ключевых принципов оценивания:

- валидности: объекты оценки должны соответствовать поставленным целям обучения;
- надежности: использование единообразных стандартов и критериев для оценивания достижений;
- объективности: разные студенты должны иметь равные возможности добиться успеха.

Основными параметрами и свойствами ФОС являются:

- предметная направленность (соответствие предмету изучения конкретной учебной дисциплины);
- содержание (состав и взаимосвязь структурных единиц, образующих содержание теоретической и практической составляющих учебной дисциплины);
- объем (количественный состав оценочных средств, входящих в ФОС);
- качество оценочных средств и ФОС в целом, обеспечивающее получение объективных и достоверных результатов при проведении контроля с различными целями.

# I. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

## 1.1 Перечень формируемых компетенций

код компетенции	формулировка компетенции
<b>ОК</b>	<b>ОБЩЕКУЛЬТУРНЫЕ КОМПЕТЕНЦИИ</b>
ОК-1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес
ОК-2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК-3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК-4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК-5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК-6	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
ОК-7	Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.
ОК-8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК-9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
<b>ПК</b>	<b>ПРОФЕССИОНАЛЬНЫЕ КОМПЕТЕНЦИИ</b>
ПК-2.1	Осуществлять сбор и анализ информации для определения потребностей клиента.
ПК-2.2	Разрабатывать и публиковать программное обеспечение и информационные ресурсы отраслевой направленности со статическим и динамическим контентом на основе готовых спецификаций и стандартов.

ПК-2.3	Проводить отладку и тестирование программного обеспечения отраслевой направленности.
ПК-2.4	Проводить адаптацию отраслевого программного обеспечения.
ПК-2.5	Разрабатывать и вести проектную и техническую документацию.
ПК-2.6	Участвовать в измерении и контроле качества продуктов.

## КОМПОНЕНТНЫЙ СОСТАВ КОМПЕТЕНЦИЙ

код и формулировка компетенции	компонентный состав компетенции		
	знает:	умеет:	владеет:
<b>ОК-1:</b> Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	<b>З1</b> - социальную значимость профессиональной деятельности; <b>З2</b> - перспективы развития в профессиональной сфере; <b>З3</b> - положительные и отрицательные стороны профессии; <b>З4</b> - ближайшие и конечные жизненные цели в проф. деятельности;	<b>У1-</b> аргументировать свой выбор в профессиональном самоопределении; <b>У2-</b> выполнять самоанализ профессиональной пригодности; <b>У3-</b> определить пути реализации жизненных планов; <b>У4-</b> определить перспективы трудоустройства	<b>В1-</b> основными видами деятельности на рабочем месте и необходимыми инструментами труда.
<b>ОК-2:</b> Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и	<b>З1</b> - основные методы и способы решения профессиональных задач;	<b>У1-</b> оценивать эффективность и качество выполнения работ по профессии;	<b>В1-</b> методами работать в команде и самостоятельно

качество.			
<b>ОК-3:</b> Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	<b>З1-</b> возможные последствия несоблюдения технологических процессов и производственных инструкций подчиненными работниками (персоналом);	<b>У1-</b> вести документацию установленного образца, соблюдать сроки ее заполнения и условия хранения;	<b>В1-</b> навыками решения стандартных и нестандартных профессиональных задач в области организации процесса производства
<b>ОК-4:</b> Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	<b>З1-</b> виды источников информации для профессиональной деятельности;	<b>У1-</b> организовывать эффективный поиск необходимой информации;	<b>В1-</b> навыками использования различных источников, включая электронные;
<b>ОК-5:</b> Использовать информационно-коммуникационные технологии в профессиональной деятельности.	<b>З1-</b> оборудование и инвентарь, используемое в области организации процесса производства;	<b>У1-</b> пользоваться необходимым оборудованием и инвентарем;	<b>В1-</b> навыками безопасного использования новейшего оборудования;
<b>ОК-6:</b> Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.	<b>З1-</b> основные принципы эффективного общения;	<b>У1-</b> взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения;	<b>В1-</b> приемами эффективного общения;
<b>ОК-7:</b> Брать на себя	<b>З1-</b> приемы самоанализа и	<b>У1-</b> аргументировать	<b>В1-</b> навыками публичной и

ответственность за работу членов команды (подчиненных), результат выполнения заданий.	коррекции результатов собственной работы;	собственную позицию и отношение к конкретным ситуациям в профессиональной деятельности;	научной речи;
<b>ОК-8:</b> Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	<b>З1-</b> приемы планирования самостоятельной работы;	<b>У1-</b> организовывать самостоятельную работу при прохождении практики и написании дневника по практике;	<b>В1-</b> навыками работы с информацией
<b>ОК-9:</b> Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	<b>З1-</b> значение инноваций в области организации процесса производства;	<b>У1-</b> применять инновации в области организации процесса обслуживания потребителей;	<b>В1-</b> навыками отслеживания инноваций в профессиональной деятельности
<b>ПК-2.1:</b> Осуществлять сбор и анализ информации для определения потребностей клиента.	<b>З1-</b> модели процесса разработки программного обеспечения; <b>З2-</b> принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного	<b>У1-</b> использовать методы для получения кода с заданной функциональностью и степенью качества; <b>У1-</b> создавать программу по разработанному алгоритму как отдельный модуль;	<b>В1-</b> участия в выработке требований к программному обеспечению; <b>В2-</b> участия в проектировании программного обеспечения с использованием специализированных программных пакетов;

	обеспечения;		
<b>ПК-2.2:</b> Разрабатывать и публиковать программное обеспечение и информационные ресурсы отраслевой направленности со статическим и динамическим контентом на основе готовых спецификаций и стандартов.	<b>З1-</b> методы и средства разработки программной документации. <b>З2-</b> методы организации работы в коллективах разработчиков программного обеспечения; <b>З3-</b> принципы структурного и модульного программирования;	<b>У1-</b> работать в среде программирования <b>У2-</b> реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	<b>В1-</b> использования инструментальных средств на этапе отладки программного продукта; <b>В2-</b> разработки кода программного продукта на основе готовой спецификации на уровне модуля;
<b>ПК-2.3:</b> Проводить отладку и тестирование программного обеспечения отраслевой направленности.	<b>З1-</b> модели процесса разработки программного обеспечения; <b>З2-</b> современные стандарты в области программного обеспечения отраслевой направленности. <b>З2-</b> основные подходы к интегрированию программных модулей;	<b>У1-</b> владеть основными методологиями процессов разработки программного обеспечения;	<b>В1-</b> участия в выработке требований к программному обеспечению; <b>В2-</b> участия в проектировании программного обеспечения с использованием специализированных программных пакетов;
<b>ПК-2.4:</b> Проводить адаптацию отраслевого программного обеспечения.	<b>З1</b> – методы адаптации программного обеспечения отраслевой направленности.	<b>У1</b> - Проводить адаптацию отраслевого программного обеспечения.	<b>В1</b> – навыками адаптации отраслевого программного обеспечения.
<b>ПК-2.5:</b> Разрабатывать и вести проектную	<b>З1</b> – способы и ГОСТы разработки	<b>У1</b> – составлять технологическую документацию.	<b>В1</b> – навыками разработки технологической

и техническую документацию.	технологической документации		документации
<b>ПК-2.6:</b> Участвовать в измерениях и контроле качества продуктов.	<b>З1</b> - основные методы контроля качества продуктов.	<b>У1-</b> контролировать качество программных продуктов	<b>В1-</b> навыками работы с программными комплексами и контроля качества этих продуктов

## 1.2 ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

### Структура дисциплины:

№ темы	тема (раздел теоретического обучения) дисциплины
1.	Общая отраслевая терминология. Технологии сбора информации. Технология проведения анкетирования и интервьюирования. Идентифицирование, анализ и структурирование объектов информационного контента
2.	Основные понятия и подходы технологии программирования. Технология программирования и основные этапы ее развития. Автоматизация разработки программного продукта, CASE-технологии.
3.	Понятие алгоритма. Алгоритмизация. Способы записи алгоритмов. Блок-схемы.
4.	Средства разработки программного продукта. Язык программирования: назначение и основные особенности. Исходный код и машинный код.
5.	Основные типы и структуры данных. Основные алгоритмы. Основные понятия структур данных. Динамические структуры данных. Поиск. Сортировка.
6.	Архитектура программного обеспечения отраслевой направленности: 1.Технология программирования при структурном подходе. Спецификация ПП при структурном подходе. 2.Технология программирования при объектном подходе. Спецификация ПП при объектном подходе. Унифицированный язык моделирования UML.
7.	Распределенная обработка информации. Введение в Интернет. Проектирование сайта. Введение в web-дизайн.

8.	Язык сценариев JavaScript. Введение. Приемы программирования.
9.	Введение в язык программирования PHP. PHP и MySQL. Основные приемы программирования на PHP. Публикация сайта в сети Интернет.
10.	Компьютерные технологии представления и управления данными. Компьютерные данные: типы данных, Система информационной обработки данных, Введение данных, Системы управления базами данных в экономике.
11.	Эффективность и технологичность программного продукта.
12.	Организация сетевых технологий.
13.	Теория проектирования удаленных баз данных.
14.	Архитектура удаленных баз данных. Отличия и преимущества удаленных баз данных от локальных баз данных.
15.	Основные технологии доступа к данным и типовые элементы доступа. Основные понятия и место применения технологий COM, ODBC, MIDAS, MTS, CORBA
16.	Введение в работу с удаленными базами данных. Типы SQL серверов и их особенности. Настройка системных файлов для работы с удалённой базой данных. Организация сеанса связи с удалённой базой данных. Настройка приложения клиента для работы с удалённой базой данных.
17.	Физическая организация удалённой базы данных: простая и составная база данных, настройка страниц обмена, понятие фрагментации и дефрагментации, способы выполнения дефрагментации. Преимущества использования компонентов TQuery по сравнению с компонентами TTable.
18.	Проектирование серверной части приложения баз данных. Проектирование структуры базы данных с помощью команд.  Создание и перемещение файла базы данных. Команды по созданию, модификации и удалению таблицы. Создание, просмотр, перестройка и удаление индексов. Вывод на экран справки о структуре таблицы.
19.	Создание серверной части приложения: алиас, файл базы данных, таблицы.
20.	Визуальные средства проектирования структуры базы данных. Проектирование структуры базы данных: таблиц, индексов, установлению взаимосвязи, установлению условий ссылочной целостности, обеспечению достоверности данных. Просмотр и модификация созданной структуры базы данных.
21.	Проектирование клиентской части приложения баз данных. Компоненты доступа к данным. Запросы на выборку данных.

	Компоненты отображения данных выборки. Запросы на изменение данных в базе данных
22.	Хранимые процедуры, триггеры и генераторы. Обеспечение достоверности, целостности и непротиворечивости данных. Каскадные воздействия. Сортировка, поиск и фильтрация данных в базе данных и в выборках. Формирование и вывод отчетов.
23.	Управление транзакциями и кэширование памяти. Перехват исключительных ситуаций и обработка ошибок.
24.	Администрирование и эксплуатация удаленных баз данных. Установка привилегий доступа к данным. Копирование и перенос данных. Восстановление данных. Копирование клиентской и серверной части приложений баз данных.

**Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы**

код компетенции	Этапы формирования компетенций (темы дисциплин)												
	1	2	3	4	5	6	7	8	9	10	11	12	13
ОК-1	+			+		+	+		+	+	+	+	+
ОК-2	+			+			+	+	+				
ОК-3			+		+	+				+	+	+	+
ОК-4	+	+	+		+	+				+		+	+
ОК-5	+	+	+	+	+		+	+	+				
ОК-6				+					+	+	+		
ОК-7				+		+	+		+			+	+
ОК-8						+							
ОК-9								+					
ПК-2.1	+	+	+	+	+	+	+	+	+	+	+	+	+
ПК-2.2			+	+	+	+	+	+	+	+	+		
ПК-2.3			+	+	+	+	+	+	+	+	+		
ПК-2.4	+	+	+	+	+	+	+	+	+	+	+		
ПК-2.5					+	+	+	+	+	+	+		
ПК-2.6			+	+	+	+	+	+	+	+	+		

Итого	+	+	+	+	+	+	+	+	+	+	+	+	+
-------	---	---	---	---	---	---	---	---	---	---	---	---	---

## II. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

### 2.1 Структура фонда оценочных средств для текущего контроля и промежуточной аттестации

№ п/п	контролируемые разделы, темы дисциплины	код контролируемой компетенции или ее части	планируемые результаты обучения (знать, уметь, владеть), характеризующие этапы формирования компетенций	Наименование оценочного средства	
				текущий контроль	промежуточная аттестация
<b>Раздел 1</b>					
1	Решения задач на ЭВМ. Основные понятия алгоритмов	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b>ОК-1</b> Знать: З1,З2,З3,З4 Уметь: У1,У2,У3,У4 Владеть: В1 <b>ОК-2</b> Знать: З1 Уметь: У1 Владеть: В1 <b>ОК-3</b> Знать: З1 Уметь: У1 Владеть: В1 <b>ОК-4</b> Знать: З1 Уметь: У1 Владеть: В1 <b>ОК-5</b> Знать: З1 Уметь: У1 Владеть: В1	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			<p><b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2</p> <p><b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2</p> <p><b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2,</p> <p><b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1</p>		
2	<p>Языки программирования и их назначение. Основные понятия языка программирования ВР</p>	<p>ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3</p>	<p><b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1</p> <p><b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-3</u></b> Знать: 31</p>	<p>-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

		ПК-2.4 ПК-2.5 ПК-2.6	Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1		
3	Простые типы	ОК-1 ОК-2	<b><u>ОК-1</u></b> Знать:	-Тестовые задания;	Экзаменационные вопросы

	<p>данных языка ВР. Совместимость типов</p>	<p>ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p>31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1</p>	<p>-вопросы для обсуждения; презентации; - лабораторные работы</p>	
--	---	---	--	--	--

			<p><b><u>ОК-2.5</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1</p>		
4	Программирование циклических алгоритмов	<p>ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p><b><u>ОК-1</u></b> Знать: З1,З2,З3,З4 Уметь: У1,У2,У3,У4 Владеть: В1</p> <p><b><u>ОК-2</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-3</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1</p> <p><b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть: В1,В2</p> <p><b><u>ПК-2.2</u></b> Знать: З1,З2,З3,</p>	<p>-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

			<p>Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1</p>		
5	Процедуры и функции, их использование в программах	<p>ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p><b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1</p>	<p>-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы</p>	Экзаменационные вопросы

			<p><b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2</p> <p><b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2</p> <p><b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2,</p> <p><b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1</p>		
6	Модули, структура модулей, особенности работы модулями	с ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<p><b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1</p> <p><b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-6</u></b> Знать: 31</p>	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			<p>Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1</p>		
<b>Раздел 2</b>					
7	<p>Основные понятия. Правила построения простейших HTML- документов.</p>	<p>ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4</p>	<p><b><u>ОК-1</u></b> Знать: З1,З2,З3,З4 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: З1 Уметь: У1</p>	<p>-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

		ПК-2.5 ПК-2.6	<p>Владеть: В1 <b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1</p>		
8	Добавление изображений и	ОК-1 ОК-2 ОК-2	<p><b><u>ОК-1</u></b> Знать: З1,З2,З3,З4</p>	-Тестовые задания; -вопросы для	Экзаменационные вопросы

	<p>мультимедиа на web-страницы.</p>	<p>ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p>Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b></p>	<p>обсуждения; презентации; - лабораторные работы</p>	
--	-------------------------------------	--	---	---	--

			Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1		
9	Оформление веб-страниц с использованием каскадных таблиц стилей.	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			<p>Владеть: В1, В2  <b><u>ПК-2.3</u></b>  Знать: 31, 32, 33,  Уметь: У1,  Владеть: В1, В2,  <b><u>ОК-2.4</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-2.5</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-2.6</u></b>  Знать: 31  Уметь: У1  Владеть: В1</p>		
10	<p>Основы языка JavaScript:  Переменные и типы данных, условные операторы и операторы циклов, функции и объекты.</p>	<p>ОК-1  ОК-2  ОК-2  ОК-4  ОК-5  ОК-6  ОК-7  ОК-8  ОК-9  ПК-2.1  ПК-2.2  ПК-2.3  ПК-2.4  ПК-2.5  ПК-2.6</p>	<p><b><u>ОК-1</u></b>  Знать:  31, 32, 33, 34  Уметь:  У1, У2, У3, У4  Владеть: В1  <b><u>ОК-2</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-3</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-4</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-5</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-6</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-7</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-8</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-9</u></b></p>	<p>-Тестовые задания;  -вопросы для обсуждения;  презентации;  - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

			Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1		
11	Особенности языка РНР. Синтаксис, грамматика функции РНР.	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b><u>ОК-1</u></b> Знать: З1,З2,З3,З4 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			<p>Владеть: В1  <b><u>ОК-7</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-8</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-9</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ПК-2.1</u></b>  Знать: З1,З2,  Уметь: У1,У2,  Владеть:В1,В2  <b><u>ПК-2.2</u></b>  Знать: З1,З2,З3,  Уметь: У1,У2  Владеть:В1,В2  <b><u>ПК-2.3</u></b>  Знать: З1,З2,З3,  Уметь: У1,  Владеть:В1,В2,  <b><u>ОК-2.4</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-2.5</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-2.6</u></b>  Знать: З1  Уметь: У1  Владеть: В1</p>		
12	Работа с классами в РНР.	<p>ОК-1  ОК-2  ОК-2  ОК-4  ОК-5  ОК-6  ОК-7  ОК-8  ОК-9  ПК-2.1  ПК-2.2  ПК-2.3  ПК-2.4  ПК-2.5  ПК-2.6</p>	<p><b><u>ОК-1</u></b>  Знать:  З1,З2,З3,З4  Уметь:  У1,У2,У3,У4  Владеть: В1  <b><u>ОК-2</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-3</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-4</u></b></p>	<p>-Тестовые задания;  -вопросы для обсуждения;  презентации;  - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

			Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1		
<b>Раздел 3</b>					
13	Базы данных – основные понятия.	ОК-1 ОК-2 ОК-2 ОК-4	<b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь:	-Тестовые задания; -вопросы для обсуждения;	Экзаменационные вопросы

		<p>ОК-5 У1,У2,У3,У4</p> <p>ОК-6 Владеть: В1</p> <p>ОК-7 <b><u>ОК-2</u></b></p> <p>ОК-8 Знать: З1</p> <p>ОК-9 Уметь: У1</p> <p>ПК-2.1 Владеть: В1</p> <p>ПК-2.2 <b><u>ОК-3</u></b></p> <p>ПК-2.3 Знать: З1</p> <p>ПК-2.4 Уметь: У1</p> <p>ПК-2.5 Владеть: В1</p> <p>ПК-2.6 <b><u>ОК-4</u></b></p> <p>Знать: З1</p> <p>Уметь: У1</p> <p>Владеть: В1</p> <p><b><u>ОК-5</u></b></p> <p>Знать: З1</p> <p>Уметь: У1</p> <p>Владеть: В1</p> <p><b><u>ОК-6</u></b></p> <p>Знать: З1</p> <p>Уметь: У1</p> <p>Владеть: В1</p> <p><b><u>ОК-7</u></b></p> <p>Знать: З1</p> <p>Уметь: У1</p> <p>Владеть: В1</p> <p><b><u>ОК-8</u></b></p> <p>Знать: З1</p> <p>Уметь: У1</p> <p>Владеть: В1</p> <p><b><u>ОК-9</u></b></p> <p>Знать: З1</p> <p>Уметь: У1</p> <p>Владеть: В1</p> <p><b><u>ПК-2.1</u></b></p> <p>Знать: З1,З2,</p> <p>Уметь: У1,У2,</p> <p>Владеть:В1,В2</p> <p><b><u>ПК-2.2</u></b></p> <p>Знать: З1,З2,З3,</p> <p>Уметь: У1,У2</p> <p>Владеть:В1,В2</p> <p><b><u>ПК-2.3</u></b></p> <p>Знать: З1,З2,З3,</p> <p>Уметь: У1,</p> <p>Владеть:В1,В2,</p> <p><b><u>ОК-2.4</u></b></p> <p>Знать: З1</p> <p>Уметь: У1</p> <p>Владеть: В1</p> <p><b><u>ОК-2.5</u></b></p> <p>Знать: З1</p>	<p>презентации;</p> <p>- лабораторные работы</p>	
--	--	--	--	--

			Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1		
14	Системы управления базами данных (СУБД).	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			<p><b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1</p>		
15	Архитектура СУБД. Проектирование баз данных.	<p>ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p><b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31</p>	<p>-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

			<p>Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1</p>		
16	Языковые средства современных СУБД.	<p>ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p><b><u>ОК-1</u></b> Знать: З1,З2,З3,З4 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1</p>	<p>-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

			<p><b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2</p> <p><b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2</p> <p><b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2,</p> <p><b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1</p>		
17	Администрирование баз данных.	<p>ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p><b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1</p> <p><b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1</p> <p><b><u>ОК-4</u></b> Знать: 31</p>	<p>-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы</p>	<p>Экзаменационные вопросы</p>

			<p>Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: З1,З2, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: З1,З2,З3, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: З1,З2,З3, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: З1 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: З1 Уметь: У1 Владеть: В1</p>		
18	Хранилища данных.                      Банки данных.	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6	<p><b><u>ОК-1</u></b> Знать: З1,З2,З3,З4 Уметь: У1,У2,У3,У4 Владеть: В1</p>	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные	Экзаменационные вопросы

	<p>ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p><b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1</p>	<p>работы</p>	
--	---	--	---------------	--

			<b>OK-2.6</b> Знать: 31 Уметь: У1 Владеть: В1		
<b>Раздел 4</b>					
19	Архитектуры удаленных баз данных.	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b>OK-1</b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b>OK-2</b> Знать: 31 Уметь: У1 Владеть: В1 <b>OK-3</b> Знать: 31 Уметь: У1 Владеть: В1 <b>OK-4</b> Знать: 31 Уметь: У1 Владеть: В1 <b>OK-5</b> Знать: 31 Уметь: У1 Владеть: В1 <b>OK-6</b> Знать: 31 Уметь: У1 Владеть: В1 <b>OK-7</b> Знать: 31 Уметь: У1 Владеть: В1 <b>OK-8</b> Знать: 31 Уметь: У1 Владеть: В1 <b>OK-9</b> Знать: 31 Уметь: У1 Владеть: В1 <b>ПК-2.1</b> Знать: 31,32, Уметь: У1,У2, Владеть: В1,В2 <b>ПК-2.2</b> Знать: 31,32,33, Уметь: У1,У2 Владеть: В1,В2 <b>ПК-2.3</b>	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			Знать: 31,32,33, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1		
20	Принципы разработки эксплуатации систем управления удаленными базами данных.	и ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31 Уметь: У1	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			<p>Владеть: В1  <b><u>ПК-2.1</u></b>  Знать: 31,32,  Уметь: У1,У2,  Владеть:В1,В2  <b><u>ПК-2.2</u></b>  Знать: 31,32,33,  Уметь: У1,У2  Владеть:В1,В2  <b><u>ПК-2.3</u></b>  Знать: 31,32,33,  Уметь: У1,  Владеть:В1,В2,  <b><u>ОК-2.4</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-2.5</u></b>  Знать: 31  Уметь: У1  Владеть: В1  <b><u>ОК-2.6</u></b>  Знать: 31  Уметь: У1  Владеть: В1</p>		
21	Технологии разработки и управления базами данных средствами языка SQL.	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b>	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b> Знать: 31 Уметь: У1 Владеть: В1		
22	Технологии доступа к удаленным базам данных.	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			<p>Владеть: В1  <b><u>ОК-5</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-6</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-7</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-8</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-9</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ПК-2.1</u></b>  Знать: З1,З2,  Уметь: У1,У2,  Владеть:В1,В2  <b><u>ПК-2.2</u></b>  Знать: З1,З2,З3,  Уметь: У1,У2  Владеть:В1,В2  <b><u>ПК-2.3</u></b>  Знать: З1,З2,З3,  Уметь: У1,  Владеть:В1,В2,  <b><u>ОК-2.4</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-2.5</u></b>  Знать: З1  Уметь: У1  Владеть: В1  <b><u>ОК-2.6</u></b>  Знать: З1  Уметь: У1  Владеть: В1</p>		
23	Восстановление данных критических ситуациях. В	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7	<p><b><u>ОК-1</u></b>  Знать:  З1,З2,З3,З4  Уметь:  У1,У2,У3,У4  Владеть: В1  <b><u>ОК-2</u></b></p>	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

	<p>ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6</p>	<p>Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть:В1,В2, <b><u>ОК-2.4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-2.6</u></b></p>	
--	--	--	--

			Знать: 31 Уметь: У1 Владеть: В1		
24	Объектно-ориентированные СУБД. Системы баз данных, основанные на правилах. Многопользовательские системы управления жизненным циклом продукции.	ОК-1 ОК-2 ОК-2 ОК-4 ОК-5 ОК-6 ОК-7 ОК-8 ОК-9 ПК-2.1 ПК-2.2 ПК-2.3 ПК-2.4 ПК-2.5 ПК-2.6	<b><u>ОК-1</u></b> Знать: 31,32,33,34 Уметь: У1,У2,У3,У4 Владеть: В1 <b><u>ОК-2</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-3</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-4</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-5</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-6</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-7</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-8</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ОК-9</u></b> Знать: 31 Уметь: У1 Владеть: В1 <b><u>ПК-2.1</u></b> Знать: 31,32, Уметь: У1,У2, Владеть:В1,В2 <b><u>ПК-2.2</u></b> Знать: 31,32,33, Уметь: У1,У2 Владеть:В1,В2 <b><u>ПК-2.3</u></b> Знать: 31,32,33, Уметь: У1, Владеть: В1,	-Тестовые задания; -вопросы для обсуждения; презентации; - лабораторные работы	Экзаменационные вопросы

			Владеть: В1, В2, <b>ОК-2.4</b> Знать: З1 Уметь: У1 Владеть: В1 <b>ОК-2.5</b> Знать: З1 Уметь: У1 Владеть: В1 <b>ОК-2.6</b> Знать: З1 Уметь: У1 Владеть: В1		
--	--	--	--	--	--

## 2.2 КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ ПО ВИДАМ ОЦЕНОЧНЫХ СРЕДСТВ

### ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

№ п/п	наименование оценочного средства	характеристика оценочного средства	Представление оценочного средства в фонде
<b>УСТНЫЕ ОЦЕНОЧНЫЕ СРЕДСТВА</b>			
1	собеседование, устный опрос	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по темам/разделам дисциплины
2	Коллоквиум	Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования преподавателя с обучающимися.	Вопросы по темам/разделам дисциплины
3.	Доклад, сообщение	Продукт самостоятельной работы	Темы докладов,

		обучающегося, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы	сообщений
<b>ПИСЬМЕННЫЕ ОЦЕНОЧНЫЕ СРЕДСТВА</b>			
4	Реферат	Продукт самостоятельной работы аспиранта, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.	Темы рефератов
5	Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
6	Проект	Конечный продукт, получаемый в результате планирования и выполнения комплекса учебных и исследовательских заданий. Позволяет оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских	Темы групповых и/или индивидуальных проектов

		навыков, навыков практического и творческого мышления. Может выполняться в индивидуальном порядке или группой обучающихся.	
7	Контрольная работа	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	комплект контрольных заданий по вариантам
8	Курсовая работа	Продукт самостоятельной работы обучающегося, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной учебно-исследовательской темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а так же собственные взгляды на нее.	Темы курсовых работ
	Курсовой проект	Курсовым проектом является письменная работа, выполняющаяся на протяжении семестра и содержащая анализ варианта экономического или инженерного решения по теме, заданной в заглавии самого курсового проекта. Любой курсовой проект является строго индивидуальным и ориентированным на развитие у студента профессиональных навыков, а также умению творчески подходить к решению практических задач, которые относятся к выбранному направлению подготовки. Курсовой проект обязательно должен состоять	Темы курсовых проектов

		из расчетной (графической) и текстовой части. В текстовую часть обязательно входит объяснительная записка, которая заполняется не только теоретическими подсчётами, но и проведёнными вычислениями и расчётами. Графическая часть включает в себя схемы, таблицы и чертежи.	
9	Лабораторная работа	Средство для закрепления и практического освоения материала по определенному разделу	Комплект лабораторных заданий
10	Задача	Это средство раскрытия связи между данными и искомым, заданные условием задачи, на основе чего надо выбрать, а затем выполнить действия, в том числе арифметические, и дать ответ на вопрос задачи.	задания по задачам
11	Расчетно-графическая работа	Средство проверки умений применять полученные знания по заранее определенной тематике для решения задач или заданий по модулю или дисциплине в целом.	комплект заданий для выполнения расчетно-графической работы
	.....		

#### **А) КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ ОТВЕТОВ НА УСТНЫЕ ВОПРОСЫ**

№ п/п	критерии оценивания	количество баллов	оценка/зачет
1.	1) полно и аргументированно отвечает по содержанию задания; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на	10	отлично

	практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные; 3) излагает материал последовательно и правильно.		
2.	студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.	8	хорошо
3.	ставится, если студент обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки.	5	удовлетворительно
4.	студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.	0	неудовлетворительно

## **Б) КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ**

№ п/п	тестовые нормы:% правильных ответов	количество баллов
1	90-100 %	9-10
2	80-89%	7-8
3	70-79%	5-6
4	60-69%	3-4

5	50-59%	1-2
6	менее 50%	0

## **В) КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ РЕШЕНИЯ ЗАДАЧ**

№ п/п	критерии оценивания	количество баллов
1	Полное верное решение. В логическом рассуждении и решении нет ошибок, задача решена рациональным способом. Получен правильный ответ. Ясно описан способ решения.	9-10
2	Верное решение, но имеются небольшие недочеты, в целом не влияющие на решение, такие как небольшие логические пропуски, не связанные с основной идеей решения. Решение оформлено не вполне аккуратно, но это не мешает пониманию решения.	7-8
3	Решение в целом верное. В логическом рассуждении и решении нет существенных ошибок, но задача решена неоптимальным способом или допущено не более двух незначительных ошибок. В работе присутствуют арифметическая ошибка, механическая ошибка или описка при переписывании выкладок или ответа, не исказившие экономическое содержание ответа.	5-6
4	В логическом рассуждении и решении нет ошибок, но допущена существенная ошибка в математических расчетах. При объяснении сложного экономического явления указаны не все существенные факторы.	3-4
5	Имеются существенные ошибки в логическом рассуждении и в решении. Рассчитанное значение искомой величины искажает экономическое содержание ответа. Доказаны вспомогательные утверждения, помогающие в решении задачи.	2-3

6	Рассмотрены отдельные случаи при отсутствии решения. Отсутствует окончательный численный ответ (если он предусмотрен в задаче). Правильный ответ угадан, а выстроенное под него решение - безосновательно.	1
7	Решение неверное или отсутствует.	0

### Г) КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ РЕФЕРАТОВ

№ п/п	критерии оценивания	количество баллов
1	выполнены все требования к написанию и защите реферата: обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы.	<i>9-10 баллов</i>
2	основные требования к реферату и его защите выполнены, но при этом допущены недочеты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объем реферата; имеются упущения в оформлении; на дополнительные вопросы при защите даны неполные ответы.	<i>7-8 баллов</i>
3	имеются существенные отступления от требований к реферированию. В частности: тема освещена лишь частично; допущены фактические ошибки в содержании реферата или при ответе на дополнительные вопросы.	<i>4-6 баллов</i>
4	тема освоена лишь частично; допущены грубые ошибки в содержании реферата или при ответе на дополнительные вопросы; во время защиты отсутствует вывод.	<i>1-3 баллов</i>

5	тема реферата не раскрыта, обнаруживается существенное непонимание проблемы.	0 баллов
---	--	----------

#### Д) КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ ДОМАШНЕГО ЗАДАНИЯ

№ п/п	критерии оценивания	количество баллов
1	Задание выполнено полностью: цель домашнего задания успешно достигнута; основные понятия выделены; наличие схем, графическое выделение особо значимой информации; работа выполнена в полном объёме.	9-10
2	Задание выполнено: цель выполнения домашнего задания достигнута; наличие правильных эталонных ответов; однако работа выполнена не в полном объёме.	8-7
3	Задание выполнено частично: цель выполнения домашнего задания достигнута не полностью; многочисленные ошибки снижают качество выполненной работы.	6-5
4	Задание не выполнено, цель выполнения домашнего задания не достигнута.	менее 5

#### Е) КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ КОНТРОЛЬНЫХ РАБОТ

№ п/п	критерии оценивания	количество баллов	оценка
1	исключительные знания, абсолютное понимание сути вопросов, безукоризненное знание основных понятий и положений, логически и лексически грамотно изложенные, содержательные, аргументированные и исчерпывающие ответы	19-20	
2	глубокие знания материала, отличное понимание сути вопросов, твердое знание основных понятий и положений по вопросам, структурированные,	17-18	

	последовательные, полные, правильные ответы		
3	глубокие знания материала, правильное понимание сути вопросов, знание основных понятий и положений по вопросам, содержательные, полные и конкретные ответ на вопросы. Наличие несущественных или технических ошибок	15-16	
4	твердые, достаточно полные знания, хорошее понимание сути вопросов, правильные ответы на вопросы, минимальное количество неточностей, небрежное оформление	13-14	
5	твердые, но недостаточно полные знания, по сути верное понимание вопросов, в целом правильные ответы на вопросы, наличие неточностей, небрежное оформление	11-12	
6	общие знания, недостаточное понимание сути вопросов, наличие большого числа неточностей, небрежное оформление	9-10	
7	относительные знания, наличие ошибок, небрежное оформление	7-8	
8	поверхностные знания, наличие грубых ошибок, отсутствие логики изложения материала	5-6	
9	непонимание сути, большое количество грубых ошибок, отсутствие логики изложения материала	3-4	
10	не дан ответ на поставленные вопросы	1-2	
11	отсутствие ответа, дан ответ на другие вопросы, списывание в ходе выполнения работы, наличие на рабочем месте технических средств, в том числе телефона	0	

### Ж) КРИТЕРИИ И ШКАЛА ОЦЕНКИ ПРЕЗЕНТАЦИЙ

№ п/	критерии оценки	максимальное количество баллов
------	-----------------	--------------------------------

п		
1	титульный слайд с заголовком	5
2	дизайн слайдов	10
3	использование дополнительных эффектов (смена слайдов, звук, графика, анимация)	5
4	список источников информации	5
5	широта кругозора	5
6	логика изложения материала	10
7	текст хорошо написан и сформированные идеи ясно изложены и структурированы	10
8	слайды представлены в логической последовательности	5
9	грамотное создание и сохранение документов в папке рабочих материалов	5
10	слайды распечатаны в форме заметок	5
	средняя оценка:	

### **3) КРИТЕРИИ и ШКАЛА ОЦЕНИВАНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.**

№ п/п	критерии оценки	максимальное количество баллов
1	работа выполнена полностью, использован правильный, оптимальный алгоритм решения; работа выполнена по плану и сделаны правильные выводы;	12-15
2	работа выполнена правильно с учетом 2-3 несущественных ошибок исправленных самостоятельно по требованию преподавателя.	7-11
3	работа выполнена правильно не менее чем на половину или допущена существенная ошибка	4-7
4	допущены существенные ошибки в ходе работы, которые студент не может исправить даже по требованию преподавателя.	1-3
5	работа не выполнена.	0

### **III ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСОВЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

## **Перечень вопросов для устного обсуждения по разделам:**

### **Введение.**

1. Общая отраслевая терминология.
2. Технологии сбора информации.
3. Технология проведения анкетирования и интервьюирования.  
Идентифицирование, анализ и структурирование объектов информационного контента.
4. Основные понятия и подходы технологии программирования.
5. Технология программирования и основные этапы ее развития.
6. Автоматизация разработки программного продукта, CASE-технологии.

### **Раздел 1. Основы алгоритмизации и программирования.**

1. Понятие алгоритма.
2. Алгоритмизация.
3. Способы записи алгоритмов.
4. Блок-схемы.
5. Средства разработки программного продукта.
6. Язык программирования: назначение и основные особенности.
7. Исходный код и машинный код.
8. Основные типы и структуры данных.
9. Основные алгоритмы.
10. Основные понятия структур данных.
11. Динамические структуры данных.
12. Поиск. Сортировка.

**Тема 1: «Решения задач на ЭВМ. Основные понятия алгоритмов»**

**Тема 2: «Языки программирования и их назначение. Основные понятия языка программирования ВР»**

### **Задание 1. Перечень контрольных вопросов по разделу для устного обсуждения**

1. Алгоритмы. Составление алгоритмов для машины Поста.
2. Способы представления алгоритмов.
3. Базовые алгоритмические структуры: следование, ветвление, цикл.
4. Работа с программой "Интерпретатор машины Поста и машины Тьюринга"

### **Задание 2. Комплект тематик для рефератов**

Написать реферат на одну из предложенных тем:

<i>№</i>	<i>Тематика рефератов</i>	<i>План</i>
1.	Создание классов и компонентов с помощью	1. Введение 2. Создание классов и компонентов с

	ModelMaker.	помощью ModelMaker. 3. Заключение. 4. Список использованной литературы.
2	Пакеты компонентов в среде Delphi.	1. Введение 2. Пакеты компонентов в среде Delphi. 3. Заключение. 4. Список использованной литературы.
3	Технология доступа к данным ADO.	1. Введение 2. Технология доступа к данным ADO. 3. Заключение. 4. Список использованной литературы.
4	Работа с библиотекой классов среды визуального программирования Delphi.	1. Введение 2. Работа с библиотекой классов среды визуального программирования Delphi. 3. Заключение. 4. Список использованной литературы.
5	Инструментальные средства для работы с базами данных.	1. Введение 2. Инструментальные средства для работы с базами данных. 3. Заключение. 4. Список использованной литературы.

### Задание 3. Лабораторная работа

Лабораторная работа на тему: Алгоритмы. Составление алгоритмов для машины Поста.

**Тема 3: «Простые типы данных языка ВР. Совместимость типов».**

**Тема 4: «Условные и безусловные конструкции языка ВР»**

#### Задание 1. Перечень контрольных вопросов по разделу:

1. История появления и развития языка Pascal.
2. Ссылочные типы. Работа с указателями в Pascal.
3. Использование динамических переменных в языке Pascal.
4. Работа с окнами в текстовом режиме работы экрана.
5. Программирование клавиатуры и звукового генератора.

#### Задание 2. Лабораторная работа

Лабораторная работа на тему: Интегрированная среда ВР.

#### Задание 3. Комплект тематик для рефератов

Написать реферат на одну из предложенных тем:

<i>№№</i>	<i>Тематика рефератов</i>	<i>План</i>
1.	Ссылочные типы. Работа с	1. Введение

	указателями в Pascal.	2. Ссылочные типы. Работа с указателями в Pascal.ModelMaker. 3. Заключение. 4. Список использованной литературы.
2	Использование динамических переменных в языке Pascal.	1. Введение 2. Использование динамических переменных в языке Pascal. 3. Заключение. 4. Список использованной литературы.
3	Работа с окнами в текстовом режиме работы экрана.	1. Введение 2. Работа с окнами в текстовом режиме работы экрана. 3. Заключение. 4. Список использованной литературы.
4	Программирование клавиатуры и звукового генератора.	1. Введение 2. Программирование клавиатуры и звукового генератора. 3. Заключение. 4. Список использованной литературы.
5	Работа с фрагментами изображения средствами языка Pascal.	1. Введение 2. Работа с фрагментами изображения средствами языка Pascal. 3. Заключение. 4. Список использованной литературы.

**Тема 5: Программирование циклических алгоритмов”.**

**Тема 6: Структурированные типы данных языка ВР. Массивы.**

**Тема 7: Записи, множества, работа с ними**

**Задание 1. Перечень контрольных вопросов по разделу:**

1. Работа с фрагментами изображения средствами языка Pascal.
2. Обращение к функциям ОС.
3. Поддержка процедур обработки прерываний в Pascal.
4. Директивы компилятора, подробная детализация и использование в приложениях.
5. Работа с указателями, процедурные и вариантыные типы данных.

**Задание 2. Лабораторная работа**

Лабораторная работа на тему: Условия.

**Задание 3. Комплект тематик для рефератов**

Написать реферат на одну из предложенных тем:

<i>№№</i>	<i>Тематика рефератов</i>	<i>План</i>
1.	Обращение к функциям ОС.	1. Введение

		2. Обращение к функциям ОС. 3. Заключение. 4. Список использованной литературы.
2	Поддержка процедур обработки прерываний в Pascal.	1. Введение 2. Поддержка процедур обработки прерываний в Pascal. 3. Заключение. 4. Список использованной литературы.
3	Директивы компилятора, подробная детализация и использование в приложениях.	1. Введение 2. Директивы компилятора, подробная детализация и использование в приложениях. 3. Заключение. 4. Список использованной литературы.
4	Работа с указателями, процедурные и вариантыные типы данных.	1. Введение 2. Работа с указателями, процедурные и вариантыные типы данных. 3. Заключение. 4. Список использованной литературы.
5	Создание и использование собственных модулей.	1. Введение 2. Создание и использование собственных модулей 3. Заключение. 4. Список использованной литературы.

**Тема 8: «Обработка символов и строк»**

**Тема 9: Процедуры и функции, их использование в программах**

**Тема 10: «Организация ввода-вывода данных. Работа с файлами».**

**Задание 1. Перечень вопросов по разделу для устного обсуждения**

1. Понятие криптографии шифра.
2. Задачи и методы криптографии.
3. Виды шифров.
4. Криптографические примитивы.
5. Основные криптографические протоколы.
6. Электронная цифровая подпись..
7. Модели основных криптоаналитических атак.

**Задание 2. Лабораторная работа**

Лабораторная работа на тему: Использование графических возможностей языка ВР.

**Задание 3. Комплект тематик для рефератов**

Написать реферат на одну из предложенных тем:

<i>№</i>	<i>Тематика рефератов</i>	<i>План</i>
1.	Создание пользовательских	1. Введение

	классов – методика, апробация, применение.	2. Создание пользовательских классов – методика, апробация, применение. 3. Заключение. 4. Список использованной литературы.
2	Профессиональная настройка среды Delphi.	1. Введение 2. Профессиональная настройка среды Delphi. 3. Заключение. 4. Список использованной литературы.
3	Работа с классами в среде визуального программирования.	1. Введение 2. Работа с классами в среде визуального программирования. 3. Заключение. 4. Список использованной литературы.
4	Обработка исключительных ситуаций, защита ресурсов и регенерация исключений.	1. Введение 2. Обработка исключительных ситуаций, защита ресурсов и регенерация исключений. 3. Заключение. 4. Список использованной литературы.
5	Создание SDI и MDI – приложений.	1. Введение 2. Создание SDI и MDI – приложений. 3. Заключение. 4. Список использованной литературы.

## **Тема 11: «Код программы, данные, адреса»**

### **Задание 1. Перечень контрольных вопросов по разделу:**

1. Создание SDI и MDI –приложений.
2. Освоение мало используемых компонент Палитры компонент среды Delphi.

### **Задание 2. Комплект тематик для презентаций**

Создать презентацию на одну из предложенных тем:

1. Перенос приложений из среды Windows в среду Linux.
2. Перенос приложений баз данных и Web – приложений
3. Работа с сокетами.
4. Компоненты IntraWeb, создание приложений с помощью IntraWeb.
5. Введение в создание компонентов.

### **Задание 3. Лабораторная работа**

Лабораторная работа на тему: Операторы циклов.

**Тема 12: «Модули, структура модулей, особенности работы с модулями».**

**Тема 13: «Графические возможности ВР».**

**Тема 14: «Методы проектирования программ»**

**Задание 1. Перечень контрольных вопросов по разделу:**

1. Работа со связанными таблицами в рамках баз данных.
2. Формирование отчетов в среде Delphi.
3. Применение языка SQL для создания запросов к базе данных.
4. Использование XML файлов при работе в среде Delphi.
5. Разработка Web-приложений.

**Задание 3. Лабораторная работа**

Лабораторная работа на тему: Структурированные типы данных: массивы.

**Задание 2. Комплект тематик для рефератов**

Написать реферат на одну из предложенных тем:

<i>№</i>	<i>Тематика рефератов</i>	<i>План</i>
1.	Работа со связанными таблицами в рамках баз данных.	1. Введение 2. Работа со связанными таблицами в рамках баз данных. 3. Заключение. 4. Список использованной литературы.
2	Формирование отчетов в среде Delphi.	1. Введение 2. Формирование отчетов в среде Delphi. 3. Заключение. 4. Список использованной литературы.
3	Применение языка SQL для создания запросов к базе данных.	1. Введение 2. Применение языка SQL для создания запросов к базе данных. 3. Заключение. 4. Список использованной литературы.
4	Использование XML файлов при работе в среде Delphi.	1. Введение 2. Использование XML файлов при работе в среде Delphi. 3. Заключение. 4. Список использованной литературы.
5	Разработка Web-приложений.	1. Введение 2. Разработка Web-приложений. 3. Заключение. 4. Список использованной литературы.

## Задание 2. Тесты по разделу.

**Выбрать из предложенных вариантов один правильный:**

1. Свойство алгоритмов, означающие, что результат выполнения алгоритма не зависит от исполнителя, а определяется только входными данными и шагами:
  - a) Результативность;
  - b) Детерминированность;
  - c) Дискретность;
  - d) Определенность.
2. Способ описания алгоритма основанный на записи содержания выполняемых действий с использованием изобразительных возможностей языка математики:
  - a) Словесный;
  - b) Формально-словесный;
  - c) Графический;
  - d) Программный.
3. Что из ниже перечисленного не входит в набор основных символов Турбо Паскаля?
  - a) латинские строчные и прописные буквы
  - b) служебные слова
  - c) десять цифр
  - d) русские строчные и прописные буквы
  - e) знак подчеркивание
4. Какого раздела не существует в программе, написанной на языке Паскаль?
  - a) Заголовка
  - b) Примечания
  - c) Описаний
  - d) Операторов
5. Какие из типов относятся к стандартным?
  - a) Целый
  - b) Интервальный
  - c) Символьный
  - d) Логический
  - e) Перечисляемый
6. Какое из следующих описаний переменных является ошибочным?
  - a) Operator: (plus, minus, times);
  - b) Matrix: array[1..10,1..10] of Real;
  - c) Done,Error: boolean;
  - d) I,J,K: integer;
7. Определить истинность составного высказывания: " $(2 \times 2 = 4 \text{ или } 3 \times 3 = 10)$  и  $(2 \times 2 = 5 \text{ или } 3 \times 3 = 9)$ "
  - a) Ложно
  - b) Истинно
  - c) не ложно и не истинно
  - d) не истинно

8. Какой оператор не относится к группе операторов ввода-вывода языка Паскаль ?
- Read(A1,A2,...AK);
  - WriteLn(A1,A2,...AK);
  - PrintLn;
  - ReadLn;
9. Какой оператор определяет равенство двух значений?
- =
  - ==
  - :=
  - :=
10. Зарезервированное слово div обозначает операцию
- левого сдвига
  - целочисленного деления
  - правого сдвига
  - нахождения остатка от деления
11. Установите соответствие между типами данных и их обозначениями

1. символьный	A. boolean
2. логический	B. integer
3. целый	C. real
4. вещественный	D. char

- 1A, 2D, 3B, 4C
  - 1D, 2B, 3C, 4A
  - 1C, 2A, 3D, 4B
  - 1D, 2A, 3B, 4C
12. В каком из условных операторов допущена синтаксическая ошибка?
- if B = 0 then WriteLn('Деление на нуль невозможно.');
  - if a > b then max := a else max := b;
  - if (a > b) and (b > 0) then c := a + b;
  - if a < b then min := a; else min := b.
13. В результате выполнения фрагмента программы  
a := 2; b := 7; c := -8;  
if a + c > b then c := c + a else c := c - a;  
переменная c принимает значение:
- 6
  - 10
  - 5
  - 1
14. В результате выполнения фрагмента программы  
read(a, b);  
if b < a then c := b else c := a;  
write(c);  
на печать будет выведено
- произведение чисел a и b, если a положительно, b отлично от нуля

- b) меньшее из чисел a и b
  - c) сумма чисел a и b, если они оба положительны
  - d) большее из чисел a и b
15. В операторе цикла с предусловием тело цикла
- a) обязательно выполняется более 1 раза
  - b) выполняется по крайней мере 1 раз
  - c) может не выполниться вообще
  - d) выполняется строго определенное число раз
16. В операторе цикла с постусловием тело цикла
- a) обязательно выполняется более 1 раза
  - b) выполняется по крайней мере 1 раз
  - c) может не выполниться вообще
  - d) выполняется строго определенное число раз
17. В операторе цикла с параметром тело цикла
- a) обязательно выполняется более 1 раза
  - b) выполняется по крайней мере 1 раз
  - c) может не выполниться вообще
  - d) выполняется строго определенное число раз
18. Установить соответствие

Оператор цикла	
1. с постусловием	A. for i:= 1 to N do ...
2. с предусловием	B. repeat ... until
3. с параметром	C. while ... do

- a) 1A, 2B, 3C
  - b) 3A, 2B, 1C
  - c) 1B, 2A, 3C
  - d) 1B, 2C, 3A
19. Какие файловые типы данных поддерживает Турбо Паскаль?
- a) Текстовые
  - b) Графические
  - c) Компонентные
  - d) Системные
  - e) Бестиповые
  - f) Архивные
20. Установить правильную последовательность строк для фрагмента программы, в

$$\sum_{i=1}^9 \frac{i+1}{i^2} \sum_{i=1}^9 \frac{i+1}{i^2}$$

котором вычисляется сумма

- 1 s:= s + n;
- 2 begin
- 3 writeln(s);
- 4 for i:= 1 to 9 do
- 5 n:= (i + 1)/(i\*i);

- 6 s:= 0; n:= 0;
- 7 end;
- a) 2145673
- b) 2645317
- c) 6425173
- d) 2645137

21. Что такое массив?
- a) Ограниченная упорядоченная совокупность однотипных величин
  - b) Ограниченная совокупность различных элементов
  - c) Совокупность ограниченного числа логически связанных компонент, принадлежащих к разным типам
22. К структурированным типам относятся
- 1) вещественный      2) файл      3) множество
  - 4) символьный      5) диапазон      6) целый
- a) 2, 3, 5
  - b) 1, 6
  - c) 3, 5
  - d) 2, 3
23. Установите соответствие между типами данных и обозначающими их зарезервированными словами

1. строковый	A. array
2. массив	B. set
3. запись	C. string
4. множество	D. record

- a) 1C, 2B, 3A, 4D
  - b) 1C, 2A, 3D, 4B
  - c) 1C, 2D, 3A, 4B
  - d) 1D, 2A, 3B, 4C
24. Выберите все верные утверждения.  
Количество элементов, входящих в множество,
- 1) равно 256
  - 2) может меняться от 0 до 256
  - 3) не превосходит 255
  - 4) может быть равно нулю
- a) 1
  - b) 3,4
  - c) 2,4
  - d) 3
25. Укажите номер строки, в которой правильно описаны массивы
- 1 a, b: array [-2..8] of integer;
  - 2 m: array [20..1] of char;
  - 3 day: array [1..'n'] of char;
  - 4 ch: array ['A'..'T'] of byte;

- 5 letter: array ['m'..'b'] of integer;  
6 ord: array ['b'..'m'] of real;
- a) 1, 3, 5
  - b) 1, 4, 6
  - c) 1, 2
  - d) 4, 6

## **Раздел 2. Основы web-программирования.**

1. Распределенная обработка информации. Введение в Интернет. Проектирование сайта. Введение в web-дизайн.
2. Язык сценариев JavaScript. Введение. Приемы программирования.
3. Основные понятия Cookies. Использование сессий.
4. Компоненты ASP: ADO.
5. Компоненты ASP: File Access, Browser Capabilities, .Ad Rotator, Content Linking.
6. Доступ к базам данных из ASP: DSN, файлы включения.
7. Доступ к базам данных из ASP: RecordSet, его методы и свойства.
8. Межплатформенный язык запросов SQL.
9. PERL. Синтаксис языка. Переменные, регулярные выражения, операторы, функции, подпрограммы и модули.
10. CGI-программирование. Доступ к базам данных.
11. Продвижение веб-сайта в Интернете: интернет-маркетинг
12. Основные правила и требования современного веб-дизайна

### ***Тема и развернутый план лабораторных занятий***

#### ***Тема 1. Структура HTML-документа.***

- 1. Язык разметки гипертекста HTML*
- 2. Разделы HTML, HEAD, Body*

### *3. Создание простейшей веб-страницы*

## **Тема 2. Форматирование текста в HTML**

- 1. Вывод и оформление текста. Физическое и логическое форматирование*
- 2. Заголовки в HTML (H1, H2, ... H6)*
- 3. Создание абзацев в HTML-документах*

## **Тема 3. Списки в HTML-документах**

- 1. Создание маркированных и нумерованных списков (<OL>, <UL>, <LI>)*
- 2. Списки определения*
- 3. Вложенные списки в HTML*

## **Тема 4. Создание ссылок средствами HTML**

- 1. Элемент A и его атрибуты*
- 2. Локальные и внешние гиперссылки*

## **Тема 5. Добавление изображений и мультимедиа на веб-страницу средствами HTML**

- 1. Тег <IMG> и его атрибуты*
- 2. Вставка фоновых изображений и фоновой музыки на веб-страницу*
- 3. Создание проигрывателя средствами языка HTML*

## **Тема 6. Создание таблиц средствами языка HTML**

- 1. Элемент <TABLE> и его атрибуты. Создание строк и столбцов в таблице*
- 2. Объединение ячеек в таблице – атрибуты Rowspan и Colspan*
- 3. Создание сложных таблиц*

## **Тема 7. Создание фреймов в HTML**

- 1. Элемент <FRAMESET> и его атрибуты*
- 2. Создание ссылок внутри фреймов*
- 3. Создание плавающих фреймов*

## **Тема 8. Создание форм в языке HTML**

- 1. Элемент FORM и его атрибуты*
- 2. Элемент TEXTAREA и его атрибуты*

3. Создание списков опций (*SELECT, OPTION*)
4. Элемент *INPUT* и его атрибуты. Создание полей формы с помощью *<INPUT>*

### **Тема 9. Карта графических ссылок**

1. Характеристика основных тегов и атрибутов
2. Создание карты графических ссылок

### **Тема 10. Введение в таблицы стилей и язык CSS**

1. Форматирование текста в CSS
2. Структурное форматирование в CSS
3. Свойства для форматирования таблиц в CSS
4. Оформление пользовательских форм в CSS

### **Тема 11. Основы языка JavaScript**

1. Встраивание JavaScript-кода в HTML-документ
2. Написание простейшего сценария на языке JavaScript

### **Тема 12. Встроенные объекты в JavaScript**

1. Работа со встроенными объектами и событиями в JavaScript
2. Объект *EVENT*
3. Интерактивность на веб-страницах

### **Тема 13. Введение в PHP**

1. Установка программного обеспечения
2. Полезные PHP-функции
3. Разработка собственных функций
4. Работа с базой данных *MySQL*

**Задания для итогового контроля остаточных знаний по дисциплине**

#### **Задачи для проверки остаточных знаний**

1. Составьте Web-страницу, в рамках которой продемонстрируете умение работать с различными шрифтами с применением языка HTML.
2. Составьте Web-страницу, в рамках которой продемонстрируете умение

- работать с различными абзацами, цитатами, цветовой гаммой шрифтов с применением языка HTML.
3. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать с различными таблицами с применением языка HTML.
  4. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать с формами с применением языка HTML.
  5. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать с картой графических ссылок с применением языка HTML.
  6. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать с элементами диалоговых форм с применением языка HTML.
  7. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать с графическими изображениями с применением языка HTML.
  8. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать с видео-файлами с применением языка HTML.
  9. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать со звуковыми файлами с применением языка HTML.
  10. Составьте Web-страницу, в рамках которой продемонстрируйте умение работать с гиперссылками.
  11. Составьте Web-страницу, в рамках которой продемонстрируйте умение составлять сценарии на языке JavaScript в рамках HTML - документов.
  12. Составьте Web-страницу, в рамках которой продемонстрируйте умение создавать динамические изменяемые HTML – документы.
  13. Составьте Web-страницу, в рамках которой продемонстрируйте умение вводить данные в формы HTML – документов.
  14. Приведите примеры позиционирования средствами CSS.
  15. Составьте веб-страницу, приведя примеры всплывающих подсказок в CSS.
  16. Пр продемонстрируйте DHTML-анимация на веб-страницах.
  17. Приведите примеры перетаскивания объектов. События MouseDown, MouseMove и MouseUp.

18. Приведите примеры управления временем. Создание временных задержек.

### Материалы итогового тестирования.

1. Выберите атрибуты тэга <BODY>:
  - 1) BGCOLOR, TEXT, VLINK, ALINK, BGSOUND
  - 2) BGCOLOR, TEXT, VLINK, BLINK
  - 3) TEXT, COLOR, VLINK, ALINK
  
2. LEFTMARGIN является атрибутом тэга:
  - 1) <BODY>
  - 2) <IMG>
  - 3) <FORM>
  - 4) <FRAME>
  
3. Какой из следующих тэгов сдвигает текст выше уровня строки и выводит его шрифтом меньшего размера?
  - 1) <SUP>
  - 2) <SUB>
  - 3) <BIG>
  - 4) <STRIKE>
  
4. ALIGN является атрибутом следующих тэгов:
  - 1) <IMG>, <BODY>, <TABLE>, <P>
  - 2) <IMG>, <HR>, <TABLE>
  - 3) <H1>, <H2>, <FONT>
  - 4) <IMG>, <HR>, <FRAME> <FRAMESET>
  
5. Элемент логического форматирования <CITE> используется:
  - 1) для выделения цитат или названий книг, статей, при этом текст обычно выводится курсивом
  - 2) для ввода небольшого куска программного кода шрифтом фиксированной ширины
  - 3) для выделения важных элементов текста
  - 4) для выделения нескольких символов шрифтом фиксированной ширины
  
6. Атрибут ALIGN тэга <IMG> может принимать следующие значения
  - 1) left, right, center, top, middle, bottom, baseline
  - 2) left, right, center, border, bordercolor
  - 3) top, middle, bottom, baseline, width, height
  - 4) среди ответов нет верных
  
7. Какие из утверждений верные:

- А. Тэг <table> создает таблицу. Все прочие элементы таблицы должны быть вложенными в него. Допускается также вложение таблиц одна в другую, т.е. содержимым ячейки может быть другая таблица
- Б. Тэг <table> имеет следующие атрибуты: align (выравнивание таблицы относительно документа), border (толщина рамки в пикселах), cellpadding (задает расстояние между ячейками таблицы), cellspacing (задает расстояние между содержимым ячейки и ее рамкой)
- В. Тэг <table> имеет следующие атрибуты: align (выравнивание таблицы относительно документа), border (толщина рамки в пикселах), cellpadding (задает расстояние между содержимым ячейки и ее рамкой), cellspacing (задает расстояние между ячейками таблицы)

- 1) А, Б
- 2) А, В
- 3) А
- 4) Б
- 5) В

8. Выберите верное утверждение:

- А. атрибут <alt> тэга area может содержать поясняющий текст к данной активной области
- Б. карта графических ссылок начинается с тэга <img usemap=""#name"»
- В. карта графических ссылок начинается с тэга <MAP KART>
- Г. форма активной области устанавливается при помощи атрибута href

- 1) А
- 2) А, Б
- 3) А, Г
- 4) Нет верных утверждений

9. Какие из следующих утверждений верные:

- А. Основная функция Интернета состоит в установлении и обслуживании соединений подключенных к сети пользователей
- Б. Доменным адресом называется последовательность нескольких слов, разделенных точками.
- В. Крайнее правое слово доменного адреса называется именем корневого домена.

- 1) А
- 2) Б и В
- 3) А и Б
- 4) А, Б, В

10. Какой тэг помещает в форму раскрывающийся список?

- 1) <form select>
- 2) <select>
- 3) <input>

4) <textarea>

11. Какой атрибут тэга <select> позволяет производить выбор сразу нескольких опций:

- 1) name
- 2) many
- 3) selected
- 4) multiple

12. Гипертекст - это:

1. информационная оболочка
2. текст, содержащий иллюстрации
3. информация в виде документов, имеющих ссылки на другие документы
4. информационное хранилище

13. Для установки гиперссылки на ресурсы Internet, что указывается в качестве значения атрибута HREF?

1. полный указатель универсальных ресурсов
2. полное имя файла
3. адрес информационно-поисковой системы, осуществляющей поиск этого ресурса
4. Нет верных утверждений

14. Для уменьшения размера шрифта используется тэг:

1. Font size=+1
2. Small
3. Font size="1"
4. Font size="-"

15. HTML – это:

1. программа просмотра WWW-документов
2. прикладная программа (приложение)
3. язык разметки гипертекста
4. протокол взаимодействия клиент-сервер

16. Тег <body> используется для обозначения?

1. начала Web-страницы
2. начала нового параграфа
3. начала основной части Web-страницы
4. содержит наименование документа

17. Язык для составления скриптов, встраиваемых в web-страницы для придания им более привлекательного вида и динамичности?

1. VBScript
2. JavaScript
3. Java
4. Object Pascal

18. Для включения программ на JavaScript в HTML-страницу необходимы тэги:

1. `<SCRIPT> ... </SCRIPT>`
2. `<SCRIPT Java> ... </SCRIPT Java>`
3. `<Java> ... </Java>`
4. `< JavaScript> ... </ JavaScript>`

19. Выберите верное утверждение:

1. В JavaScript тип данных переменной не указывается
2. В JavaScript тип данных переменной обязательно должен быть указан
3. В JavaScript переменные нельзя использовать без предварительного описания?
4. Нет верных утверждений

20. Выберите верное утверждение:

1. В JavaScript тип данных переменной обязательно должен быть указан
2. В JavaScript переменные нельзя использовать без предварительного описания?
3. Все переменные JavaScript делятся на локальные и глобальные
4. Нет верных утверждений

21. Переменные JavaScript, которые видны только в рамках одной функции?

1. Глобальные
2. Локальные
3. Логические
4. Строковые

22. Переменные JavaScript, описанные в рамках какого-либо тэга `<SCRIPT> ... </SCRIPT>`, и видны для всех функций из любых тэгов `<SCRIPT> ... </SCRIPT>` на данной странице, и существуют, пока загружена страница?

1. Логические
2. Строковые
3. Локальные
4. Глобальные

- 23.Какая функция JavaScript преобразует текстовый тип данных в целочисленный?
1. IntToStr ( )
  2. ParseInt ( )
  3. ParseFloat ( )
  4. StrToInt ( )
- 24.Какая функция JavaScript преобразует текстовый тип данных в числовой с плавающей точкой?
1. IntToStr ( )
  2. ParseInt ( )
  3. ParseFloat ( )
  4. StrToInt ( )
- 25.Специальный объект, который отправляется в обработчик событий при возникновении любого события:
1. alert
  2. event
  3. prompt
  4. нет верных ответов
- 26.Метод объекта Window, т.е. объект Window, который может выдавать пользователю какое-либо предупреждение в окне сообщений:
1. alert
  2. prompt
  3. confirm
  4. event
- 27.При использовании, какого метода JavaScript, в появляющемся на экране окне пользователь может сделать выбор между кнопками ОК и Cancel?
1. event
  2. confirm
  3. prompt
  4. alert
- 28.Метод JavaScript, который дает пользователю возможность вписать собственный ответ на вопрос, вместо того чтобы просто выводить на экран информацию или делать выбор между кнопками ОК и Cancel:
1. alert
  2. event
  3. confirm
  4. prompt

29. Самой мощной особенностью объектно-ориентированного программирования является ...

1. наличие оболочки визуального программирования
2. наличие классов и механизм наследования классов
3. объединение данных и методов, которые оперируют этими данными
4. повторное использование исходного кода

30. Наследование – это ...

1. отношение между классами, при котором один использует структуру и поведение другого класса, с возможностью дальнейшего дополнения или переопределения
2. отношение между объектами, при котором один использует структуру и поведение другого объекта, с возможностью дальнейшего дополнения или переопределения
3. отношение между объектом и классом, при котором объект использует структуру и поведение класса, с возможностью дальнейшего дополнения или переопределения

31. Объект – это ...

1. структура данных
2. экземпляр некоторого класса
3. класс

32. Класс – это ...

1. тип данных
2. описание объектов
3. структура данных

33. TCP протокол ...

1. с образованием постоянного соединения гарантирует доставку каждого пакета
2. не использует постоянного соединения и не гарантирует доставку каждого пакета

34. UDP протокол ...

1. используется в сервисах, где требуется гарантированная доставка сообщения
2. используется там, где более важна скорость передачи сообщения

35. Какое из приведенных высказываний является ложным?

1. можно поменять порт, по которому работает Web-сервер
2. можно ограничить вход на Web-сервер по IP-адресам сетей и компьютеров,

3. можно поменять протокол, по которому работает Web-сервер
  4. можно ограничить вход на Web-сервер по именам и паролям пользователей
36. Что нужно сделать, чтобы на одном компьютере одновременно работало несколько Web-серверов?
1. можно поменять порт, по которому работает Web-сервер
  2. можно ограничить вход на Web-сервер по IP-адресам сетей и компьютеров,
  3. можно поменять протокол, по которому работает Web-сервер
  4. можно ограничить вход на Web-сервер по именам и паролям пользователей
37. Пополните следующий ряд: HTML, XML, ...
1. JSP
  2. Delphi
  3. C++
  4. XHTML
38. Пополните следующий ряд: ASP, PHP, PerlScript, Python, ...
1. JSP
  2. CSS
  3. C++
  4. XHTML
39. Пополните следующий ряд: ASP, PHP, Java-сервлеты, PerlScript...
1. JavaScript
  2. Java Server Pages
  3. JScript
  4. VBScript
40. Среда ASP (Active Server Pages) по умолчанию может содержать скрипты, написанные на VBScript и ... (Вообще, можно добиться, чтобы поддерживало PerlScript, PythonScript)
1. Java
  2. JScript
  3. Delphi
  4. CGI
41. Самым распространенным Web-браузером в настоящее время является ...
1. Netscape Navigator
  2. Microsoft Internet Explorer
  3. Opera

4. Mosaic

42. Первым появился ...

1. Netscape Navigator
2. Microsoft Internet Explorer
3. Opera
4. Mosaic

43. Последним появился ...

1. Netscape Navigator
2. Microsoft Internet Explorer
3. Opera
4. Mosaic

44. В настоящее время падает популярность ...

1. Netscape Navigator
2. Microsoft Internet Explorer
3. Opera
4. Mosaic

45. Что недоступно для программ на JavaScript?

1. Управление работой браузера
2. Управление содержимым Web-страницы
3. Работа с файлами на диске пользователя
4. Обмен данными с Web-сервером

46. Что Java-апплетам недоступно?

1. Управление работой браузера
2. Управление содержимым Web-страниц
3. Работа с файлами на диске пользователя
4. Обмен данными с Web-сервером

47. JavaScript ...

1. изначально позиционировалась как язык для создания Интернет-приложений, но сейчас – это полноценный язык программирования
2. остается языком расширения возможностей HTML в браузере, также PDF

48. Что такое SQL?

1. драйвер, который транслирует вызовы на языке Java, в особый формат, необходимый для работы сервера баз данных
2. система управления базами данных
3. средство создания динамических Web-страниц с синтаксисом Visual Basic

4. структурированный язык запросов для работы с базами данных

49. Что такое MYSQL?

1. драйвер, который транслирует вызовы на языке Java, в особый формат, необходимый для работы сервера баз данных
2. система управления базами данных
3. средство создания динамических Web-страниц с синтаксисом Visual Basic
4. структурированный язык запросов для работы с базами данных

50. Что такое ASP?

1. драйвер, который транслирует вызовы на языке Java, в особый формат, необходимый для работы сервера баз данных
2. система управления базами данных
3. средство создания динамических Web-страниц с синтаксисом Visual Basic
4. структурированный язык запросов для работы с базами данных

### **Задания для самостоятельной работы студентов**

#### **Перечень вопросов по дисциплине для самостоятельного изучения**

1. Основные понятия Cookies. Использование сессий.
2. Компоненты ASP: ADO.
3. Компоненты ASP: File Access, Browser Capabilities, .Ad Rotator, Content Linking.
4. Доступ к базам данных из ASP: DSN, файлы включения.
5. Доступ к базам данных из ASP: RecordSet, его методы и свойства.
6. Межплатформенный язык запросов SQL.
7. PERL. Синтаксис языка. Переменные, регулярные выражения, операторы, функции, подпрограммы и модули.
8. CGI-программирование. Доступ к базам данных.
9. Продвижение веб-сайта в Интернете: интернет-маркетинг
10. Основные правила и требования современного веб-дизайна

#### **Тематика рефератов и творческих работ студентов**

1. Требования к защите сайта от несанкционированного доступа
2. Создание или редактирование страниц сайта (разделов)

3. Основные правила и требования современного веб-дизайна
4. Подбор цвета при Web-дизайне и программировании
5. Безопасное web-программирование
6. Динамическое веб-программирование
7. Возможности CSS при создании веб-сайтов
8. Ссылки и изображения, карты изображений
9. Сетевые технологии и системное администрирование
10. Интерактивные элементы JavaScript на web-страницах
11. Размещение на web-страницах таблиц, форм и анимации
12. Обзор технологий, используемых для обеспечения безопасности
13. Применение основных правил веб-дизайна при создании сайта
14. Интерактивные эффекты на веб-страницах. Создание выпадающего меню
15. Веб-графика: оптимизация, создание интерактивных кнопок, GIF-анимация
16. Стандартные и нестандартные компоненты веб-сайта
17. Авторизация, идентификация и аутентификация пользователей
18. Разграничение прав доступа
19. Защита веб-сайтов от sql-injection и code-injection
20. Настройка PHP и Apache в целях повышения безопасности
21. Обмен данными: XML, RSS
22. Использование регулярных выражений
23. Импорт данных из неподготовленных источников (извлечение данных из HTML страниц других сайтов)
24. Подготовка веб-приложения к экспорту данных
25. Построение систем распределённой обработки данных на основе веб-приложений
26. Расширенное взаимодействие с пользователем
27. Доступ к заголовкам запроса, переменные окружения сервера
28. Проверка передаваемых значений

29. Передача файлов через формы
30. Использование cookies и сессий
31. Определение и установка типа передаваемых данных
32. Переадресация на другие страницы
33. Повторные запросы страницы
34. Сбор и анализ статистики. Системы сбора и анализа статистики
35. Взаимодействие веб-сайта с пользователями
36. Расширение возможностей веб-форм
37. Построение карты сайта, поиск по сайту
38. Улучшение навигации по сайту (быстрый переход, контекстно-зависимые меню и блоки ссылок)
39. Повышение интерактивности веб-сайта, использование JavaScript, Ajax
40. Общие вопросы работы с PHP, Apache, MySQL. Установка. Настройка.  
Выбор версий

### **Раздел 3. Базы данных.**

1. Компьютерные технологии представления и управления данными.
2. Компьютерные данные: типы данных, Система информационной обработки данных, Введение данных, Системы управления базами данных в экономике.
3. Эффективность и технологичность программного продукта.
4. Организация сетевых технологий.
5. Основные свойства баз данных.
6. Классификация баз данных.
7. Возможности СУБД.
8. Основные достоинства и недостатки СУБД.
9. Типовая структура современной СУБД.
10. Трёхуровневая архитектура данных.
11. Жизненный цикл базы данных.

**Задания для текущего контроля усвоения материала.**

### **Лабораторная работа 1. Создание однотабличной базы данных в среде Microsoft Visual FoxPro.**

#### ***Теоретическая часть***

При создании приложения используется проект, который объединяет элементы приложения **Visual FoxPro** и группирует их по типам. База данных в **Visual FoxPro** – это совокупность таблиц, отношений между таблицами, индексов, триггеров, хранимых процедур. База данных является частью проекта, поэтому её целесообразно создавать в окне проекта.

Структурными элементами базы данных являются:

- **Поле** – элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. Для описания поля используются следующие характеристики: имя, длина, тип и точность (для числовых данных). В структуре записи файла указываются поля, значения которых являются ключами: первичными и внешними.
- **Первичный ключ** - это одно или несколько полей, однозначно

идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей - составным ключом.

- **Внешний ключ** - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение внешнего ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по внешнему – несколько.
- **Файл (таблица)** – совокупность экземпляров записей одной структуры. Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики,
- **Запись** – совокупность логически связанных полей.
- **Экземпляр записи** - отдельная реализация записи, содержащая конкретные значения ее полей.

Таблицы составляют основу вашей базы данных. В них будет храниться вся необходимая информация. В дальнейшем данные в таблице будут дополняться новыми данными, редактироваться или исключаться из таблицы. Поля таблицы предназначены для хранения в них данных. Это могут быть числа, текстовая информация, даты, графические файлы и т.д. В **Visual FoxPro** допустимыми являются типы полей, указанные в таблице 1.

Таблица 1. Некоторые типы данных, используемых в СУБД **FoxPro**

Тип	Описание	Пример
Integer	Целые числа	9846
Numeric	Данные, с фиксированной точкой	3.1456
		-56.235

		“01/07/04”
Logical	Поля, содержащие только одно из двух возможных значений (да/нет)	True; False
Date	Дата	01/07/04
DateTime	Дата и время	01/07/04 12:30:00 pm
Memo	Очень длинный текст или комбинация текста и чисел	

Таблица 2. Некоторые команды СУБД FoxPro

Команда	Описание
CREATE	создание элементов БД (базы, таблиц, отчётов, запросов и т.д.)
MODIFY (STRUCTURE)	изменение элементов БД (базы, таблиц, отчётов, запросов и т.д.)
BROWSE, EDIT, CHANGE, APPEND	команды редактирования и просмотра содержимого таблиц
OPEN DATABASE	Открытие БД
CLOSE	закрытие элементов БД (базы, таблиц, отчётов, запросов и т.д.)
QUIT	Выход из Visual FoxPro

### Цель работы

1. Создать проект *Информационные системы*
2. Создать БД *Штат*
3. Создать в БД *Штат* таблицу *Сотрудники*
4. Заполнить таблицу данными

### Практическая часть

1. Создать в служебной папке *Мои документы* новую папку и присвоить ей имя, пример: *3курс4группаfoxlab* (указывать свой курс и группу)

2. Запустить программу **Microsoft Visual FoxPro:**  
**Пуск/программы/Visual FoxPro**
3. Ознакомиться с элементами рабочего окна программы

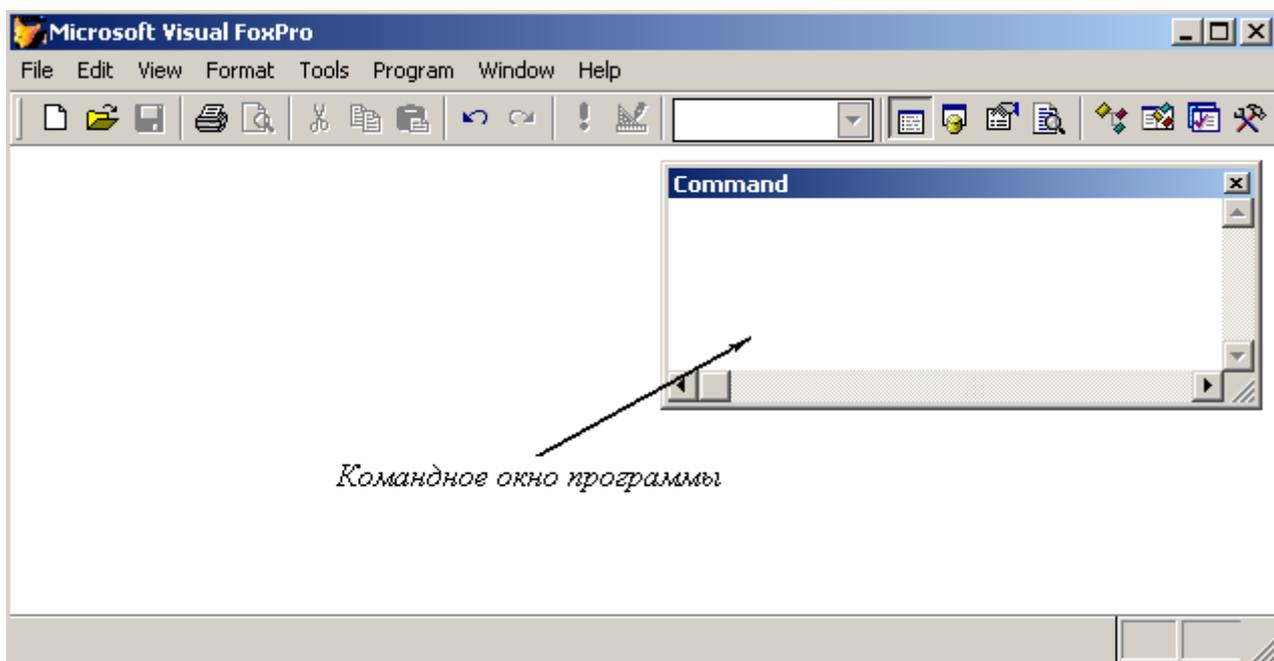


Рис.1. Главное окно **Visual FoxPro 8.0**

4. Создать новый проект: **File/New/Project/New File**, указать созданную нами ранее папку, присвоить имя проекту *Информационные системы* и сохранить. Все создаваемые в последующем элементы приложения будут храниться в проекте *Информационные системы*
5. БД создаётся аналогично: **File/New/Database/New file**, присвоить имя *Штам* и сохранить. Структура проекта и его элементы отражаются в окне программы **Project Manager** (Менеджер проекта)

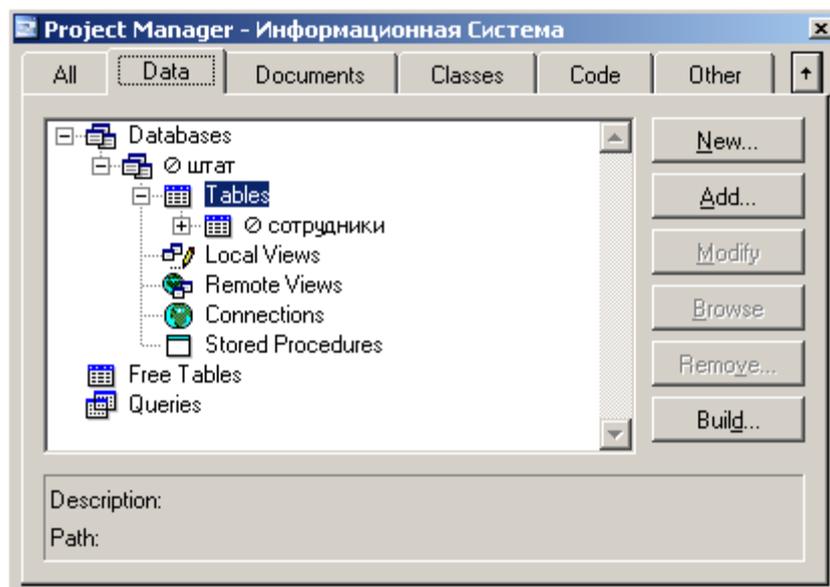


Рис.2. Окно **Project Manager**

6. Добавить БД в проект: в окне Project Manager щёлкнуть на вкладке **Data/Databases/Add**, в открывшемся диалоговом окне выбрать БД и нажать **ОК**
7. Создать таблицу в БД *штат*: в окне **Project Manager** (см. рис.2.) щёлкнуть клавишей мыши на вкладке **Data/Databases/штат/Tables/New/New table**, присвоить имя *Сотрудники* и сохранить
8. В появившемся окне **Table Designer** (Конструктор таблиц) (см. рис.3.) на вкладке **Fields** (Поля) создать структуру таблицы в соответствии с таблицей 3.
- 9.

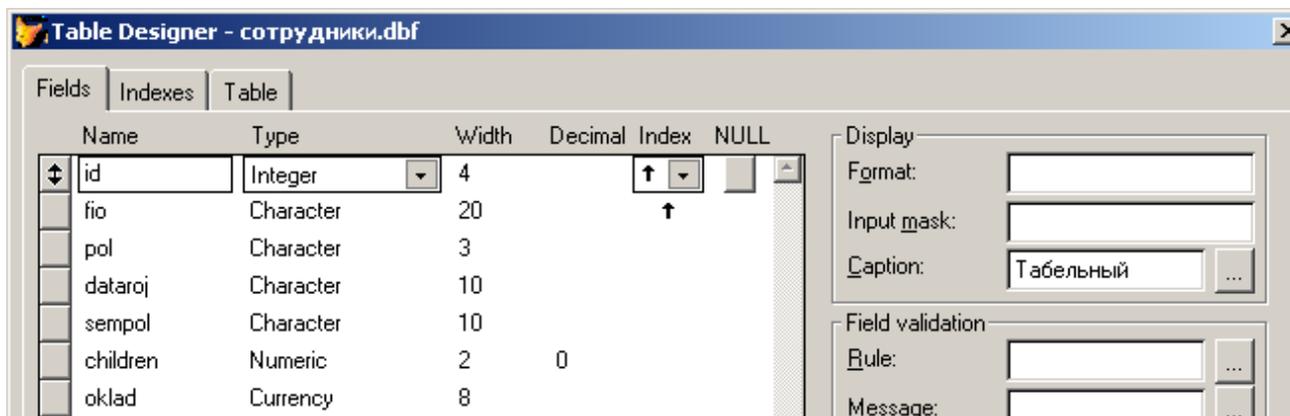


Рис.3. Окно конструктора таблицы **Table Designer** вкладка **Fields**

Таблица 3.Определение полей таблицы *Сотрудники*  
в окне конструктора таблицы **Table Designer**

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
id	Integer	4	↑Ascending	Табельный номер
fio	Character	20		ФИО
pol	Character	3		Пол
dataroj	Character	10		Дата рождения
sempol	Character	10		Семейное положение
children	Numeric	2		Количество детей
oklad	Carrency	8		Оклад
doljnost	Character	15		Должность
adress	Character	30		Место жительства

10.Перейти на вкладку **Indexes** (Индексы) окна конструктора таблицы **Table Designer** и присвоить созданному индексу значения в соответствии с таблицей 4. Это необходимо для создания ключевого поля

Order	Name	Type	Expression	Filter	Collate
↑	id	Primary	id		Machine

11.Для ввода и редактирования данных в таблице *Сотрудники* в командном окне программы введите команду APPEND и нажмите клавишу Enter

Командное окно программы **FoxPro** (см. рис.1.) предназначено для ввода команд с клавиатуры и последующего их выполнения. Любые

действия и операции над элементами приложения в СУБД FoxPro могут осуществляться при помощи команд, вводимых в программное окно.

ФИО	Пол	Дата рождения	Семейное положение	Количество детей	Оклад	Должность	Место жительства
Довгаль П.У.	муж	17.05.1954	женат	3	9000.0000	инспектор	Гагарина 87 кв 132
Прутков И.С.	муж	01.09.1971	женат	1	15000.0000	завотделом	Панина 198 кв 3

Рис.4. Просмотр и редактирование содержимого таблицы в режиме APPEND

12. После ввода данных (см. таблицу 4.) в командном окне программы введите команду BROWSE и нажмите клавишу Enter. Результат должен соответствовать рис.5.

ФИО	Пол	Дата рождения	Семейное положение	Количество детей	Оклад	Должность
Гуськов Г.Г.	муж	22.02.1976	холост	0	7000.0000	экономис
Довгаль П.У.	муж	17.05.1954	женат	3	9000.0000	инспектор
Прутков И.С.	муж	01.09.1971	женат	1	15000.0000	завотдел
Косыгин Н.И.	муж	28.03.1979	холост	0	7000.0000	экономис
Патриков Е.П.	муж	14.08.1980	холост	0	4500.0000	консульт
Белкина Б.Ю.	жен	24.11.1959	замужем	4	12000.0000	бухгалтер
Петренко Е.П.	жен	29.09.1981	незамужем	0	7000.0000	экономис
Петросова Г.Н.	жен	19.06.1976	замужем	2	4500.0000	консульт
Орлова К.Д.	жен	31.12.1977	незамужем	0	7000.0000	экономис
Косыгина А.К.	жен	01.10.1969	замужем	1	6000.0000	менеджер

Рис.5. Просмотр таблицы в режиме BROWSE

Таблица 4. Содержимое таблицы *Сотрудники*

Таб. номер	ФИО	Пол	Дата рождения	Семейное положение	Количество плат	Оклад	Должность	Место жительства
1	Гуськов Г.Г.	муж	22.02.1976	холост	0	7000	экономист	Лаптиева 156 кв 43
2	Довгаль П.У.	муж	17.05.1954	женат	3	9000	инспектор	Гагарина 87 кв 132
3	Прутков И.С.	муж	01.09.1971	женат	1	15000	завотделом	Панина 198 кв 3
4	Косыгин Н.И.	муж	28.03.1979	холост	0	7000	экономист	Николаева 30 кв 77
5	Патриков Е.П.	муж	14.08.1980	холост	0	4500	консультант	Нурадилова 18г кв 84
6	Белкина Б.Ю.	жен	24.11.1959	замужем	4	12000	бухгалтер	Луначевского 64 кв 9
7	Петренко Е.П.	жен	29.09.1981	незамужем	0	7000	экономист	Ленина 101 кв 15
8	Петрова Г.Н.	жен	19.06.1976	замужем	2	4500	консультант	Мира пр. 120 кв 2
9	Орлова К.Д.	жен	31.12.1977	незамужем	0	7000	экономист	Седова 18 кв 112
10	Косыгина А.К.	жен	01.10.1969	замужем	1	6000	менеджер	Панина 198 кв 3

13. В окне BROWSE установите курсор в нижней части таблицы и нажмите комбинацию клавиш **Ctrl + Y** на клавиатуре. В таблицу добавится новая строка

14. Прежде чем удалить строку в **Visual FoxPro** её сначала необходимо пометить на удаление. Для этого щелкнуть клавишей мыши в ячейке слева от удаляемой записи в области узкого, непоименованного

столбца, ячейка станет помеченной чёрным цветом, или выделить необходимую строку и выбрать команду **Toggle Deletion Mark** (Установка метки на удаление) пункта **Table** (Таблица) системного меню **Visual FoxPro**, запись будет помечена на удаление. Удалить запись, выбрав команду **Remove Deleted Records** (Удалить запись) пункта **Table** (Таблица) системного меню **Visual FoxPro**. Запись будет удалена

## Задание

Создать многотабличную БД и присвоить имя *Библиотека*. Создать следующие таблицы:

### 1. Таблица *Книги*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
Kodknigi	Integer	4		Код книги
nazvanie	Character	40		Название
razdel	Character	15		Раздел
izdat	Character	20		Издательство
godizdan	Numeric	4		Год издания
mestohran	Character	5		Место хранения

Определение полей таблицы *Книги* в окне конструктора таблицы **Table Designer**

## Designer

Код книги	Название	Раздел	Издательство	Год издания	Место хранения
1	Практический курс программирования	Информатика	Наука	1983	6-11
2	TURBO PASCAL для школьников	Информатика	Финансы и статистика	1999	6-22
3	Занимательная	Математика	Тригон	1998	3-14

	математика				
4	HTML в действии	Информатика	Питер	1997	5-4
5	Национальное счетоводство	Экономика	Финансы и статистика	1998	4-11
6	Самоучитель Visual FoxPro 8.0	Информатика	БХВ-Петербург	2003	5-34
7	Язык телодвижений	Психология	Наука	1984	2-17
8	Теория машин	Техника	Машиностроение	1957	3-15
9	Теория гипноза	Психология	Наука	1999	2-31
10	Карьера менеджера	Экономика	Парадокс	1998	1-212

Содержимое таблицы *Книги*

## 2. Таблица *Разделы*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
razdel	Character	15		Раздел

Определение полей таблицы *Разделы* в окне конструктора таблицы **Table**

### Designer

Раздел
Экономика
Информатика
Психология
Математика
Техника

Содержимое таблицы *Разделы*

## 3. Таблица *Издательство*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
izdat	Character	20		Издательство
gorod	Character	15		Город

Определение полей таблицы *Издательство* в окне конструктора таблицы

### Table Designer

Издательство	Город
Финансы и статистика	Москва
Тригон	Санкт-Петербург
Питер	Санкт-Петербург
Наука	Москва
Машиностроение	Москва
Парадокс	Минск
БХВ-Петербург	Санкт-Петербург

Содержимое таблицы *Издательство*

## 4. Таблица *Автор\_книги*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
kodavknigi	Integer	4		Код автора книги

kodknigi	Integer	4		Код книги
kodavtora	Integer	4		Код автора

Определение полей таблицы *Автор\_книги* в окне конструктора таблицы

### Table Designer

Код автора книги	Код книги	Код автора
1	1	1
2	1	2
3	2	3
4	3	4
5	4	5
6	5	6
7	6	7
8	7	8
9	8	9
10	9	10
11	10	11

Содержимое таблицы *Автор\_книги*

#### 5. Таблица *Авторы*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
kodavtora	Integer	4		Код автора
familiya	Character	20		Фамилия
imya	Character	15		Имя

Определение полей таблицы *Авторы* в окне конструктора таблицы **Table**

### Designer

Код автора	Фамилия	Имя
1	Фролов	Геннадий
2	Илюхин	Виктор
3	Попов	Владимир
4	Акимова	Светлана
5	Морис	Брюс
6	Кулагина	Галина
7	Омельченко	Людмила
8	Алан	Пиз
9	Мальцев	Анатолий
10	Горин	Дмитрий

11	Якокка	Ли
----	--------	----

Содержимое таблицы *Авторы*

15. Для закрытия всех элементов приложения в командном окне программы введите команду **CLOSE ALL**

16. Для завершения работы с программой **Visual FoxPro** в командном окне программы введите команду **QUIT**.

## Лабораторная работа №2. Создание отношений между таблицами в многотабличной БД Библиотека

### Теоретическая часть

Среди требований, предъявляемых к СУБД, основное место занимает возможность быстрого поиска необходимой информации. Средством, позволяющим решить эту проблему, является применение индексов. В **Visual FoxPro** для создания первичных ключей, определяющих отношения между таблицами и условия целостности данных также предназначены индексы. В этом случае индексы должны быть уникальными, то есть значения индексированного поля должны быть неповторяющимися (уникальными).

Для создания индекса таблицы используется вкладка **Indexes** (Индексы) окна конструктора таблиц **Table Designer**. Все индексы в **Visual FoxPro** имеют имена, задаваемые в поле **Name** (Имя). Для задания типа создаваемого индекса используется список **Type** (Тип).

Таблица 1. Описание типов индекса

Тип индекса	Описание
<b>Regular</b> (Обычный)	Создаётся индекс, в котором для каждой записи таблицы хранится значение индексного выражения. Если несколько записей имеют одинаковое значение индексного выражения, то каждое значение хранится отдельно и содержит ссылку на связанную с ней запись
<b>Unique</b> (Уникальный)	Создаётся индекс, в котором хранятся только неповторяющиеся значения индексного выражения. Если две или более записей содержат одинаковое значение индексного выражения, то будет храниться только одно значение и ссылка на первую из записей с одинаковым

	значением индексного выражения. Таблица может иметь несколько уникальных индексов
<b>Candidate</b> (Кандидат)	Создаётся уникальный индекс, который не содержит полей с пустыми значениями. Этот индекс обладает всеми качествами первичного ключа и не является им только по той причине, что таблица не может содержать более одного первичного ключа
<b>Primary</b> (Первичный)	Создаётся уникальный индекс, который используется для связывания таблиц и определения условий целостности данных. Поля, входящие в первичный ключ, не должны допускать ввода пустых значений. В отличие от уникального индекса, таблица может иметь только один первичный ключ

Обычно, в **Visual FoxPro** при создании форм, отчётов и запросов используются несколько таблиц, между которыми установлены постоянные отношения. Такие таблицы называются связанными. Из двух связанных таблиц одна является главной (родительской), а другая подчинённой (дочерней). При создании индексов для родительской таблицы должен быть определён ключ типа **Primary** (Первичный) или типа **Candidate** (Кандидат), а для дочерней таблицы – индекс для связи с родительской таблицей типа **Regular** (Обычный).

Таблица 2. Типы отношений между таблицами

Типы отношений	Описание
Отношение “один-к-одному”	Каждая запись в одной таблице соответствует только одной записи в другой таблице
Отношение “один-ко-многим”	Наиболее распространённый тип отношений, каждой записи в одной таблице может соответствовать несколько записей в другой таблице
Отношение “много-к-одному”	Отношение “много-к-одному” можно сравнить с отношением “один-ко-многим”, рассматриваемое с другой точки зрения
Отношение “много-ко-многим”	Нескольким записям одной таблицы может соответствовать несколько записей в другой таблице

Одним из самых важных требований, предъявляемых к базам данных, является целостность данных, которую определяют установленные между таблицами отношения. Для определения целостности данных в **Visual FoxPro** используется окно построителя условий целостности данных **Referential Integrity Builder** (Построитель целостности данных), которое содержит перечень всех установленных отношений между таблицами (см. рис.3.).

Таблица 3. Описание действий **Visual FoxPro**, в зависимости от выбранной опции, при изменении значения первичного ключа или ключа типа

**Candidate**

Наименование опции	Описание
<b>Cascade</b> (Каскадное изменение)	При изменении значений полей первичного ключа или ключа-кандидата в родительской таблице автоматически осуществляется каскадное изменение всех соответствующих значений в дочерней таблице
<b>Restrict</b> (Запрет изменения)	Не позволяет изменять значения полей первичного ключа или ключа-кандидата в родительской таблице, если в дочерней таблице имеется хотя бы одна запись, содержащая ссылку на изменяемую запись
<b>Ignore</b> (Игнорировать)	Позволяет изменять значения полей первичного ключа или ключа-кандидата в родительской таблице независимо от существования связанных записей в дочерней таблице. Целостность данных при этом не поддерживается

Таблица 4. Действия **Visual FoxPro**, в зависимости от выбранной опции, при удалении записи из родительской таблицы

Наименование опции	Описание
<b>Cascade</b> (Каскад)	При удалении записи из родительской таблицы автоматически осуществляется каскадное удаление всех записей из дочерней таблицы, связанных с удаляемой записью
<b>Restrict</b> (Запрет)	Не позволяет удалять записи в родительской таблице, если в дочерней таблице имеется хотя бы одна запись,

	содержащая ссылку на удаляемую запись. При попытке удаления записи возникает ошибка, которую вы можете обработать программно
<b>Ignore</b> (Игнорировать)	Позволяет удалять записи в родительской таблице независимо от существования связанных записей в дочерней таблице. Целостность данных при этом не поддерживается

Таблица 5. Действия **Visual FoxPro**, в зависимости от выбранной опции, при добавлении новой записи в родительскую таблицу

Наименование опции	Описание
<b>Restrict</b> (Запрет)	Не позволяет вводить запись, если значение индексного выражения дочерней таблицы не соответствует одной из записей в родительской таблице
<b>Ignore</b> (Игнорировать)	При вводе данных в дочернюю таблицу не анализируется значение индексного выражения. Целостность данных при этом не поддерживается

### Цель работы

1. Создать первичный ключ для таблицы, создать индексы дочерних таблиц для связи с родительской таблицей
2. Создать отношения между таблицами
3. Определить условия целостности данных в базе данных проекта

### Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть созданный проект: **File/Open/Информационная система**
3. Открыть БД *итат*: в окне **Project Manager** (Менеджер проекта) вкладка **Data/Databases/Библиотека/Open**
4. Необходимо модифицировать таблицы БД *Библиотека*, создать индексированные поля, а также первичные ключи для осуществления связей между таблицами. Для модификации (изменения структуры) таблицы в окне проекта **Project Manager** (Менеджер проекта) установить курсор на модифицируемую таблицу *Книги* и нажать

кнопку **Modify** (Модифицировать) или в окне конструктора БД **Database Designer** (Конструктор базы данных) установить курсор в таблицу *Книги*, вызвать контекстное меню и выбрать команду **Modify** (Модифицировать). На экране откроется диалоговое окно **Table Designer** (Конструктор таблицы)

5. В окне **Table Designer** (Конструктор таблицы) перейти на вкладку **Indexes** (Индексы) (см. рис.1.) и ввести значения в соответствии с таблицей 6., нажать кнопку **ОК**, система попросит подтвердить сохранение изменений, нажать **Yes**

Таблица 6.Определение индексов таблицы *Книги* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
↑	kodknigi	Primary	kodknigi		Machine
↑	razdel	Regular	razdel		Machine
↑	izdat	Regular	izdat		Machine

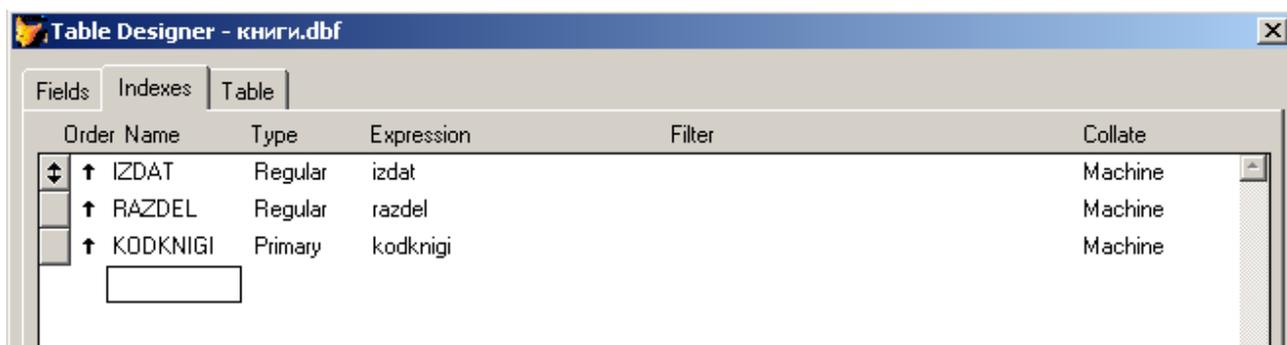


Рис.1. Вкладка **Indexes** (Индексы) окна **Table Designer** таблицы *Книги*

Индекс, с присвоенным типом **Primary** является первичным ключом таблицы

6. Аналогично внести изменения в таблицы БД *Библиотека: Разделы, Издательство, Автор\_книги, Авторы* в соответствии с таблицами 7, 8, 9, 10

Таблица 7.Определение индексов таблицы *Разделы* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
↑	razdel	Primary	razdel		Machine

Таблица 8.Определение индексов таблицы *Издательство* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
↑	izdat	Primary	izdat		Machine

Таблица 9.Определение индексов таблицы *Автор\_книги* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
↑	kodavknigi	Primary	kodavknigi		Machine
↑	kodavtora	Regular	kodavtora		
↑	kodknigi	Regular	kodknigi		

Таблица 10. Определение индексов таблицы *Авторы* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
↑	kodavtora	Primary	kodavtora		Machine

7. В окне **Database Designer** (Конструктор базы данных) выбрать родительскую таблицу *Разделы*. Таблицы в конструкторе БД обозначаются прямоугольниками, в нижней части которых после надписи **Indexes** (Индексы) расположен список индексов, созданных для данной таблицы. Первичный ключ в этом списке выделяется значком ключа, расположенным с левой стороны от наименования индекса (см. рис.2.). Установить курсор мыши на первичный ключ

**razdel** таблицы *Разделы*. Нажать кнопку мыши и, не отпуская ее, переместить курсор мыши на индекс **razdel** дочерней таблицы *Книги*. Отпустить кнопку мыши. Созданные отношения между таблицами отображаются в виде линий (см. рис.2.). Аналогичным образом, создать связь между таблицами *Издательство* и *Книги* (Таблица *Издательство* – родительская, таблица *Книги* – дочерняя, первичный ключ – **izdat**)

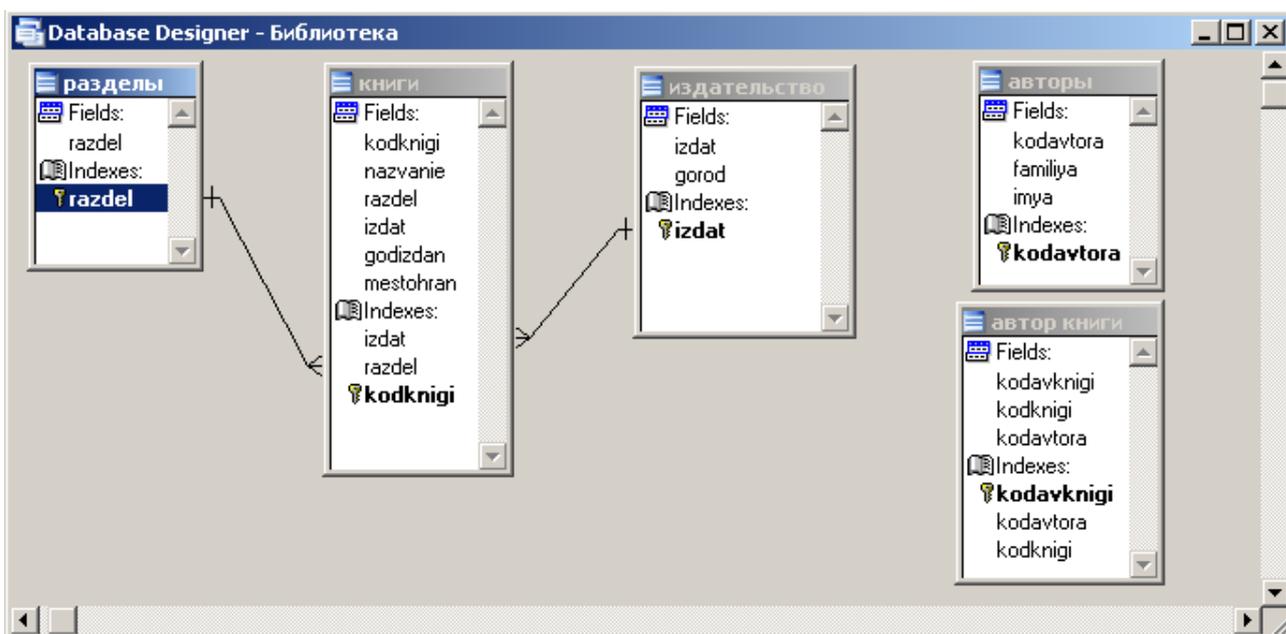


Рис.2. Отображение связей между таблицами в окне **Database Designer** (Конструктор базы данных)

8. Для редактирования отношений между связанными таблицами дважды щелкнуть левой кнопкой мыши на линии, появится диалоговое окно **Edit Relationship** (Редактирование отношения), в котором слева приведено наименование родительской таблицы и расположен раскрывающийся список индексов таблицы, справа размещена аналогичная информация о дочерней таблице. В этом диалоговом окне указан также тип установленного отношения между таблицами.
9. В окне **Edit Relationship** (Редактирование отношения) нажать кнопку **Referential Integrity** (Целостность данных), в появившемся диалоговом окне **Referential Integrity Builder** (Построитель целостности данных)

выбрать отношение **издательство-книги**. В полях **Update** (Изменить), **Delete** (Удалить), **Insert** (Заменить) установить тип действий **Restrict** (Запрет изменения) (см. таблица 3, 4, 5). Провести аналогичные действия для отношения **разделы-книги**. Результат описанных действий, которые необходимы для обеспечения целостности данных, представлен на рис.3.

10. Для сохранения выполненных действий нажать кнопку **ОК**, система потребует подтверждение сохранения, нажать кнопку **Да**

11. Нажать кнопку **ОК** для закрытия диалогового окна **Edit Relationship**

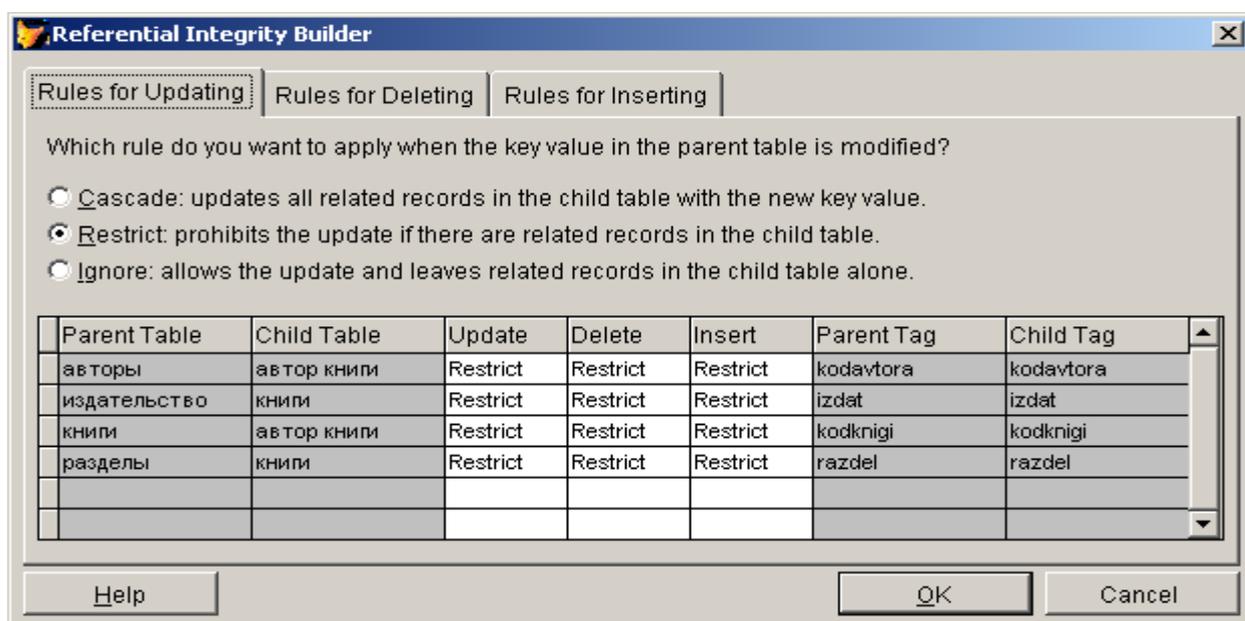


Рис.3. Диалоговое окно **Referential Integrity Builder** (Построитель целостности данных)

### Задание

1. Создать связь между таблицами *Книги* и *Автор\_книги* (kodknigi)
2. Создать связь между таблицами *Авторы* и *Автор\_книги* (kodavtora)
3. Установить целостность данных в созданных отношениях: в полях **Update** (Изменить), **Delete** (Удалить), **Insert** (Заменить) установить тип действий **Restrict** (Запрет изменения)

### Лабораторная работа №3. Создание простейшей однотабличной формы в среде Visual FoxPro.

#### Теоретическая часть

В **Visual FoxPro** для просмотра, ввода и редактирования данных, хранящихся в таблицах, используются формы, являющиеся более наглядным средством представления информации. Важным преимуществом форм является, то, что они позволяют работать не с одной, а с несколькими связанными таблицами, что, в свою очередь, также увеличивает наглядность.

При создании форм в **Visual FoxPro** разработчик может использовать следующие средства:

- **Form Wizard** (мастер форм)
- **Form Builder** (построитель формы)
- **Builder** (построитель объектов формы)
- **Auto Format Builder** (построитель автоформата)
- **Form Designer** (конструктор форм).

Чтобы создать форму для одной или связанных таблиц с возможностью задания отображаемых в форме полей, стиля их отображения и указания типа кнопок управления, можно использовать **Form Wizard** (мастер создания форм).

Для самостоятельной разработки формы с заданными свойствами или изменения формы, созданной с помощью мастера, вам необходимо использовать **Form Designer** (конструктор форм).

Для облегчения размещения в конструкторе форм полей и надписей, оформленных в соответствии с выбранным стилем, можно использовать **Form Builder** (Построитель формы).

### **Цель работы**

1. Создать форму *Сотрудники* с помощью **Form Wizard** (мастер форм)
2. Запустить форму *Сотрудники*;
3. Редактировать данные в таблице *Сотрудники* с помощью созданной формы *Сотрудники*.

## Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть созданный проект: **File/Open/Информационная система**
3. Открыть БД *umat*: в окне **Project Manager** (Менеджер проекта) вкладка **Data/Databases/umat/Open**
4. Создать форму *Сотрудники*, щелкнув клавишей мыши на вкладке **Documents/Forms/New/Form Wizard/Form Wizard**, нажать клавишу **Ok**



Рис.1. Диалоговое окно **Wizard Selection**

5. В появившемся окне **Form Wizard** в области **Databases and tables** (Базы данных и таблицы) выбрать необходимую БД *umat* и указать таблицу *Сотрудники*, для которой создается форма. Из области **Available fields** (Имеющиеся поля) выбрать поля, которые будут размещены в форме, в соответствии с рис.2, используя кнопки расположенные между списками для перехода к следующему шагу нажать кнопку **Next** (Далее)

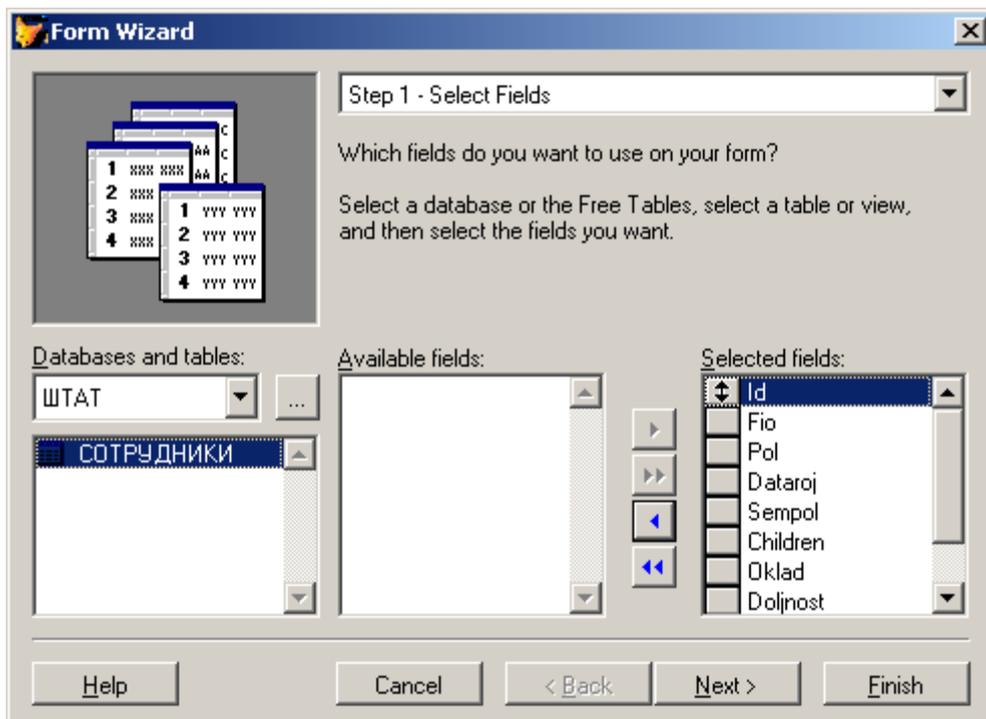


Рис.2. Диалоговое окно выбора полей для отображения.

6. В появившемся диалоговом окне в области **Style** установить стиль отображения объектов и в области **Button type** типы кнопок управления в соответствии с рис.3, нажать кнопку **Next**

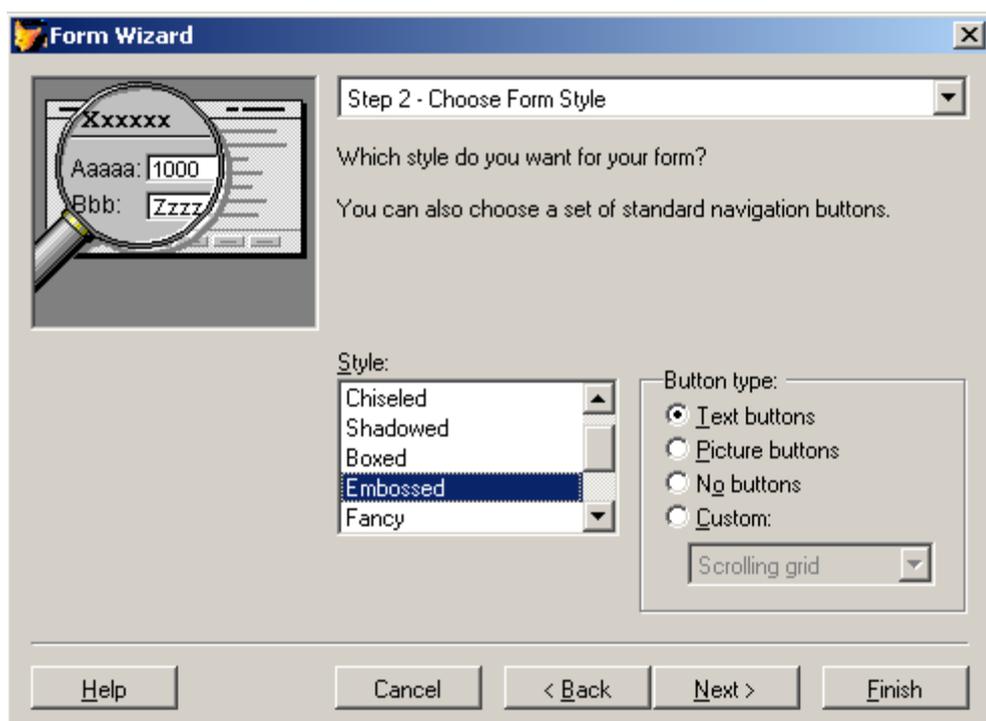


Рис.3. Окно выбора стиля отображения полей и управляющих кнопок.

7. В появившемся диалоговом окне задать критерий сортировки данных, указав поля по которым будет осуществляться упорядочивание в соответствии рис.4. нажать кнопку **Next**

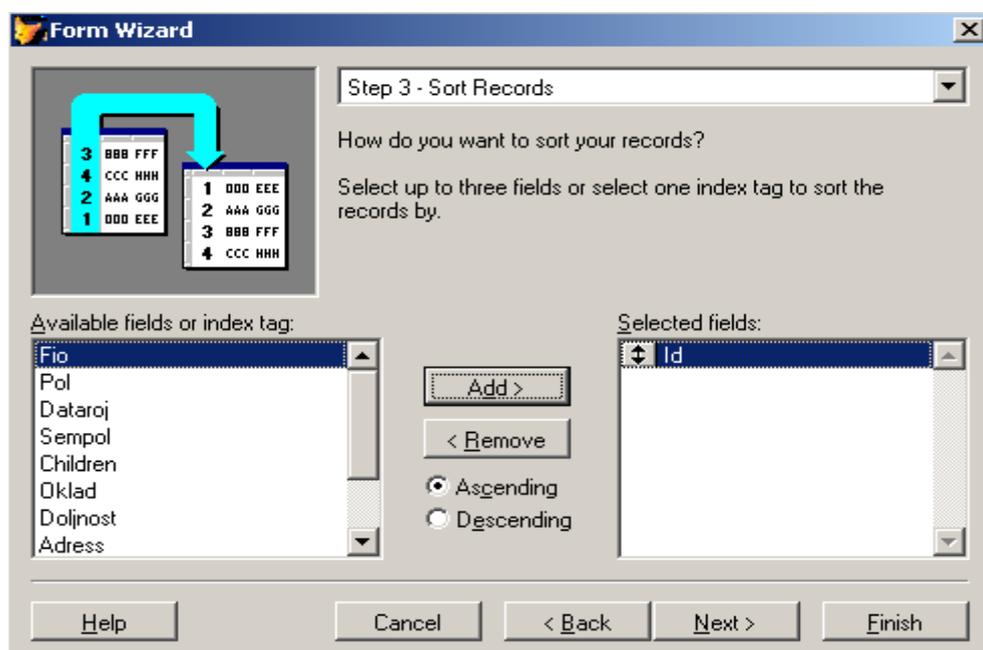


Рис.4. Установка критерия упорядочения данных

8. На заключительном шаге создания форм в области **Type a title for your form** (Тип заголовка формы) указать имя формы *Сотрудники*. Воспользовавшись кнопкой **Preview** (Просмотр) просмотрите, как будет выглядеть создаваемая форма, после просмотра нажмите кнопку **Return to Wizard**.

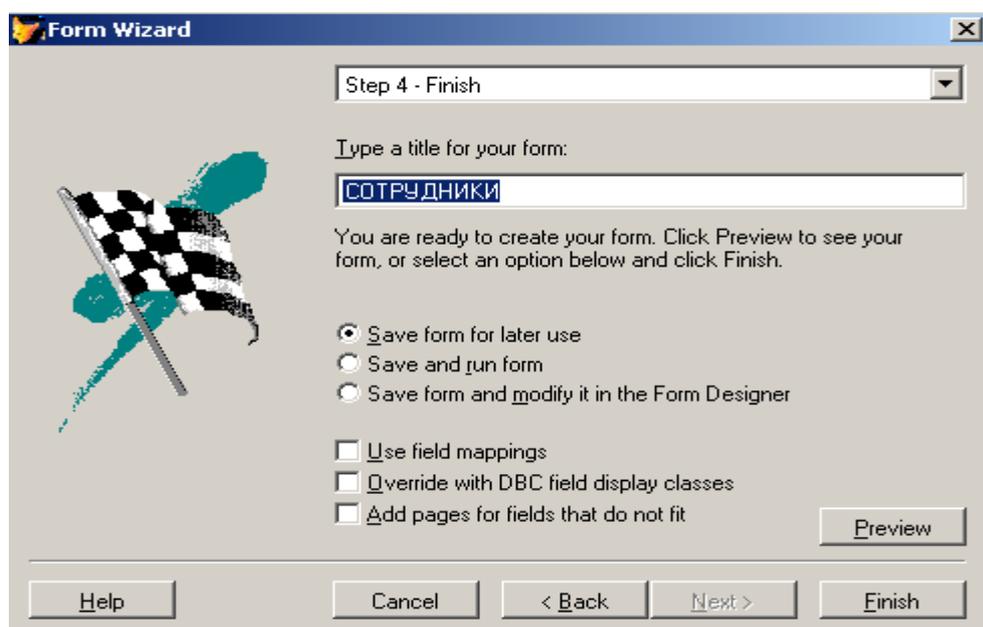


Рис.5.Окно задания заголовка формы.

Если что-то не так, вернитесь к предыдущим шагам, воспользовавшись кнопкой **Back**. Нажать кнопку **Finish** (Готово) и в появившемся диалоговом окне **Save As** (Сохранить как) указать папку, в которой будет размещена форма *Сотрудники*

9. Запустить в окне **Project Manager** форму *Сотрудники*:  
**Documents/Forms/Сотрудники/Run**

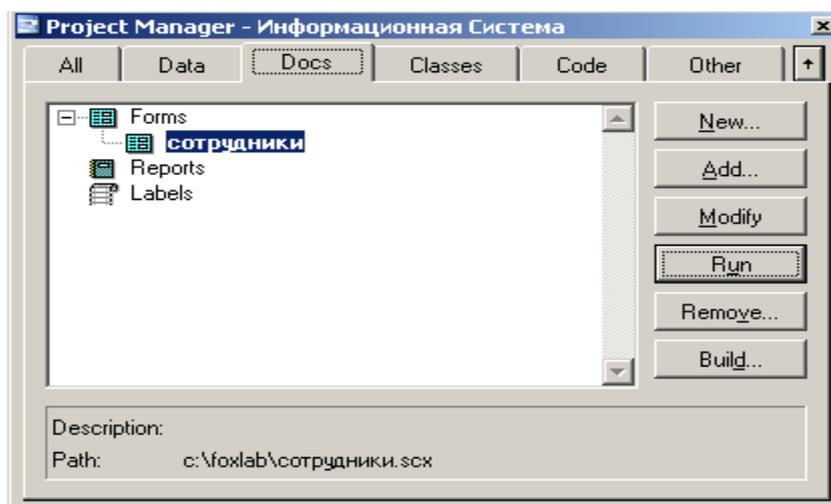


Рис.6. Окно **Project Manager**

Или с помощью меню программы **Visual FoxPro: Program/Do** задать имя формы *Сотрудники* и тип файла **Form**, нажать кнопку **Do**. С помощью кнопок (см. таблицу 1.), находящихся в нижней части формы можно вводить или редактировать записи в таблице сотрудники.

Рис.7.Форма *Сотрудники*

10. Ввести в таблицу *Сотрудники* дополнительно шесть записей, одну удалить, две отредактировать

Таблица 1.Кнопки управления и редактирования данных

Top	Prev	Next	Buttom	Find	Save
Первая запись	Предыдущая запись	Следующая запись	Последняя запись	Найти	Сохранить
Print	Add	Edit	Delete	Exit	Revent
Вывести на печать	Добавить запись	Редактировать запись	Удалить запись	Выход	Отменить

### Задание

Создать форму для таблицы *Книги* БД *Библиотека*

1. В диалоговом окне выбора полей для отображения (см. рис.2.) выбрать все поля таблицы *Книги*
2. В окне выбора стиля отображения полей и управляющих кнопок (см. рис.3.) в области **Style** (стиль) выбрать стиль **Embossed**, а в области **Button type** (Типы кнопок управления) выбрать пункт **No Buttons**
3. Установить критерий упорядочения данных по полю **kodknigi** (см. рис.4.)
4. В окне задания заголовка формы в области **Type a title for your form** (Тип заголовка формы) указать имя формы *Книги*. Просмотреть вид создаваемой формы нажатием кнопки **Preview** (Просмотр).
5. После сохранения запустить форму *Книги*, она должна иметь вид как на рис.8.

КНИГИ

Код книги: 1

Название: Практический курс программирования

Раздел: Информатика

Издательство: Наука

Год\_издания: 1983

Место\_хранения: 6-11

Рис.8. Форма *Книги*

## Лабораторная работа №4. Создание формы с помощью конструктора форм

### Теоретическая часть

Любая форма в **Visual FoxPro** состоит из объектов, каждый из которых имеет характерные свойства. Для любого объекта вы можете указать действия, выполняемые написанной разработчиком программой при наступлении определённых событий. Процесс создания формы в конструкторе форм состоит в размещении в форме объектов и определении свойств, а также связанных с ними событий и выполняемых действий.

Процесс создания формы включает следующие действия:

- настройка параметров формы
- определение среды окружения, то есть выбор используемых в форме таблиц и установка связей между ними
- размещение в форме объектов: текста, полей различных типов, линий, рисунков, кнопок управления.

### Цель работы

1. Настройка параметров формы
2. Ознакомление с объектами формы

3. Размещение объектов в форме с помощью **Form Builder** (Построитель форм)

### **Практическая часть**

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть проект *Информационная система*
3. Открыть БД *штат*
4. Выбрать в окне **Project Manager** вкладку **Documents/Forms/New/New Form**. Откроется окно **Form Designer** (Конструктор форм) для создания новой формы
5. Открыть окно окружения формы **Data Environment** (Среда окружения): пункт меню **View/ Data Environment**. Для размещения таблицы в среде окружения выбрать команду **Add** (Добавить) из меню **Data Environment** (Среда окружения) или вызвать контекстное меню на окне окружения формы **Data Environment** и выбрать пункт **Add**. В появившемся диалоговом окне **Add Table or View** (Добавить таблицу или представление данных), выбрать из списка таблиц открытой базы *штат* таблицу *Сотрудники*, для которой создаётся форма, и нажать кнопку **ОК**
6. Открыть окно свойств таблицы, размещённой в окне окружения **Data Environment**. Для этого установить на таблицу *Сотрудники* курсор, вызвать контекстное меню правой клавишей мыши и выбрать команду **Properties** (Свойства)
7. В окне **Properties** (свойства) выделить свойство **Order** (Упорядочение). Для упорядочения данных в форме в поле коррекции свойства нажмите кнопку раскрытия списка и из списка индексов таблицы выберите индекс (**id**), по которому хотите упорядочить данные.
8. Закройте окно определения среды окружения **Data Environment**.
9. Для задания свойств формы установить курсор в форму, вызвать контекстное меню, выбрать команду **Properties** (Свойства). Откроется

окно **Properties** (Свойства). В его верхнем списке, указывающем название объекта, для которого осуществляется настройка свойств содержится текст Form1 (Форма1)

10. В окне **Properties** (Свойства) скорректировать свойство Caption (Надпись), введя в текстовое поле заголовок формы *Сотрудники тест*
11. Свойство формы AutoCenter (Автоцентр) должно иметь значение True (Истина), чтобы форма располагалась в центре экрана
12. Изменить свойства FontName (Наименование шрифта), указав шрифт **Times New Roman** и FontSize (Размер шрифта) указав размер шрифта **12**

После того как вы определили параметры формы, разместили в окружении используемые таблицы, можно приступать к размещению объектов в форме. Осуществим размещение полей таблицы *Сотрудники* и надписей к ним, используя **Form Builder** (построитель формы)

13. Для запуска построителя форм установить курсор в форму, вызвать контекстное меню, выбрать команду **Builder** (Построитель), откроется диалоговое окно **Form Builder** (построитель формы), содержащее две вкладки: **Field Selection** (Выбор поля) и **Styles** (Стиль). В вкладке **Field Selection** (Выбор поля) из области **Databases and tables** (Базы данных и таблицы) выбрать БД *штат* и таблицу *Сотрудники*, затем из списка **Available fields** (Имеющиеся поля) перенести в **Selected fields** (Выбранные поля) все необходимые поля используя кнопки расположенные между списками. Перейти на вкладку **Styles** (Стиль), выбрать из списка стилей стиль Embossed и нажать кнопку **ОК**. На форме будут размещены объекты формы (см. рис. 1.).

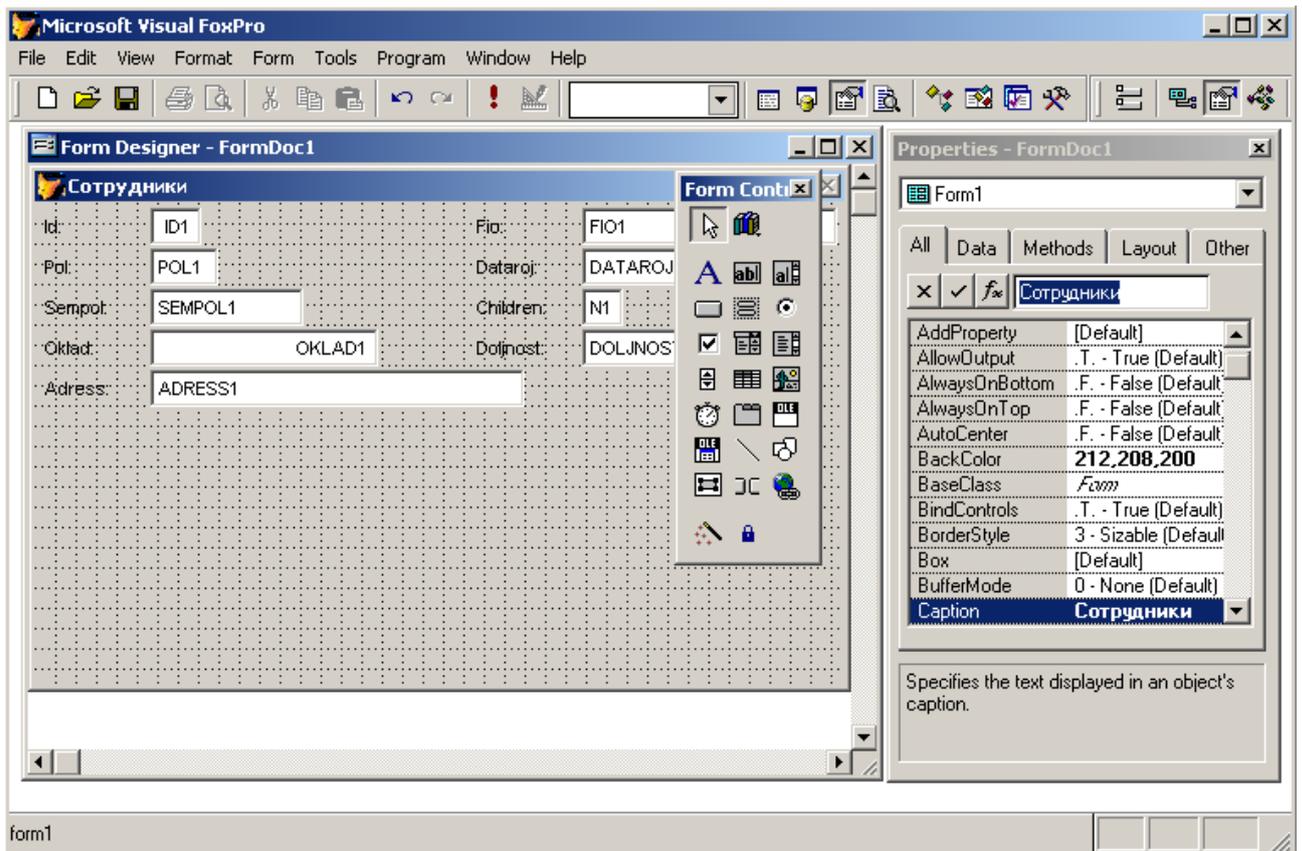


Рис.1. Окно **Form Designer** (Конструктор форм)



Рис.2. Объект - блок кнопок управления

14. Для размещения на создаваемой форме кнопок управления воспользуемся ранее созданной формой *Сотрудники*. Открыть форму *Сотрудники* в режиме редактирования: окно программы **Project Manager/Documents/Forms/сотрудники/Modify**. В появившемся окне **Form Designer** формы *Сотрудники* нажатием левой клавиши мыши выделить объект **BUTTONSET1** (блок кнопок управления, см. рис.2.) и нажать комбинацию клавиш **Ctrl+C** на клавиатуре или установить курсор в форму, вызвать контекстное меню, выбрать команду **Copy**. Закрыть окно **Form Designer** формы *Сотрудники*.
15. В окне **Form Designer** формы *Сотрудники тест* установить курсор в форму, вызвать контекстное меню, выбрать команду **Paste** или нажать

комбинацию клавиш **Ctrl+V**. Блок кнопок управления записями таблицы (см. рис.2.) в создаваемой форме будет скопирован.

16. Задать цвет фона формы. Выделить в окне **Properties** (Свойства) свойство формы **BackColor** (цвет фона), нажать расположенную с правой стороны поля редактирования свойства кнопку и в открывшемся диалоговом окне **Цвет** выберите цвет, который вы хотите использовать для фона

17. Расположить объекты формы в соответствии с рис.3.. Для перемещения объектов формы можно использовать метод перетаскивания мышью или кнопками управления курсором на клавиатуре, предварительно выделив необходимый объект, а также комбинацией клавиш **Ctrl+[кнопки управления курсором]**, использовать все выше перечисленные способы, выяснить их различия. Для изменения размеров формы или объектов формы необходимо выделить объект (форму), подвести курсор к углу объекта (формы), когда курсор примет вид разнонаправленной стрелки нажать левую клавишу мыши и удерживая её, изменить размеры объекта (формы)

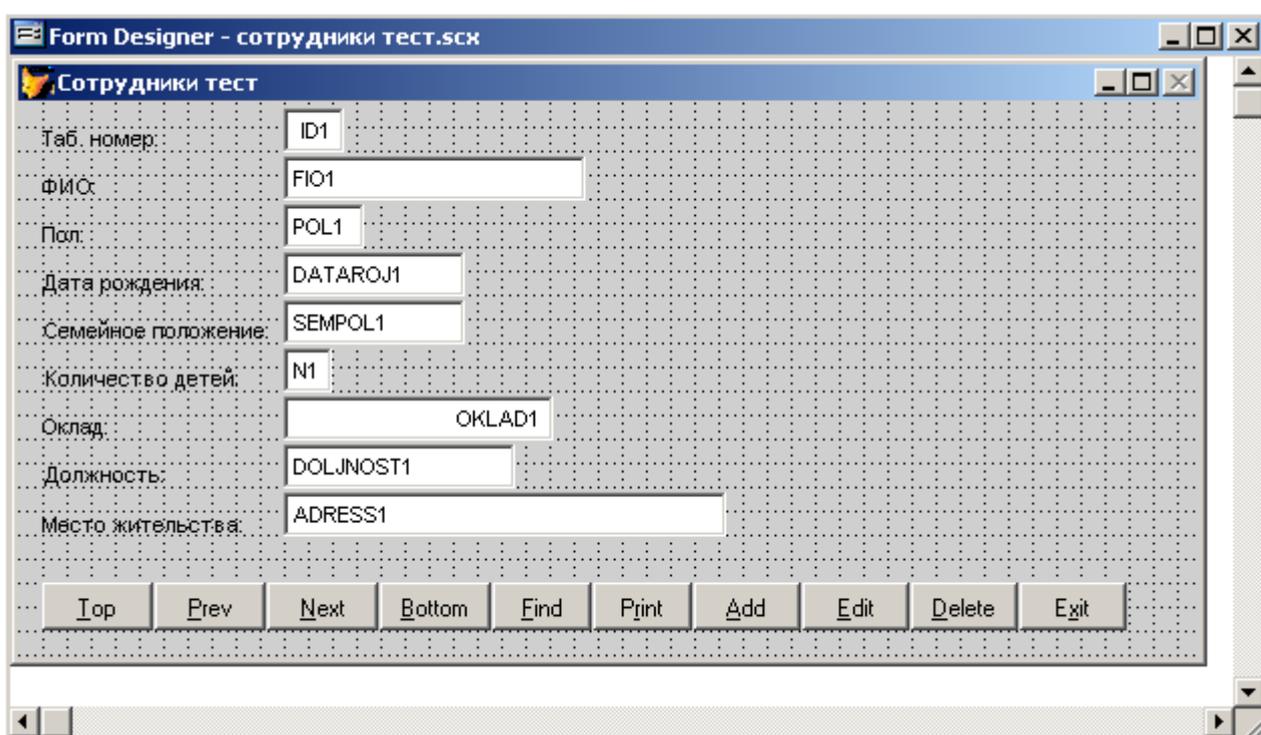


Рис.3. Вид отредактированной формы

18. Запустить отредактированную форму: установить курсор в форму, вызвать контекстное меню, выбрать команду **Run Form** или нажать кнопку [!] на панели инструментов. Система попросит сохранить созданную форму, нажать кнопку **Yes**, в появившемся окне указать папку и имя файла (*Сотрудники тест*), сохранить.

### Задание

Создать кнопки навигации в форме *Книги*

1. Открыть форму *Книги* в окне **Form Designer** (Конструктор форм)
2. Нажать кнопку **Command Group** (Группа кнопок)  на панели инструментов **Form Controls** (элементы управления формы) и щёлкнуть в месте их предполагаемого размещения в форме
3. В Окне **Properties** (свойства) размещённого объекта выделить свойство **ButtonCount** (Количество кнопок) и указать необходимое количество размещаемых в форме кнопок, указав число **5**
4. Увеличить с помощью мыши размеры рамки, окружающей данный объект. Перейти в режим редактирования: установить на объект курсор, вызвать контекстное меню и выбрать команду **Edit** (Редактировать). Выделяя поочерёдно кнопки, переместить их, расположив горизонтально, в одну линию
5. Выйти из режима редактирования, щёлкнув вне области объекта **Command Group** (Группа кнопок)
6. Скорректировать размер рамки, окружающей объект: в свойстве **AutoSize** (Авторамер) установить значение **True** (Истина)
7. Открыть окно свойств объекта **Command Group** (Группа кнопок), нажать кнопку раскрытия списка в верхней части данного окна, поочерёдно выбрать из списка элементы **Command1**, **Command2**,

Command3, Command4, Command5 и, используя свойство Caption (Надпись) задать названия кнопок соответственно: **Первая**, **Следующая**, **Предыдущая**, **Последняя**, **Выход** (см. рис.4.)

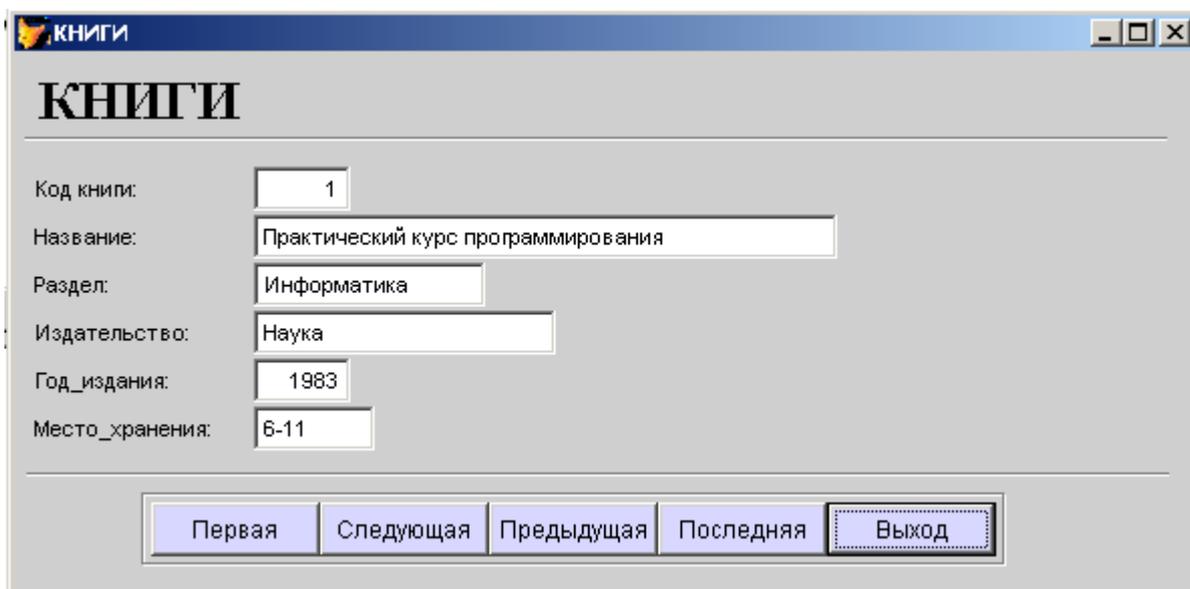


Рис.4. Форма *Книги*

8. Определить команды, которые будут выполняться при нажатии на созданные кнопки. Установить на объект Command Group (Группа кнопок) курсор, вызвать контекстное меню и выбрать команду **Edit** (Редактировать), выделить кнопку **Первая**, вызвать контекстное меню и выбрать команду **Code** (Код программы), в появившемся окне ввести следующие команды:

Для кнопки **Первая**:

\* Переходим на первую запись и обновляем информацию в форме

```
IF !BOF()
```

```
GO TOP
```

```
ENDIF
```

```
_screen.ActiveForm.Refresh()
```

9. После ввода кода программы закрыть окно.

10. Ввести аналогично команды для остальных кнопок:

Для кнопки **Следующая**:

\* Переходим на следующую запись и обновляем информацию в форме

IF !EOF()

SKIP

ENDIF

\_screen.ActiveForm.Refresh()

Для кнопки **Предыдущая**:

\* Переходим на предыдущую запись и обновляем информацию в форме

IF !BOF()

SKIP - 1

ENDIF

\_screen.ActiveForm.Refresh()

Для кнопки **Последняя**:

\* Переходим на последнюю запись и обновляем информацию в форме

IF !BOF()

GO BOTTOM

ENDIF

\_screen.ActiveForm.Refresh()

Для кнопки **Выход**:

\* Запрашиваем и выходим, если Да

IF MESSAGEBOX("Выход из формы?", 4+32+256, "Выход")=6

\_screen.ActiveForm.Release()

```
ELSE  
_screen.ActiveForm.Refresh()  
ENDIF
```

11. После ввода команд закрыть окно процедур
12. Запустить форму на выполнение. Проверить результат проделанной работы

### **Лабораторная работа №5. Создание и редактирование табличного отчёта в среде Visual FoxPro.**

#### **Теоретическая часть**

Отчет – форматированное представление данных, выводимое на экран, принтер или в файл. Отчет, создаваемый в **Visual FoxPro**, может быть представлен в табличном виде или в свободной форме. Табличные отчеты используются для печати данных, представленных в виде списка. При создании отчета в **Visual FoxPro** разработчик может использовать следующие средства:

- **Report Wizard** – Мастер отчета. Позволяет быстро создать отчёт, применяя сортировку, группировку данных и заданный разработчиком стиль оформления;
- **Report Designer** – Конструктор отчета. Позволяет разрабатывать собственные отчёты или модифицировать отчёты, созданные с помощью мастера;
- **Quick Report** – Быстрый отчет. Данное средство предназначено для размещения в конструкторе отчёта полей и задания среды окружения.

#### **Цель работы**

1. Создать однотобличный отчёт *Сотрудники* с помощью **Report Wizard** (Мастер отчёта)

2. Отредактировать отчёт в окне конструктора отчётов **Report Designer** (конструктор отчёта)

### **Практическая часть**

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть проект *Информационная система*
3. Открыть БД проекта. Для этого необходимо на вкладке **Data** (Данные) установить курсор на её название и нажать кнопку **Open** (Открыть) окна проекта. При этом на стандартной панели инструментов в списке **Databases** (Базы данных) появится название открытой БД
4. Перейти на вкладку **Documents/Reports/New/New Report /Report Wizard**
5. В появившемся диалоговом окне **Wizard Selection** указать тип создаваемого отчета **Report Wizard** (Мастер отчетов) и нажать кнопку **ОК**
6. Появляется первое диалоговое окно мастера отчёта (см. рис.1.). Выбрать из верхнего списка области **Databases and tables** (Базы данных и таблицы) базу данных *штат*, а из нижнего – таблицу *сотрудники*



Рис.1. Диалоговое окно создания однотабличного отчета с помощью мастера.

7. Перенести из списка **Available fields** (Имеющиеся поля) в список **Selected fields** (Выбранные поля) поля, которые вы хотите разместить в создаваемом отчете (в соответствии с рис.1.). После формирования списка отображаемых в отчете полей нажать кнопку **Next** (Далее) для перехода к следующему шагу в создании отчета
8. В следующем окне мастера создания отчета указать поля, по которым будут осуществляться группировка данных в отчете (рис.2.)



Рис.2. Диалоговое окно определения полей для группировки данных в отчете.

В центре диалогового окна расположены три раскрывающихся списка, позволяющих задать до трех группировок данных в отчете. Эти списки содержат все поля таблицы. Для осуществления группировки данных в отчете выбрать нужное поле из раскрывающегося списка 1. При создании второй и третьей группировки используются, соответственно, списки 2 и 3. В нашем примере группировка данных по полям не осуществляется, это диалоговое окно мастера отчётов необходимо пропустить и нажать кнопку **Next**

9. В следующем диалоговом окне мастера задается стиль отображения объектов в отчете (рис.3.). Список **Style** (Стиль) содержит несколько вариантов отображения объектов (полей, линий, заголовков и т.д.) в отчете. С помощью области просмотра в верхнем левом углу диалогового окна можно просмотреть тот стиль, который мы выбрали. Выбрав стиль, нажать кнопку

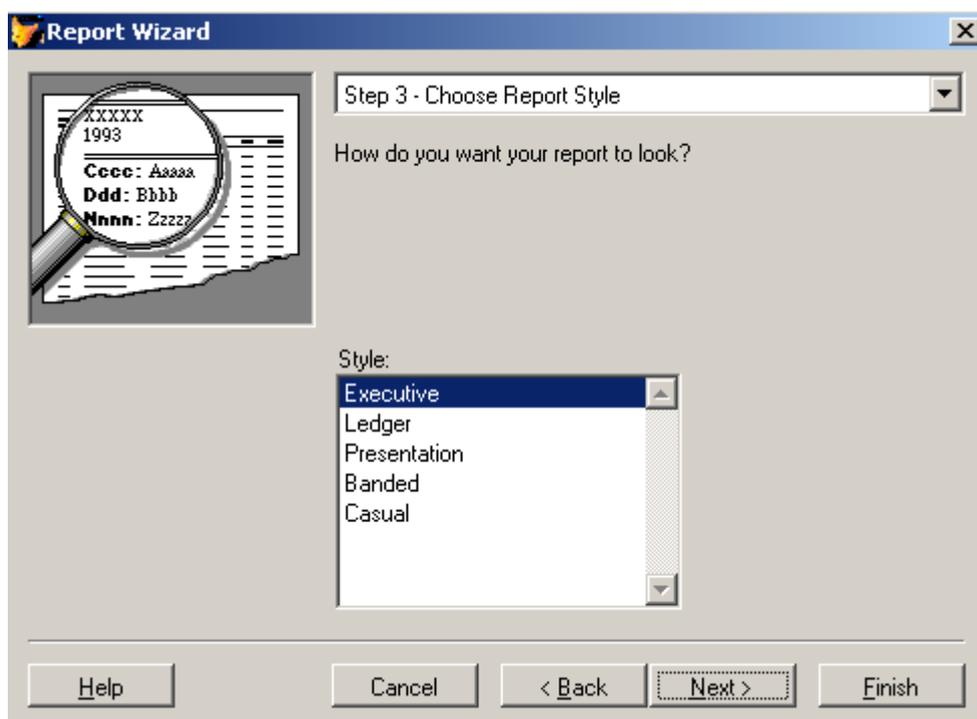


Рис.3. Диалоговое окно выбора стиля отображения объектов.

10. Указать порядок размещения объектов в отчете (рис.4.) и ориентацию страницы отчета. В области **Number of Columns** (Количество колонок) указать **1**, в области **Orientation** (Ориентация) Указать **Portrait** (Книжная), в области **Field Layout** (Расположение полей) указать Rows (Строка). После установки требуемых опций, нажать кнопку **Next**

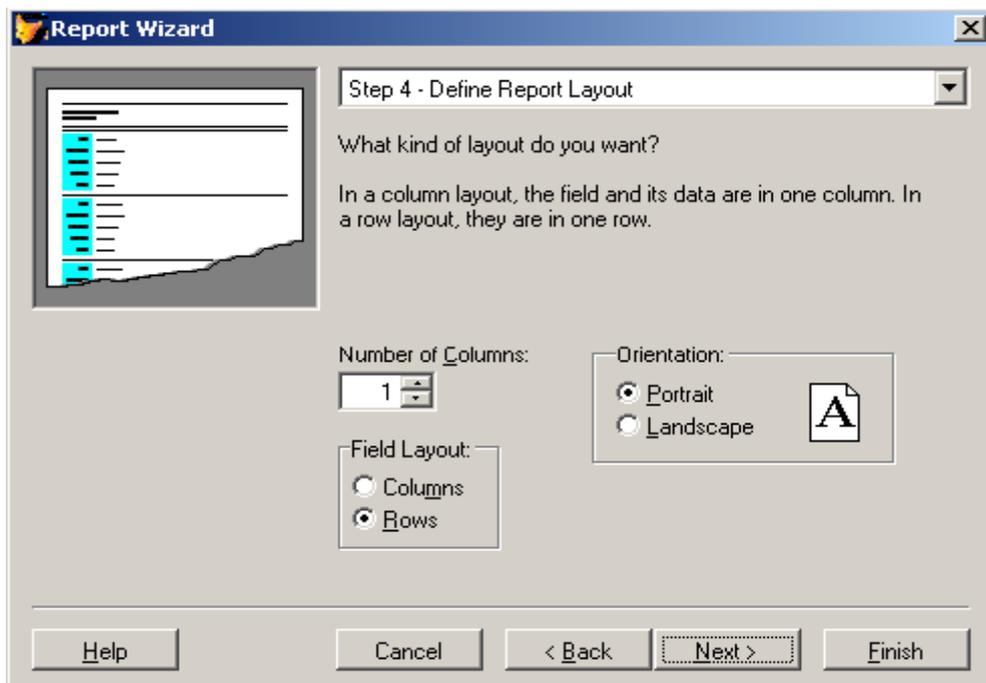


Рис.4. Диалоговое окно установки порядка размещения объектов.

11. Задать поле, по которому требуется упорядочение данных в отчете в соответствии с рис.5. Из списка **Available fields or index tag** (Выбранные поля и индексы) перенести в список **Selected fields** (Выбранные поля) поле **Fio** (упорядочивание данных по фамилии). Для переноса полей используйте кнопку **Add** (Добавить). Сформировав список полей, нажать кнопку **Next** (Далее);

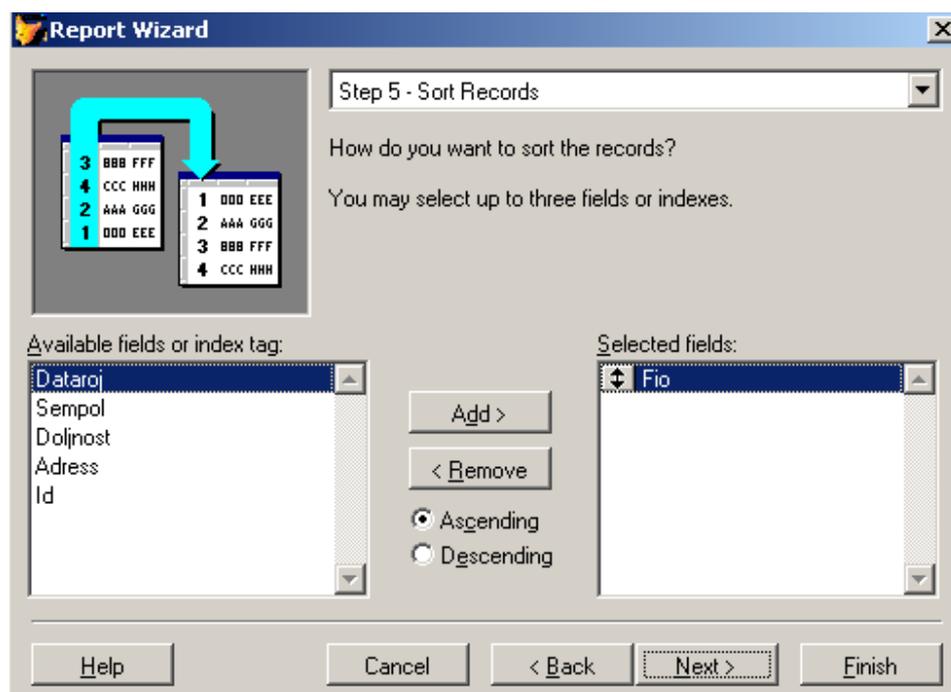


Рис.5. Диалоговое окно установки критерия упорядочения данных.

12. На заключительном шаге создания отчета в области **Type a title for your report** (Введите имя заголовка отчета) указать имя отчёта *Сотрудники*. Воспользовавшись кнопкой **Preview** (Просмотр) посмотреть, как будет выглядеть создаваемый отчет, после просмотра нажать кнопку **Close Preview** на панели **Print Preview** (см. рис.6.), если что-то не так, вернуться к предыдущим шагам, воспользовавшись кнопкой **Back**. Нажать кнопку **Finish** (Готово) и в появившемся диалоговом окне **Save As** (Сохранить как) указать папку, в которой будет размещен отчет *Сотрудники*.



Рис.6. Панель просмотра отчёта **Print Preview**

13. Просмотреть созданный отчёт: в окне **Project Manager** выбрать вкладку **Documents/Reports/Сотрудники/Preview** или команду **Preview** (Просмотр) из меню программы **View** (Вид)
14. Открыть отчёт *Сотрудники* в окне **Report Designer** (конструктор отчёта): в окне **Project Manager** выбрать вкладку **Documents/Reports/Сотрудники/Modify**

Таблица 1. Типы полос отчёта

<b>Title</b> (Титул)	В этой полосе размещается информация, появляющаяся перед основным отчётом (имя отчёта, сопроводительное письмо и т. д.) и называется титульной
<b>Page Header</b> (Верхний колонтитул)	Данные, помещённые в эту полосу, печатаются в начале каждой страницы (название отчёта, текущая дата и т. д.)

<b>Detail</b> (Детали)	Эта полоса содержит данные полей из таблицы или результат вычислений над ними
<b>Page Footer</b> (Нижний колонтитул)	В нижнем колонтитуле печатается название отчёта, дата, номер страницы, итоговые значения по данным текущей страницы

15. Выделить все объекты в полосе **Title** (Титул) и перенести их в полосу **Page Header** (Верхний колонтитул) используя метод перетаскивания мышью или кнопками управления курсором на клавиатуре, предварительно выделив необходимый объект, или комбинацией клавиш **Ctrl+[кнопки управления курсором]**

16. В полосе **Title** (Титул) напечатать следующий текст: *“Отчёт - форматированное представление данных, выводимое на экран, принтер или в файл. Табличный отчёт используется для печати данных, представленных в виде списка”*

17. Поместить набранный текст в прямоугольник со скругленными краями: нажать на панели инструментов **Report Controls** кнопку  и поместить данный объект в отчёт. Отредактировать его расположение в соответствии с рис.7.

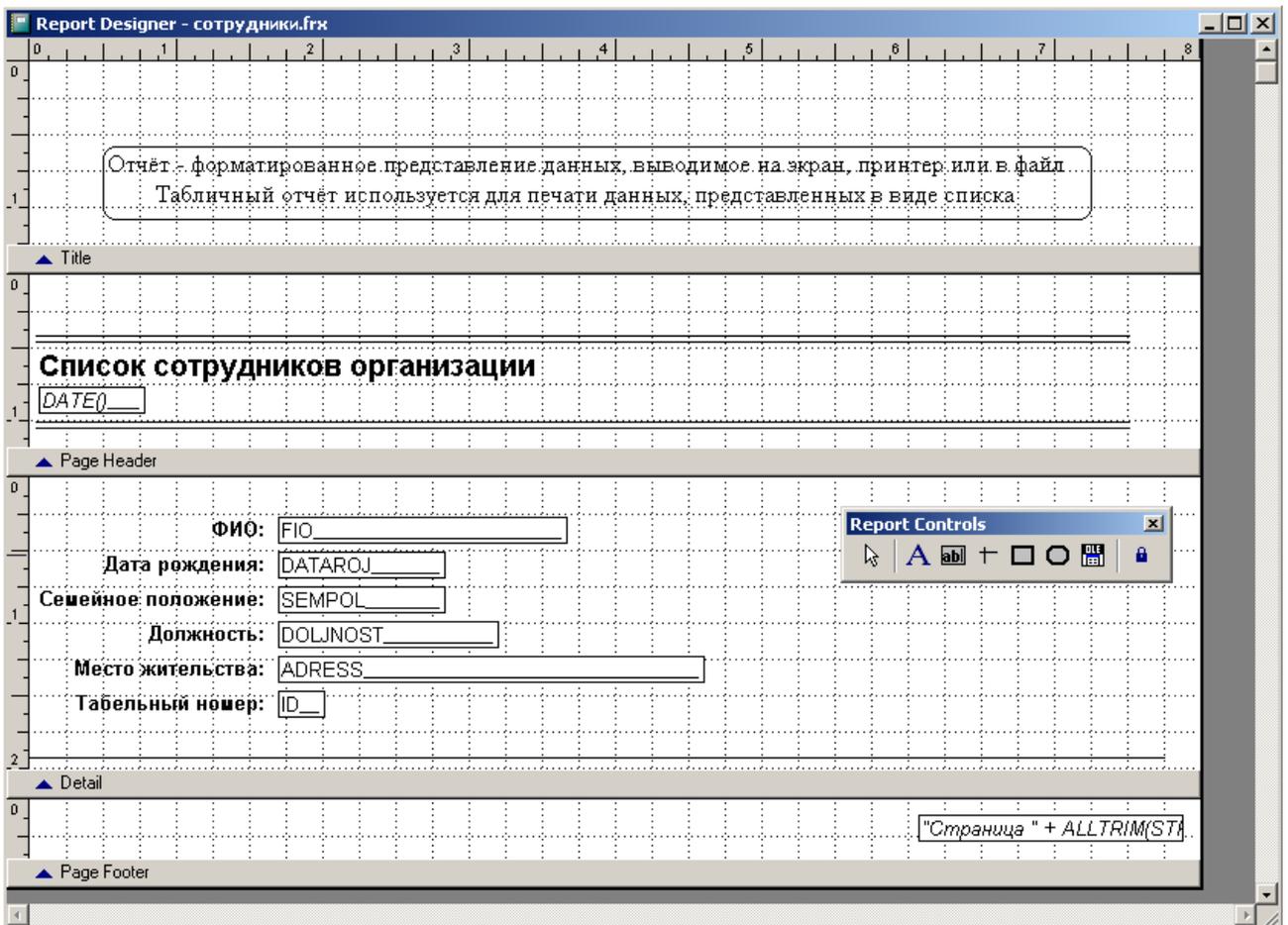


Рис.7. Окно **Report Designer** (Конструктор отчётов)

18. В процессе работы просматривайте результаты редактирования, используя пункт меню **View/Preview**
19. В полосе **Detail** (Детали), ниже поля Табельный номер провести линию, используя кнопку  панели инструментов **Report Controls** и отредактировать её расположение в отчёте в соответствии с рис.7.
20. В полосе **Page Footer** (Нижний колонтитул) переместить элемент `"Page " + ALLTRIM(STR( PA` в правый нижний угол в соответствии с рис.7.
21. Вызвать свойства элемента `"Page " + ALLTRIM(STR( PA` отчёта двойным щелчком мыши или через контекстное меню команда **Properties** и в появившемся диалоговом окне **Report Expression** (Выражение отчёта) в поле ввода **Expression** заменить слово *Page* на слово *Страница*, нажать **ОК**

22. Просмотреть результат выполненной работы: пункт меню **View/Preview**

### Задание

Создать отчет для таблицы *Книги* БД *Библиотека*

1. В диалоговом окне **Report Wizard** (Мастер отчетов) (см. рис.1.) выбрать следующие поля таблицы *Книги*: **Nazvanie, Razdel, Izdat, Godizdan**
2. В диалоговом окне выбора стиля отображения объектов (см. рис.3.) в поле области **Style** (Стиль) выбрать стиль **Executive**
3. В диалоговом окне установки порядка размещения объектов (см. рис.4.) в поле области **Number of Columns** (Количество колонок) указать **1**, в области **Field Layout** (Расположение полей) указать **Rows** (Строка)
4. В диалоговом окне установки критерия упорядочения данных (см. рис.5.) из списка **Available fields or index tag** (Выбранные поля и индексы) перенести в список **Selected fields** (Выбранные поля) поле **Nazvanie**
5. Просмотреть вид создаваемого отчета нажатием кнопки **Preview** (Просмотр).
6. После сохранения запустить отчет *Книги*, она должна иметь вид как на рис.8.

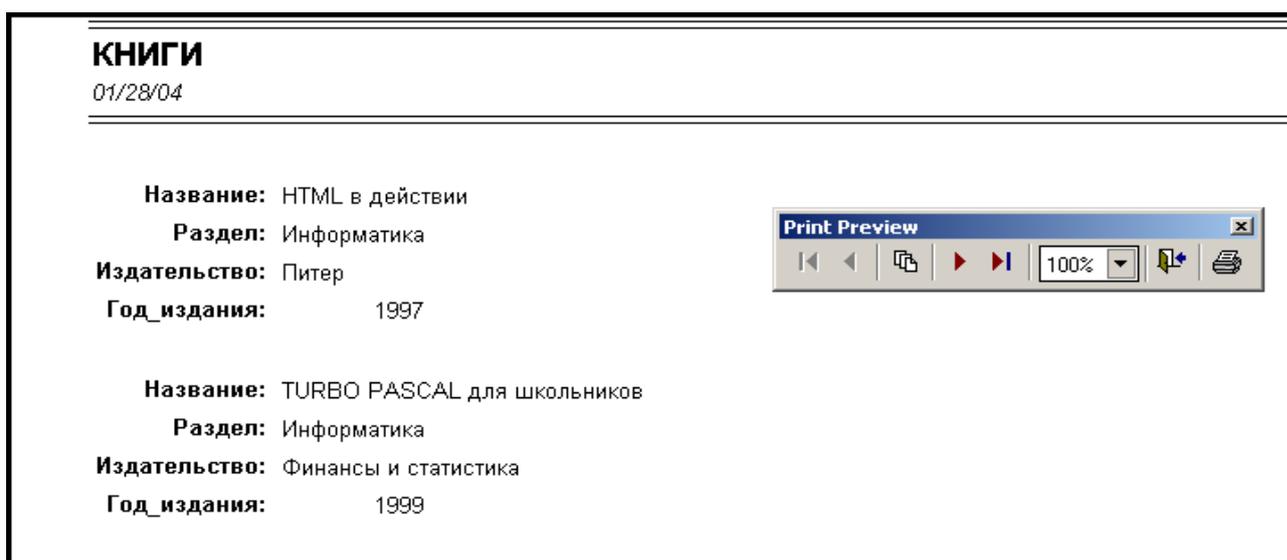


Рис.8. Отчёт *Книги*

## Лабораторная работа №6. Создание запросов в среде Visual FoxPro.

### Теоретическая часть

Одним из основных назначений разработанного приложения является быстрый поиск информации в БД и получение ответов на разнообразные вопросы. Для этих целей в **Visual FoxPro** используются средства, называемые запросами.

При создании запросов в **Visual FoxPro** разработчик может использовать следующие средства:

- **Query Wizard** (Мастер запросов)
- **Query Designer** (Конструктор запросов)
- Команда SELECT языка **Visual FoxPro**

Результатом запроса является таблица, которую можно сохранить в массиве, создаваемой новой таблице, отобразить на экране в режиме Browse (Просмотр) или вывести в виде отчёта.

Для создания запросов можно использовать **Query Wizard** (Мастер запросов), который последовательно запрашивает наименования таблиц, используемых в запросе, перечень полей таблиц, критерий упорядочения и условия фильтрации данных. С помощью **Query Designer** (Конструктор запросов) можно формировать различной сложности критерии для выбора записей из одной или нескольких таблиц, над полями, выбираемыми из таблиц с помощью запросов, можно выполнять различные вычисления.

В верхней части окна **Query Designer** (Конструктор запросов) расположена панель, на которой отображаются используемые в запросе таблицы. Ниже находятся вкладки, предназначенные для выбора полей запроса и формирования условий выборки. Назначение этих вкладок приведено в таблице 1.. Открывая в окне **Query Designer** (Конструктор запросов) необходимые вкладки, можно выполнять следующие действия:

- Выбирать поля результирующей таблицы запроса
- Формировать вычисляемые поля

- Задавать критерии для выборки, группировки и упорядочения данных
- Указывать, куда выводить результаты выборки

Таблица 1. Назначение вкладок окна **Query Designer** (Конструктор запросов)

Вкладка	Назначение
<b>Fields</b> (Поля)	Позволяет указать поля исходных таблиц, выбираемые в результирующий запрос
<b>Join</b> (Объединение)	Позволяет задать условия объединения таблиц
<b>Filter</b> (Фильтр)	Позволяет определить фильтры, накладываемые для выбора записей
<b>Order By</b> (Упорядочение)	Позволяет задать критерии упорядочения данных
<b>Group By</b> (Группировка)	Позволяет задать условия группировки данных
<b>Miscellaneous</b> (Разное)	Позволяет задать дополнительные условия, такие как признак выборки повторяющихся значений, количество или процент выбора данных

Таблица 2. Назначение команд меню **Query** и кнопок панели инструментов **Query Designer** окна **Query Designer** (Конструктор запросов)

Команда меню	Кнопка	Назначение
<b>Add Table</b> (Добавить таблицу)		Добавляет в запрос новую таблицу
<b>Remove Table</b> (Удалить таблицу)		Удаляет выбранную таблицу из запроса
<b>Remove Join Condition</b> (Удалить условие объединения)		Удаляет условие объединения таблиц
<b>Output Fields</b> (Результирующие поля)		Открывает вкладку <b>Fields</b> для выбора полей результирующей таблицы
<b>Join</b> (Объединение)		Открывает вкладку <b>Join</b> для создания условия объединения таблиц
<b>Filter</b> (Фильтр)		Открывает вкладку <b>Filter</b> для задания фильтра
<b>Order By</b> (Упорядочение)		Открывает вкладку <b>Order By</b> для определения критерия упорядочения данных
<b>Group By</b> (Группировка)		Открывает вкладку <b>Group By</b> для определения условия группировки

		данных
<b>Miscellaneous</b> (Разное)		Открывает вкладку <b>Miscellaneous</b> для задания дополнительных параметров запроса
<b>Query destination</b> (Результат запроса)		Открывает диалоговое окно <b>Query destination</b> , в котором указывается куда выводить результат запроса
<b>View SQL</b> (Показать SQL)		Открывает диалоговое окно, в котором отображается SQL-оператор, соответствующий данному запросу
<b>Maximize the table view</b> (Максимизировать панель отображения)		Раскрывает панель отображения используемых в запросе таблиц на весь экран. Повторное нажатие возвращает панели первоначальный размер
<b>Add Join</b> (Добавить условие объединения)		Открывает диалоговое окно <b>Join Condition</b> для задание условия объединения таблиц
<b>Comments</b> (Комментарии)		Открывает диалоговое окно, в котором вы можете ввести краткое описание создаваемого запроса
<b>Run Query</b> (Выполнить запрос)		Запускает запрос на выполнение

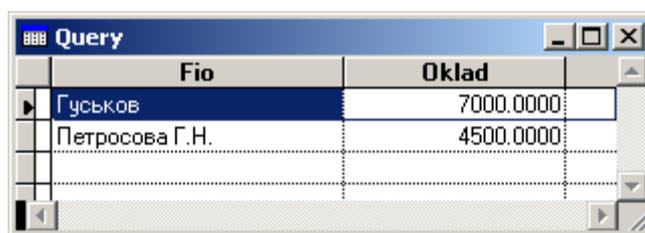
### Цель работы

1. Создать запрос с помощью **Query Wizard** (Мастер запросов)
2. Создать запрос с помощью **Query Designer** (Конструктор запросов)

### Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть проект *Информационная система*
3. Открыть БД *Штат* проекта
4. Создать запрос о сотрудниках, имеющих более одного ребёнка и получающих зарплату менее 9000: в окне **Project Manager** щёлкнуть клавишей мыши на вкладке **Data/Queries/New/Query Wizard**, в появившемся диалоговом окне **Wizard Selection** (Выбор мастера) выбрать пункт **Query Wizard** (Мастер запросов) и нажать кнопку **OK**

5. В появившемся диалоговом окне выбора исходной таблицы и полей в области **Databases and tables** (Базы данных и таблицы) выбрать необходимую БД *итат* и указать таблицу *Сотрудники*. Из области **Available fields** (Имеющиеся поля) выбрать поля **Fio** и **Oklad**. Для перехода к следующему шагу нажать кнопку **Next** (Далее)
6. В появившемся диалоговом окне выбора условий выборки в области **Field** (Поле) верхней строки выбрать поле **Children** (Количество детей) таблицы *Сотрудники*, в области **Operator** (Оператор Условия) выбрать условие *more then*, в области **Value** (Выражение) установить значение **1**, в нижней строке внести следующие данные соответственно: в области **Field** (Поле) – поле **Oklad** (Зарплата), в области **Operator** (Оператор Условия) – *less than*, в области **Value** (Выражение) – **9000**. Нажать кнопку **Next** (Далее)
7. В появившемся диалоговом окне выбора условия сортировки данных из области **Available fields** (Имеющиеся поля) выбрать поле **Fio**. Нажать кнопку **Next** (Далее)
8. Нажать кнопку **Preview** (Просмотр) и просмотреть результат запроса (запрос будет выглядеть в виде простой таблицы см. рис.1.), закрыть таблицу запроса. Нажать кнопку **Finish** (Готово), в появившемся диалоговом окне **Save As** (Сохранить как) указать необходимую папку и присвоить запросу имя *Зарплата дети*



Fio	Oklad
Гуськов	7000.0000
Петросова Г.Н.	4500.0000

Рис.1. Запрос *Зарплата дети*

9. Запустить запрос: в окне **Project Manager** щёлкнуть клавишей мыши на вкладке **Data/Queries/Зарплата дети/Run**
10. Закрыть запрос *Зарплата дети*

11. Создать новый запрос, показывающий количество сотрудников организации, получающих определённую зарплату: в окне **Project Manager** щёлкнуть клавишей мыши на вкладке **Data/Queries/New/New Query**. В появившемся диалоговом окне **Add Table or View** (Добавить таблицу или представление данных) в области **Database** (База данных) выбрать БД *Штам*, в области **Tables in Database** таблицу *Сотрудники*. Нажать кнопку **Add**, при этом в верхней области окна **Query Designer** (конструктор запросов) (см. рис.2.), находящегося позади окна **Add Table or View** (Добавить таблицу или представление данных) появится выбранная таблица. Нажать кнопку **Close** (Заккрыть) для закрытия окна **Add Table or View**.

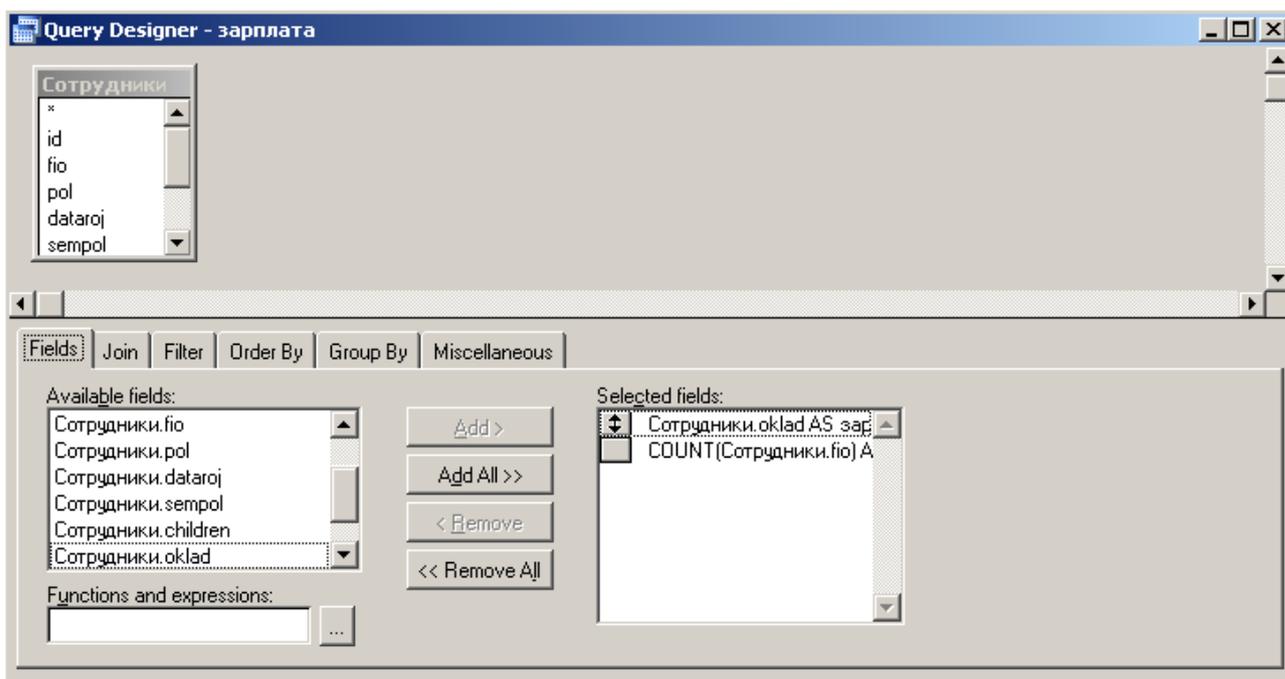


Рис.2. Окно **Query Designer** (конструктор запросов)

12. На вкладке **Fields** (Поля) окна **Query Designer** (конструктор запросов) в области **Available fields** (Имеющиеся поля) выделить поле *Сотрудники.oklad*, оно автоматически перенесётся в область **Function and expressions** (Функции и выражения), нажать расположенную справа от поля кнопку вызова построителя выражения, откроется диалоговое окно **Expression Builder** (Построитель выражения) (см. рис.3.)

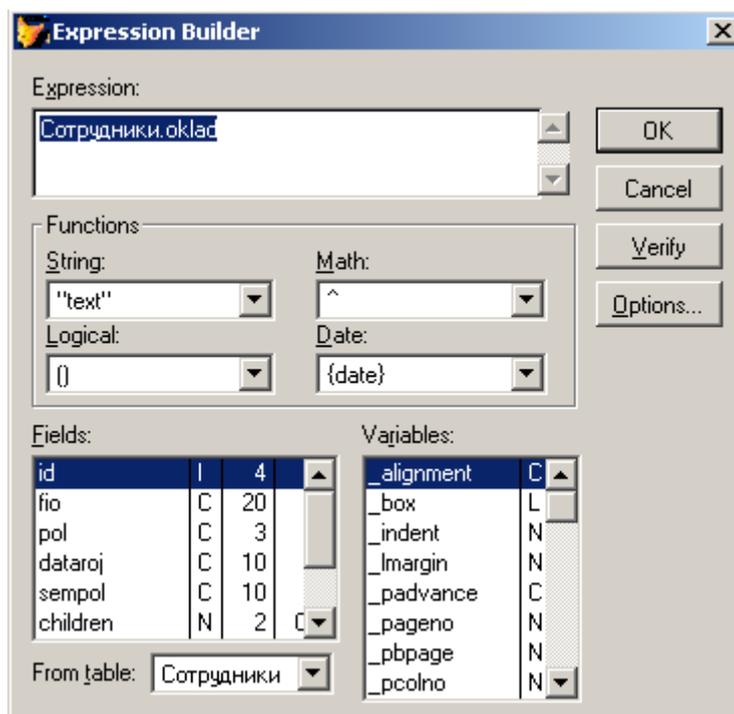


Рис.3. Диалоговое окно **Expression Builder** (Построитель выражения)

13. В поле ввода **Expression** (Выражение) диалогового окна **Expression Builder** (Построитель выражения) используя поля таблиц, расположенные в списке **Fields** (Поля), функции области **Functions** (Функции), и ввод данных с клавиатуры сформировать следующее выражение: Сотрудники.oklad AS Зарплата

14. Для проверки правильности набранного выражения нажать кнопку **Verify** (Проверить), нажать кнопку **OK**

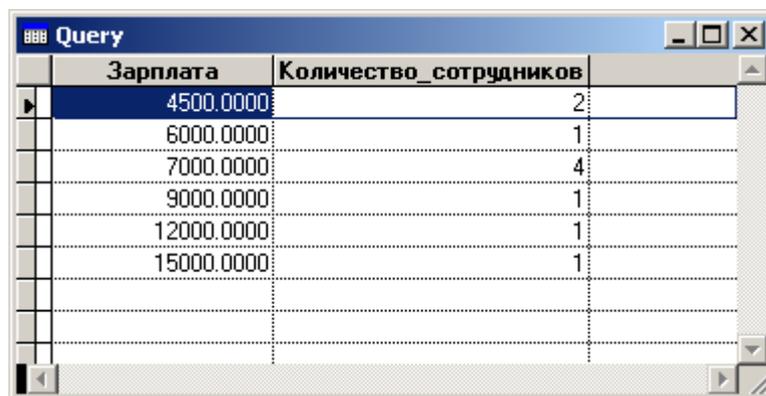
15. Для переноса созданного выражения из области **Function and expressions** (Функции и выражения) в область **Selected Fields** (Выбранные поля) нажать находящуюся между данными областями кнопку **Add** (Добавить)

16. Руководствуясь предыдущими пунктами 12-15 практической части лабораторной работы создать следующее выражение:

COUNT(Сотрудники.fio) AS Количество\_сотрудников

17. Выбрать вкладку **Order By** (Упорядочение) окна **Query Designer** (конструктор запросов) и из области **Selected Fields** (Выбранные поля) в

- область **Ordering criteria** (Критерий упорядочения) перенести поле Сотрудники.oklad
18. Выбрать вкладку **Group By** (Группировка) окна **Query Designer** (конструктор запросов) и из области **Available fields** (Имеющиеся поля) в область **Grouped Fields** (Поля группировки) перенести поле Сотрудники.oklad
19. Выбрать пункт меню **Query** (Запрос) окна программы **Visual FoxPro**, выбрать команду **Comments** (Комментарии) (см. таблицу 2.). В появившемся окне **Comment** (Комментарий) в поле ввода области **Add Comment** (Добавить комментарий) ввести строку: Запрос, показывающий количество сотрудников организации, получающих определённую зарплату. Закрыть окно **Comment** (Комментарий)
20. Запустить созданный запрос: установить курсор в окне **Query Designer** (конструктор запросов), вызвать контекстное меню, выбрать команду **Run Query**, или нажать кнопку [!] на панели инструментов. Закрыть запрос



	Зарплата	Количество_сотрудников
	4500.0000	2
	6000.0000	1
	7000.0000	4
	9000.0000	1
	12000.0000	1
	15000.0000	1

Рис.4. Запрос *Зарплата*

21. Просмотреть SQL-оператор, соответствующий запросу: установить курсор в окне **Query Designer** (конструктор запросов), вызвать контекстное меню, выбрать команду **View SQL** (Показать SQL), откроется диалоговое окно, в котором отображается SQL-оператор, соответствующий данному запросу. Закрыть окно **Query Designer** (конструктор запросов), система предложит сохранить запрос, нажать кнопку **Yes** (Да), в появившемся

диалоговом окне **Save As** (Сохранить как) указать необходимую папку и присвоить запросу имя *Зарплата*

## **Задание**

Создать многотабличный запрос, выводящий в таблицу названия, год издания и автора книг по информатике

1. Создать новый запрос, добавить таблицы *Авторы*, *Автор\_книги* и *Книги* из БД *Библиотека* в окно **Query Designer** (конструктор запросов)
2. Из области **Available fields** (Имеющиеся поля) вкладки **Fields** (Поля) внести в область **Selected Fields** (Выбранные поля) используя построитель выражения **Expression Builder** следующие выражения:

Книги.nazvanie AS название\_книги

Книги.razdel AS раздел

Книги.godizdan AS год\_издания

ALLTRIM(Авторы.familiya)+" "+ALLTRIM(Авторы.imya) AS автор

3. Необходимо задать условие выбора записей из таблицы (Книги только по информатике) по полю *razdel* (Раздел). Перейти на вкладку **Filter** (Фильтр), в области **Field Name** (Имя поля) нажать кнопку раскрытия списка и выбрать поле *Книги.razdel*, в области **Criteria** (Критерии) выбрать значение **==**, в области **Example** (Образец) ввести значение “Информатика”
4. Запустить запрос и посмотреть результат (см. рис.5.), закрыть запрос, закрыть окно **Query Designer** (конструктор запросов), сохранить запрос, присвоив ему имя *Книги по информатике*

Название_книги	Раздел	Год_издания	Автор
Практический курс программирования	Информатика	1983	Фролов Геннадий
Практический курс программирования	Информатика	1983	Илюхин Виктор
TURBO PASCAL для школьников	Информатика	1999	Попов Владимир
HTML в действии	Информатика	1997	Морис Брюс
Самоучитель Visual FoxPro 8.0	Информатика	2003	Омельченко Людмила

Рис.5. Запрос Книги по информатике

## Лабораторная работа № 7. Создание меню приложения в среде Visual FoxPro

### Теоретическая часть

В соответствии со стандартами **Windows** в любом приложении рекомендуется иметь строку меню, которая в **Visual FoxPro** содержит команды, предназначенные для вызова форм, формирования отчётов, запросов и т.д.

Строкой меню называется горизонтальное меню, располагаемое в верхней части экрана. Примером строки меню является основное меню **Visual FoxPro**, а также меню программ, работающих в среде **Windows**. Созданное в конструкторе **Menu Designer** (Конструктор меню) меню может замещать основное меню **Visual FoxPro** или добавляться к нему.

Для создания меню необходимо выполнить следующие действия:

- Открыть окно конструктора меню **Menu Designer** (Конструктор меню)
- Описать вид меню, текст, пункты меню и его атрибуты
- Сгенерировать меню. При этом создаётся программа, которая в результате запускается на выполнение

В **Visual FoxPro** имеется возможность создания меню в виде строки **Menu**, или всплывающего меню **Shortcut**, в котором основные пункты расположены по вертикали.

## Menu Designer (Конструктор меню)

Кнопка	Назначение
<b>Insert</b> (Вставить)	Добавляет в меню новый пункт
<b>Insert Bar</b> (Вставить команды системного меню)	Открывает диалоговое окно <b>Insert System Menu Bar</b> , содержащее команды системного меню <b>Visual FoxPro</b> , позволяя разместить их в создаваемом пользовательском меню
<b>Delete</b> (Удалить)	Удаляет текущий пункт меню
<b>Move Item</b> (Переместить элемент)	Открывает одноимённое диалоговое окно, позволяющее указать пункт меню, в который переносится текущий подпункт
<b>Preview</b> (Просмотр)	Размещает создаваемое меню на экране, позволяя просмотреть его внешний вид

Таблица 2. Типы меню

Тип меню	Назначение
<b>Command</b> (Команда)	При выборе пункта меню данного типа будет выполняться связанная с ним команда
<b>Pad Name</b> (Наименование строки меню)	При выборе пункта меню никаких действий выполняться не будет. Как правило, используется в качестве дополнительного пояснения к меню
<b>Submenu</b> (Подменю)	При выборе пункта меню раскрывается связанное с данным пунктом ниспадающее меню
<b>Procedure</b> (Процедура)	При выборе пункта меню вызывается процедура, определённая для данного пункта меню

Таблица 3. Типы пункта меню

Тип пункта меню	Действие
<b>Submenu</b> (Подменю)	Раскрывается связанное с данным пунктом меню ниспадающее меню
<b>Procedure</b> (Процедура)	Выполняется процедура, определённая в конструкторе меню
<b>Command</b> (Команда)	Выполняется команда, расположенная в поле рядом с типом пункта меню

### Цель работы

1. Создать структуру меню **Menu** в окне **Menu Designer** (Конструктор меню)
2. Определить действия для пунктов меню

3. Сохранить, сгенерировать, запустить меню

### Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть проект *Информационная система*
3. Открыть окно **Menu Designer** (Конструктор меню): в окне **Project Manager** щёлкнуть клавишей мыши на вкладке **Other/Menus/New**, в появившемся диалоговом окне **New Menu** (Новое меню) выбрать пункт **Menu** (Меню), появится окно **Menu Designer** (Конструктор меню) (см. рис.1.)

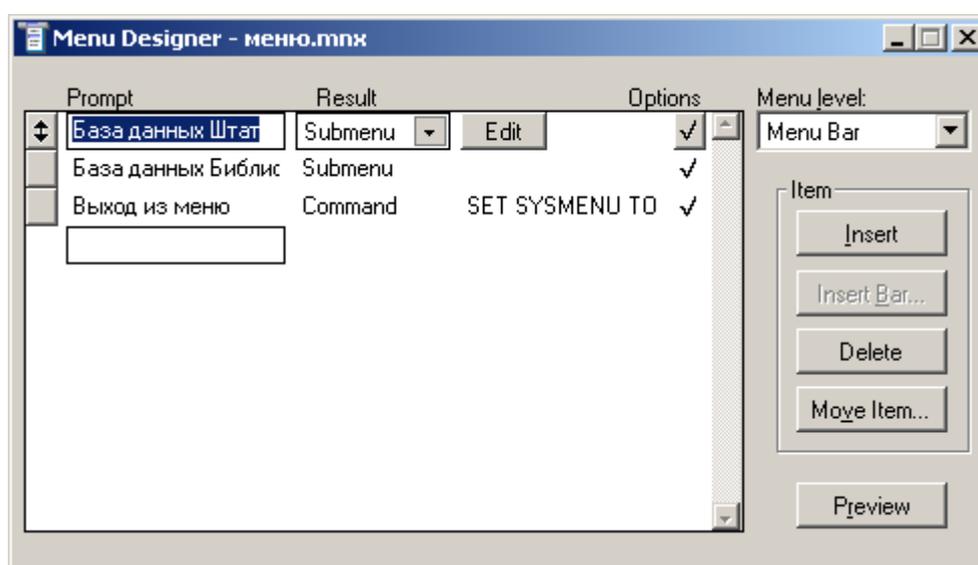


Рис.1. Окно **Menu Designer** (Конструктор меню)

Область конструктора меню, над которой расположены надписи **Prompt** (Приглашение), **Result** (Результат) и **Options** (Опции), предназначена для формирования меню. В поле **Prompt** (Приглашение) вводят наименования пунктов меню, раскрывающийся список **Result** (Результат) используется для указания типа пункта меню, а кнопка **Options** (Опции) открывает диалоговое окно **Prompt Options** (Опции элемента меню), в котором можно определить дополнительные параметры данного элемента меню. В списке **Menu level** (Уровень меню) указывается уровень текущего меню. Слева в конструкторе меню размещены кнопки (см. таблицу 1.)



Рис.2. Схема создаваемого меню *Меню*

4. В поле **Prompt** (Приглашение) ввести наименование первого пункта меню – **База данных Штат**, нажать для перехода на следующее поле клавишу **Enter** или **Tab** на клавиатуре, курсор окажется в списке **Result** (Результат) (см. рис.1.) Указать тип пункта меню **Submenu** (Подменю), нажать кнопку **Options** (Опции), откроется диалоговое окно **Prompt Options** (Опции элемента меню), в котором, в области **Shortcut** (Всплывающее меню) установить курсор в поле **Key Label** (Метка) и нажать комбинацию клавиш **Alt + F1** на клавиатуре. Мы определили комбинацию клавиш быстрого вызова пункта меню **База данных Штат**. Нажать кнопку **OK**
5. Перейти на следующую строку и ввести наименование второго пункта – **База данных Библиотека**, тип пункта меню **Submenu** (Подменю), определить комбинацию клавиш **Alt + F2** быстрого вызова пункта меню **База данных Библиотека** (см. пункт 4. лабораторной работы)

6. Перейти на следующую строку и ввести наименование третьего пункта – Выход из меню, в списке **Result** (Результат) указать тип пункта меню **Command** (Команда), в появившейся справа строке ввода ввести команду – SET SYSMENU TO DEFAULT. Определить комбинацию клавиш **Alt + F4** быстрого вызова действия пункта меню Выход из меню
7. Пункты меню База данных Библиотека и База данных Штат имеют свои подменю. Для создания подменю перейти на строку База данных Библиотека, нажать кнопку **Create** (Создать), находящуюся справа от списка **Result** (Результат), на экране появится пустое окно конструктора меню. Ввести в поле **Prompt** (Приглашение) наименование подпункта – Таблицы базы данных, в списке **Result** (Результат) указать тип пункта меню **Submenu** (Подменю), нажать кнопку **Create** (Создать), так как он тоже имеет подменю (см.рис.2.)
8. В появившемся пустом окне конструктора меню ввести в поле **Prompt** (Приглашение) наименование подпункта – Автор книги, в списке **Result** (Результат) указать тип пункта меню **Procedure** (Процедура), нажать кнопку **Create** (Создать) и в открывшемся окне редактирования процедуры ввести следующие команды:

```
OPEN DATABASE \<Имя вашей папки>\библиотека.dbc  
USE библиотека!автор_книги  
SELECT автор_книги  
BROWSE  
CLOSE DATABASES
```

9. Закрывать окно редактирования процедуры, перейти на следующую строку и ввести наименование следующего пункта – Авторы, в списке **Result** (Результат) указать тип пункта меню **Procedure** (Процедура), нажать кнопку **Create** (Создать) и в открывшемся окне редактирования процедуры ввести следующие команды:

```
OPEN DATABASE \<Имя вашей папки>\библиотека.dbc
USE библиотека!авторы
SELECT авторы
BROWSE
CLOSE DATABASES
```

10. Закрывать окно редактирования процедуры, перейти на следующую строку и ввести наименование следующего пункта – Издательство, в списке **Result** (Результат) указать тип пункта меню **Procedure** (Процедура), нажать кнопку **Create** (Создать) и в открывшемся окне редактирования процедуры ввести следующие команды:

```
OPEN DATABASE \<Имя вашей папки>\библиотека.dbc
USE библиотека!издательство
SELECT издательство
BROWSE
CLOSE DATABASES
```

11. Закрывать окно редактирования процедуры, перейти на следующую строку и ввести наименование следующего пункта – Книги, в списке **Result** (Результат) указать тип пункта меню **Procedure** (Процедура), нажать кнопку **Create** (Создать) и в открывшемся окне редактирования процедуры ввести следующие команды:

```
OPEN DATABASE \<Имя вашей папки>\библиотека.dbc
USE библиотека!книги
SELECT книги
BROWSE
CLOSE DATABASES
```

12. Закрывать окно редактирования процедуры, перейти на следующую строку и ввести наименование следующего пункта – Разделы, в списке **Result** (Результат) указать тип пункта меню **Procedure** (Процедура), нажать кнопку **Create** (Создать) и в открывшемся окне редактирования процедуры ввести следующие команды:

```
OPEN DATABASE \<Имя вашей папки>\библиотека.dbc
```

```
USE библиотека!разделы
```

```
SELECT разделы
```

```
BROWSE
```

```
CLOSE DATABASES
```

13. Закрывать окно редактирования процедуры. Для перехода в меню верхнего уровня использовать список **Menu Level** (Уровень меню): выбрать из выпадающего списка **Menu Level** (Уровень меню) пункт – Базаданных

14. Под строкой Таблицы базы данных ввести наименование следующего пункта – Форма Книги, в списке **Result** (Результат) указать тип пункта меню **Command** (Команда), в появившейся справа строке ввода ввести команду – DO FORM \<Имя вашей папки>\книги.scx

15. Перейти на следующую строку и ввести наименование следующего пункта – Отчёт Книги, в списке **Result** (Результат) указать тип пункта меню **Command** (Команда), в появившейся справа строке ввода ввести команду – REPORT FORM \<Имя вашей папки>\книги.frx PREVIEW

16. Перейти на следующую строку и ввести наименование следующего пункта – Запрос, выводящий на экран названия, год издания и авторов книг по информатике, в списке **Result** (Результат) указать тип пункта меню **Procedure** (Процедура), нажать кнопку **Create** (Создать) и в открывшемся окне редактирования процедуры ввести следующие команды:

```
OPEN DATABASE \<Имя вашей папки>\библиотека.dbc
DO "\<Имя вашей папки>\книги по информатике.qpr"
CLOSE DATABASES
```

17. Закрывать окно редактирования процедуры. Для просмотра создаваемых пунктов меню, нажать кнопку **Preview** (Просмотр), при этом основное меню Visual FoxPro будет заменено создаваемым меню. После просмотра нажать кнопку ОК диалогового окна **Preview** (Просмотр)

Для улучшения внешнего вида, а также для объединения в группы схожих по смыслу команд, в меню можно использовать разделительные линии. Разделительные линии представляют собой пункт меню, в котором в поле ввода **Prompt** (Приглашение) вместо наименования пункта вводятся символы \-. Также **Visual FoxPro** позволяет справа от команд пользовательского меню расположить графическое изображение, для этого необходимо выполнить следующие действия: нажать кнопку **Options** (Опции) для необходимого пункта меню, откроется диалоговое окно **Prompt Options** (Опции элемента меню), в котором, в области **Picture** (Изображение) указать опцию **Resource** (Ресурс), нажать кнопку, располагаемую справа от поля, находящегося под опцией. Откроется диалоговое окно **Insert System Menu Bar** (Вставить из системного меню). Из списка графических изображений, используемых **Visual FoxPro** в системном меню выбрать наиболее соответствующее пункту меню значение и нажать кнопку **ОК**. Выбранное значение переносится в область просмотра области **Picture** (Изображение). Нажать кнопку **ОК** для закрытия окна **Prompt Options** (Опции элемента меню)

18. Для перехода в меню верхнего уровня выбрать из выпадающего списка **Menu Level** (Уровень меню) пункт – **Menu Bar**

19. Указать месторасположение создаваемого меню: пункт меню **Visual FoxPro View/General Options**, откроется диалоговое окно **General Options** (Основные параметры), в области **Location** (Размещение) выбрать пункт

**Append** (Добавить), для добавления создаваемого меню в основное меню  
**Visual FoxPro**

**Задание**

Добавить в пункт меню – База данных Штат подпункты, определить действия для этих подпунктов меню в соответствии с таблицей 4.

Таблица 4. Наименование пунктов меню и определение их действий

<b>Prompt</b> (Приглашение)	<b>Result</b> (Результат)	Команды и процедуры
Таблица Сотрудники	Procedure	<pre>OPEN DATABASE \&lt;Имя вашей папки&gt;\штат.dbc USE штат!сотрудники SELECT Сотрудники BROWSE LAST CLOSE DATABASES</pre>
Форма Сотрудники	Command	<pre>DO FORM \&lt;Имя вашей папки&gt;\сотрудники.scx</pre>
Форма Сотрудники тест	Command	<pre>DO FORM \&lt;Имя вашей папки&gt;\тест.scx</pre>
Отчёт Сотрудники	Command	<pre>REPORT FORM \&lt;Имя вашей папки&gt;\сотрудники.frx PREVIEW</pre>
Запрос, показывающий количество сотрудников организации, получающих определённую зарплату	Procedure	<pre>OPEN DATABASE \&lt;Имя вашей папки&gt;\штат.dbc DO "&lt;Имя вашей папки&gt;\зарплата.qpr" CLOSE DATABASES</pre>
Запрос, показывающий сотрудников, имеющих более одного ребёнка и получающих зарплату менее 9000	Procedure	<pre>OPEN DATABASE \&lt;Имя вашей папки&gt;\штат.dbc DO "&lt;Имя вашей папки&gt;\зарплата дети.qpr" CLOSE DATABASES</pre>

20. Сохранить созданное меню: пункт меню **File/Save As**, в появившемся диалоговом окне указать имя своей папки, куда следует сохранить файл, присвоить имя *Меню* и нажать кнопку **Save** (Сохранить)
21. Необходимо сгенерировать меню: пункт меню **Menu/Generate**, откроется диалоговое окно **Generate Menu** (Генерация меню), нажать кнопку **Generate** (Генерация)
22. После завершения генерации запустить программу меню на выполнение: в окне проекта **Project Manager** установить курсор на наименование созданного меню *Меню* и нажать кнопку **Run** (Запустить)

## Лабораторная работа №8. Создание запросов в Visual FoxPro с помощью команд языка запросов SQL

### Теоретическая часть

Очень важным требованием к реляционным СУБД является наличие мощного и в тоже время простого языка, позволяющего выполнять все необходимые пользователям операции. В последние годы таким повсеместно принятым языком стал язык реляционных БД SQL (Structured Query Language).

В **Visual FoxPro** команды языка SQL (см таблицу 1.) можно вводить непосредственно в окне **Command** (командное окно **Visual FoxPro**) или в диалоговом окне SQL-оператора, вызываемого командой **View SQL** (Показать SQL) контекстного меню окна **Query Designer** (Конструктор запросов).

Таблица 1. Некоторые команды языка SQL

Некоторые команды SQL	Описание
CREATE DATABASE	Создать базу данных
CREATE TABLE	Создать таблицу
DELETE	удалить строки в таблице
INSERT	добавить строки в таблицу
SELECT	отобразить строки из таблиц

<i>UPDATE</i>	изменить строки в таблице
---------------	---------------------------

Типы данных и полей в **Visual FoxPro** и в **SQL** различны (см. таблицу

2.)

Таблица 2. Соответствие типов данных

Сокращенное обозначение	Типы данных <b>Visual FoxPro</b>	Типы данных <b>SQL</b>
C	Character	char
Y	Currency	money
D	Date	datetime
T	Date Time	datetime
B	Double	float
F	Float	float
G	General	image
I	Integer	int
L	Logical	bit
M	Memo	text
M (binary)	Memo binary	image
C (binary)	Character binary	binary
N	Numeric	float

Команды SQL мало зависят от текущего состояния среды **Visual FoxPro**, они сами открывают нужные им базы данных и индексные файлы. В командах SQL могут быть использованы любые функции **Visual FoxPro**. Формирование запросов из базы данных осуществляется командой SELECT, которая является мощным средством обработки запросов. Она располагает массой опций-возможностей и имеет следующий синтаксис:

```

SELECT <что выводится>
      FROM <откуда (источник)> INTO <куда (получатель)>
      WHERE <каким условиям должно отвечать>
      GROUP BY <колонки, по которым выполняется группирование>
      HAVING <условие группировки записей в одну строку>
      ORDER BY <в каком порядке выводить данные>

```

Более полный формат команды SELECT:

```

SELECT

```

[DISTINCT] [<псевдоним>.] <выражение> [AS <колонка>]  
 [, [<псевдоним>.] <выражение> [AS <колонка>]...]  
 FROM <БД> [, <БД>...]  
 [[INTO <получатель>]/TO FILE <файл>  
 [ADDITIVE]/TO PRINTER]]  
 [NOCONSOLE] [PLAIN] [NOWAIT]  
 [WHERE <условие связи>  
 [AND <условие связи>...]  
 AND/OR <условие отбора>  
 AND/OR <условие отбора>...]]]  
 GROUP BY <колонка> [, <колонка>...]  
 [HAVING <условие отбора >]  
 [ORDER BY <колонка> [ASC/DESC][, <колонка> [ASC/DESC]...]]

### Цель работы

1. Создать базу данных, таблицы и связи с помощью команд языка SQL
2. Редактировать таблицы и содержимое таблиц с помощью команд языка SQL
3. Создать запросы к БД с помощью команды SELECT языка SQL

### Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть командное окно программы: пункт меню **Window/Command window**
3. Для создания базы данных ввести в командное окно следующие команды языка SQL:

CLOSE ALL

&& Закрывать все файлы приложений

CLEAR

&& Очистить окно программы **Visual**

**FoxPro**

CREATE DATABASE mydata1  
<mydata1>

&& Создать БД и присвоить ей имя

Необходимо создать следующие таблицы, ключевые поля, связи:

Таблица *Salesman*:

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)
Name	Type	Width	Index
SalesID	Integer	4	↑ Primary
SaleName	Character	20	

Таблица *Customer*:

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)
Name	Type	Width	Index
SalesID	Integer	4	↑ Regular
CustID	Integer	4	↑ Primary
CustName	Character	20	↑ Unique
CustAmt	Money (Currency)	4	
City	Character	15	

4. Для создания таблицы *Salesman* с ключевым полем SalesID ввести в командное окно следующие команды языка SQL:

```
CREATE TABLE Salesman ;  
(SalesID i PRIMARY KEY, ;  
SaleName C(20))
```

5. Для создания таблицы *Customer* с ключевым полем CustId, с индексным полем CustName (индекс типа Candidate) и с индексным полем SalesId, для создания отношения один-ко-многим с ключевым

полем таблицы *Salesman*, ввести в командное окно следующие команды языка SQL:

```
CREATE TABLE Customer ;  
  (SalesID i, ;  
   CustID i PRIMARY KEY, ;  
   CustName c(20) UNIQUE, ;  
   CustAmt y(4), ;  
   City c(15), ;  
   FOREIGN KEY SalesId TAG SalesId REFERENCES Salesman)
```

6. Просмотреть созданную базу данных: в командном окне ввести команду

```
MODIFY DATABASE
```

7. Для ввода в таблицу *Customer* новой строки в командном окне набрать следующие команды языка SQL:

```
INSERT INTO Customer (SalesID, CustID, CustName, CustAmt, City);  
VALUES (1, 1, "Магомедов", 3000, "Махачкала")
```

8. Аналогично заполнить таблицу используя данные из таблицы 3.

Таблица 3. Содержимое таблицы *Customer*

SalesID	CustID	CustName	CustAmt	City
2	2	Сидоров	3000	Москва
3	3	Петров	5000	Саратов
4	4	Иванов	2500	Москва
5	5	Гаджиев	4700	Махачкала

9. Для удаления строк из таблицы в командном окне набрать следующие команды языка SQL:

```
DELETE FROM customer WHERE City = && Пометить записи на  
"Москва" удаление
```



4	Омельченко
5	Кондратьев

2. Вывести из таблицы *Salesman* всех продавцов, фамилии которых начинаются на букву "О"
3. Удалить из таблицы *Salesman* всех продавцов, фамилии которых начинаются на букву "О"
4. Создать следующую таблицу *Workers*:

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)
Name	Type	Width	Index
NN	Integer	4	↑ Primary
FIO	Character	20	
Pol	Character	3	
Dataroj	datetime		
City	Character	15	

5. Вывести общее количество работников в таблице *Workers*

### Лабораторная работа №9. Построение концептуальной схемы **Теоретическая часть**

Концептуальная схема строится на основе технического задания, в котором отражаются информационные потребности пользователей, описание наиболее существенных объектов и их свойств, возможный документооборот. Построение адекватной концептуальной схемы является залогом создания эффективной базы данных.

#### **Цель работы**

1. Создать EER-модель предметной области "Учебный процесс" в первом приближении
2. Установить все типы связей и грамотно расположить все компоненты схемы для её большей наглядности

### **Условия схемы:**

1. Так как диаграмма будет достаточно сложной, то на первом этапе можно не указывать атрибуты (или указать только ключевые).
2. Для упрощения концептуальной модели сделаны некоторые упрощения: не сказано кто и где ведёт определённый вид занятий, а также выделены только лекции, все остальные виды занятий, а также формы контроля отнесены в один раздел Прочие.
3. Для упрощения диаграммы символы (ромбы) для обозначения связей не используются, но вместо них на линиях связи помещаются точки, разделяющие линии связи на две части. Каждая такая часть используется для обозначения степени участия объекта, прилегающего с соответствующей стороны к точке в связи. Если объект подсоединяется к точке двойной линией, то степень его участия в связи – полная, если одинарной, то степень его участия в связи – частичная.

### **Пояснение предметной области.**

1. Факультет объединяет ряд кафедр. Среди кафедр есть общеобразовательные и кафедры, выпускающие специалистов. Кафедра может готовить специалистов по нескольким специальностям. Лица, работающие на кафедре, относятся к категории Сотрудник.
2. Сотрудники делятся на две части: инженерно-лаборантский состав и преподавательский состав. Каждый преподаватель должен относиться к одной из категорий: ассистент, старший преподаватель, доцент, профессор. Кроме ассистентов, все категории входят в лекторский состав. Лектор имеет право читать лекции и руководить дипломниками.
3. Преподавание студентам каждой специальности ведётся в соответствии с утверждённым учебным планом, где определены дисциплины, подлежащие изучению, виды занятий, отводимое под них количество часов, а также форма контроля знаний по дисциплине.

4. По данной специальности обучение ведётся на одном и более курсов. На каждом курсе может быть одна или несколько групп одной и той же специальности. В каждой группе обучается ряд студентов.

#### **Технические подробности.**

1. В рассматриваемой модели проведена спецификация ряда объектов с образованием суперклассов и подклассов.
  - a. Суперкласс Кафедра включает два подкласса: Выпускающая (ответственная за подготовку специалистов) и Общая (принимает участие в ведении общеобразовательных дисциплин).
  - b. Суперкласс Сотрудник включает два подкласса: Преподаватель (ведёт занятия) и Инж\_лабор\_состав (обеспечивает процесс обучения).
  - c. В свою очередь Преподаватель является суперклассом для двух подклассов: Лектор (ведет хотя бы один курс лекций) и Ассистент (ведет практические и лабораторные занятия).
  - d. Лектор также является суперклассом с тремя подклассами: Старший\_преподаватель, Доцент и Профессор, т.к. для двух последних необходимо указывать номер соответствующего диплома.
  - e. Суперкласс Студент включает подкласс Дипломник, так как студент, работающий над дипломной работой, нуждается в руководителе.
  - f. Суперкласс Дисциплина имеет два подкласса Лекции и Прочие (практические, лабораторные, зачеты, экзамены, коллоквиумы и т.д.)
2. Объекты, выделенные в рассматриваемой модели, а также некоторые относящиеся к ним атрибуты приведены в таблице спецификации требований пользователя. Из состава атрибутов выделены те из них, которые можно использовать в качестве первичного ключа. У объектов-подклассов, полученных путём специализации, позиция

первичного ключа не заполнена. Т.к. подкласс наследует все атрибуты суперкласса, в частности и идентификационные, то, следовательно, первичные ключи у них одинаковые.

**Таблица спецификации требований пользователя.**

<b>Объект</b>	<b>Атрибуты</b>	<b>Первичный ключ</b>
Факультет	код_факультета наименование_факультета	код_факультета
Кафедра	код_кафедры наименование_кафедры	код_кафедры
Общая		
Выпускающая		
Сотрудник	табельный_номер ФИО стаж	Табельный_номер
Инж_лабор_состав	должность	
Преподаватель	педагогический_стаж	
Лектор		
Ассистент		
Старший_преподаватель		
Доцент	диплом_доцента	
Профессор	диплом_профессора	
Специальность	шифр_специальности наименование_специальности	Шифр_специальности
Курс	номер_курса	Номер_курса
Учебный_план	шифр_специальности	
Дисциплина	код_дисциплины наименование_дисциплины	Код_дисциплины
Лекции	количество_часов	
Прочие		
Группа	код_группы староста количество_студентов	Код_группы
Студент	номер_зачётной_книжки ФИО адрес	Номер_зачётной_книжки
Дипломник	тема_диплома	

**Задание.** По приведённой в таблице информации и сопутствующим пояснениям требуется построить EER-диаграмму. Необходимые важнейшие связи для указанных объектов, соответствующие реальным, установлены и приведены в следующей таблице. Для каждой связи указаны показатель кардинальности и степень участия сторон в связи.

<b>Связь</b>	<b>Объекты</b>	<b>Показатель кардинальности</b>	<b>Степень участия</b>
Входит	Факультет Кафедра	1:N	Полная Полная
Работает	Кафедра Сотрудник	1:N	Полная Частичная
Готовит	Выпускающая Специальность	1:N	Полная Полная
Утвержден	Специальность Учебный план	1:N	Полная Полная
Проходит	Специальность Курс	1:N	Полная Полная
Состоит	Курс Группа	1:N	Полная Полная
Ведет	Преподаватель Дисциплина	M:N	Полная Полная
Включает	Учебный_план Дисциплина	1:N	Полная Полная
Учиться_в	Группа Студент	1:N	Полная Полная
Обучается_по	Дисциплина Студент	M:N	Полная Полная
Читает_по	Лектор Лекции	1:N	Полная Частичная
Руководит_дипломником	Лектор Дипломник	1:N	Полная Частичная

**Необходимые пояснения.**

В некоторых случаях реализуется частичная связь: не каждый Сотрудник работает на кафедре (есть другие структуры ВУЗа), не каждый Лектор руководит дипломником или читает лекции.

## **Лабораторная работа №10. Перевод концептуальной схемы в структуру базы данных.**

### **Теоретическая часть**

Концептуальная схема выражается с помощью специального графического представления. Объекты изображаются в прямоугольниках и обозначаются существительными, атрибуты изображаются в овалах, ключевой атрибут подчёркивается. Связи отображаются ромбами и стрелками, обозначаются глаголами. Стрелки обозначают кардинальность связи.

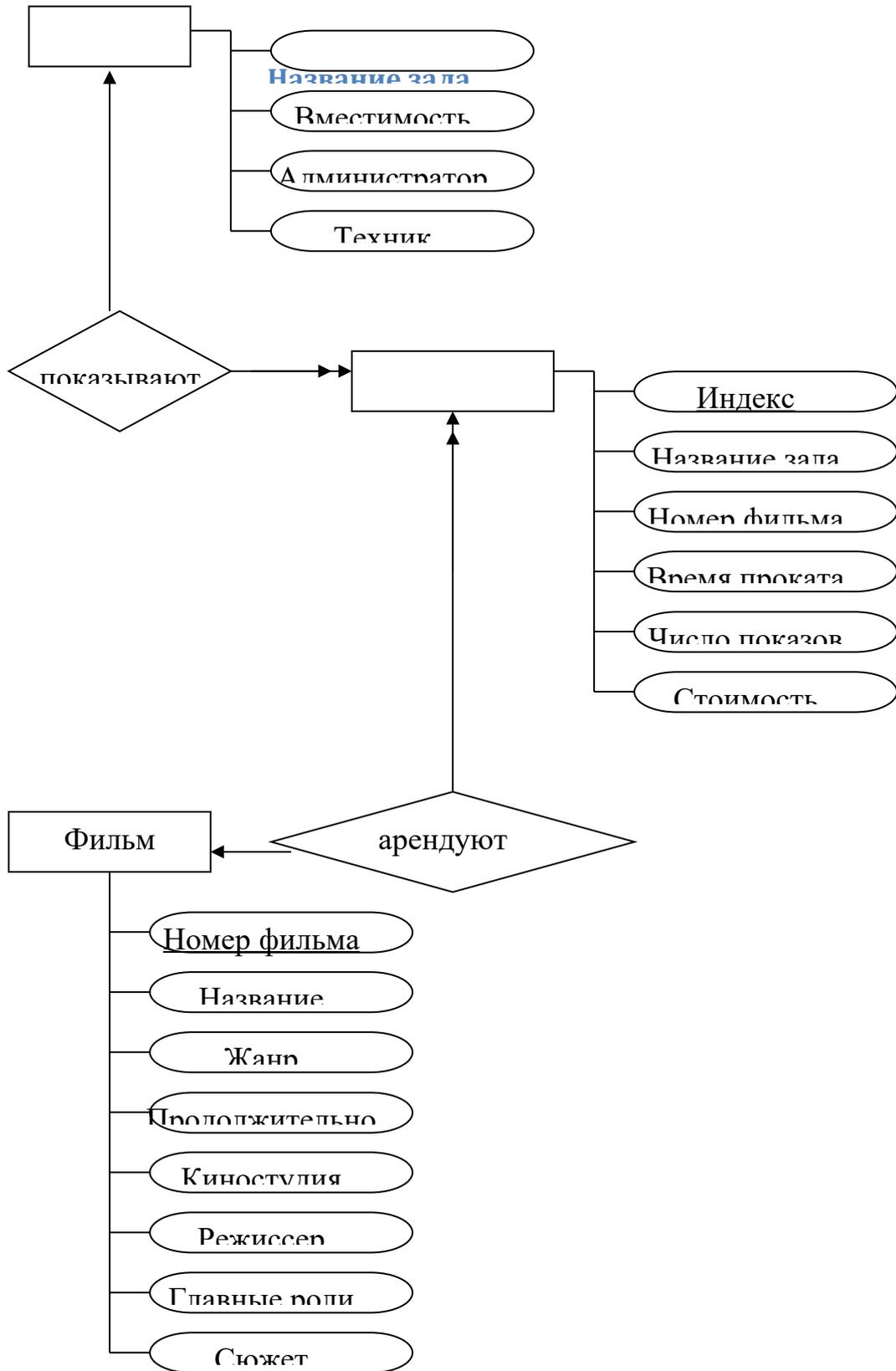
### **Цель работы**

1. Создать базу данных на основе имеющейся концептуальной схемы
2. Выбрать сформировать таблицы, типы полей и установить связи между таблицами
3. Заполнить таблицы экспериментальными данными
4. Составить отчёт по связанным таблицам о работе кинотеатра

### **Задание**

Ориентируясь по представленной схеме, создать базу данных, содержащую три связанные таблицы, установить связи между таблицами, наполнить таблицы экспериментальными данными (не менее 5 записей в таблице). Составить отчёт о работе кинотеатра, который должен содержать: название фильма и его жанр, зал, в котором фильм показывают, число показов.

## Концептуальная схема базы данных "Кинотеатр"



## Лабораторная работа №11. Работа с базами данных в Delphi.

### Теоретическая часть

Среда Delphi 7 работает с таблицами баз данных, которые физически располагаются на диске. Поэтому такие таблицы называются *физическими*. Для доступа к данным, содержащимся в физических таблицах, применяются наборы данных.

**Набором данных** называется совокупность записей, выделенных по определенным правилам из одной или нескольких физических таблиц базы данных

Набор данных, по определению, не представляет собой физическую таблицу, поэтому будем называть набор данных *логической таблицей*. Именно логическими таблицами работает большинство стандартных компонент Delphi 7. Можно представлять себе взаимодействие физической и логической таблицы по аналогии, как физического файла и файловой переменной. Среда Delphi 7 имеет несколько стандартных компонентов для работы с наборами данных. К их числу относятся такие компоненты, как Table, Query, StoredProc и др. Доступ к данным в Delphi 7 обеспечивает класс TDataSet, который представляет наборы данных в виде совокупности строк и столбцов. Строки являются *записями*, а столбцы — *полями* таблицы базы данных. Класс TDataSet обеспечивает возможность редактирования набора данных, а также предоставляет средства для перемещения (навигации) по записям. Многие из свойств, событий и методов класса TDataSet являются абстрактными. Они не могут использоваться непосредственно классом TDataSet, а лишь в его классах-потомках. Прямым потомком класса TDataSet является класс TBDEDataSet. Этот класс инкапсулирует в себе функциональные возможности Borland Database Engine (BDE) — процессора баз данных фирмы Borland. BDE — это совокупность файлов динамических библиотек (DLL) и драйверов, которые отвечают за доступ к данным. Класс TBDEDataSet имеет класс-потомок TDBDataSet, в котором определены дополнительные свойства и методы, обеспечивающие возможность

связывания набора данных с физическими таблицами базы данных. Рассмотрим два компонента Table и Query, которые являются потомками класса TDBDataSet. Изучим их общие свойства на примерах операций с наборами данных. Эти компоненты расположены на вкладке **VBE** палитры компонентов Delphi 7.

### Цель работы

1. Создать приложение, связанное с базой данных.
2. Реализовать навигацию и поиск в базе данных.
3. Реализовать вычисляемые поля.
4. Реализовать фильтрацию данных и обработку запросов.

### Практическая часть.

#### Часть 1.

1. Создайте свою папку в папке Мои документы. Все остальные файлы (и база данных, и проект Delphi) необходимо сохранять в этой папке.
2. Запустите Microsoft Access.
3. Создайте новую базу данных, назовите её "Куратор".
4. В режиме Конструктора создайте таблицу "Студенты":

Название поля	Тип поля	Длина поля	Ключевое
№Студента	счётчик	–	да
ФИО	текстовое	40	
Адрес	текстовое	25	
Год рождения	числовое	целое	
Национальность	текстовое	15	
Пол	текстовое	3	
Характеристика	поле МЕМО		
Количество пропусков	числовое	целое	
Средний балл	числовое	одинарное	

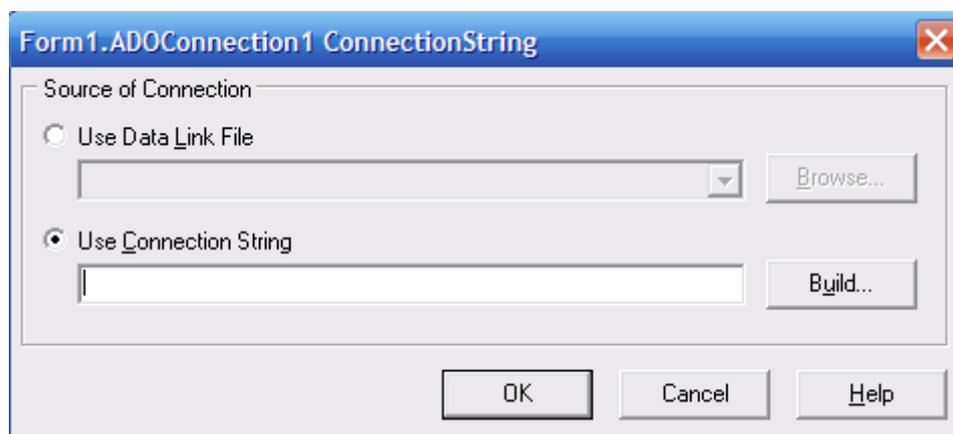
Для поля Средний балл надо установить длину поля Одинарное с плавающей точкой, и Число десятичных знаков = 2.

Поле №Студента сделайте ключевым. Не забудьте сохранить таблицу.



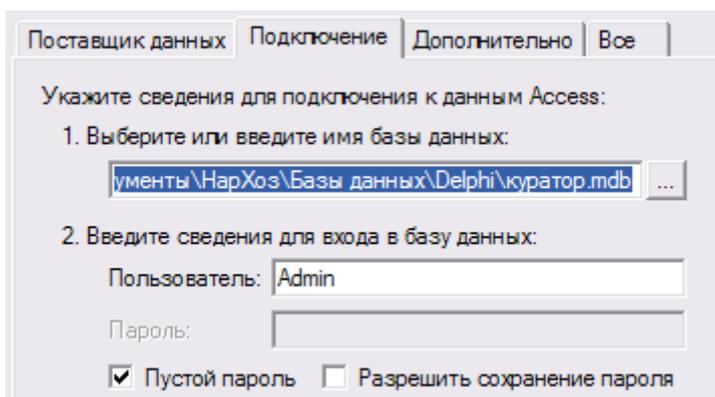
	Имя поля	Тип данных
🔑	№студента	Счетчик
	ФИО	Текстовый
	Адрес	Текстовый
	Год рождения	Числовой
	Национальность	Текстовый
	Пол	Текстовый
	Характеристика	Поле МЕМО
	Количество пропусков	Числовой
▶	Средний балл	Числовой

5. Закройте Microsoft Access.
6. Запустите Delphi. Так как базы данных фирмы Microsoft построены на технологии ADO, то будет лучше использовать компоненты, специально предназначенные для работы по этой технологии.
7. Прежде всего требуется установить связь приложения Delphi с базой данных. Для этого используется компонент ADOConnection. Можно настраивать связь для каждого компонента отдельно, но лучше установить её один раз с помощью указанного компонента, а затем остальными ссылаться на него. Поместите ADOConnection  на форму, он находится на вкладке ADO. Выберите свойство ConnectionString, нажмите на троеточие, появится окно



Нажмите кнопку Build. В появившемся списке поставщиков связи выберите Microsoft Jet 4.0 OLE DB Provider, нажмите кнопку Далее (или выберите

вкладку Подключение). В появившемся окне в строке для имени базы данных также нажмите троеточие:

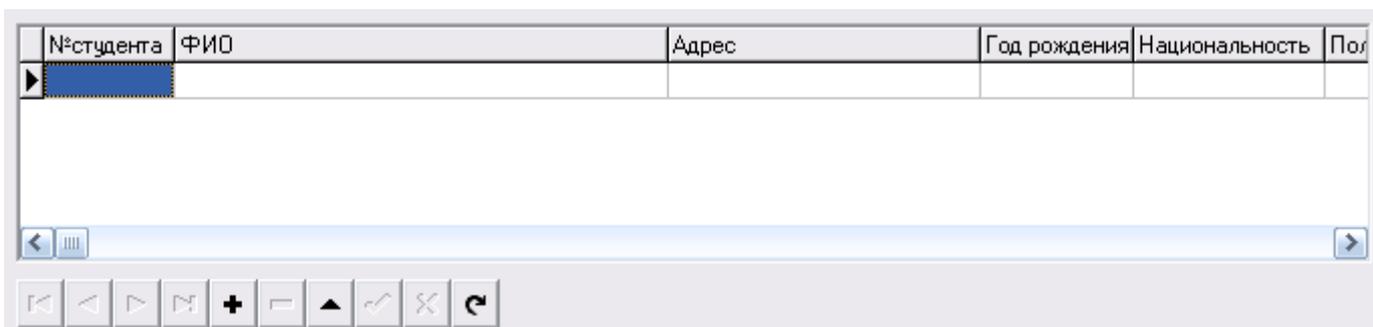


найдите свою папку и выделите в ней файл своей базы данных Куратор, путь до него должен подставиться в эту строку. Все остальные настройки оставьте без изменений. Нажмите ОК в этом и следующем за ним окне.

8. Далее у этого компонента необходимо свойство LoginPrompt перевести в значение false, чтобы не выдавался запрос пароля на каждое действие пользователя.
9. Сохраните свой проект под именем curator, а файл формы (Unit) под именем unit\_k.
10. Теперь поместим компонент, позволяющий получать данные из таблицы: ADOTable . Он находится на той же вкладке ADO. Выберите его свойство Connection (оно красного цвета) и в выпадающем списке выберите созданное ранее подключение ADOConnection1. Затем в свойстве TableName в выпадающем списке выберите название своей таблицы Студенты.
11. Затем с вкладки Data Access на форму поместите компонент DataSource , который позволяет использовать полученные из таблицы данные в приложении. В его свойстве DataSet выберите AdoTable1.
12. Теперь разместите компоненты, позволяющие просматривать данные в приложении. Прежде всего поместите на форму компонент DBGrid , который позволяет просматривать данные в виде таблицы. В его

свойство DataSource подставляем DataSource1. Растяните его на всю ширину формы, чтобы отображать как можно больше полей.

13. Аналогично установите компонент DBNavigator.
14. Чтобы активизировать форму, но при этом не забивать оперативную память на этапе разработки, выберем форму и для события OnCreate (дважды щелкните на пустой строке в Инспекторе объектов напротив этого события) напишите следующий код: `ADOTable1.Active:=true;`
15. Чтобы освобождать оперативную память после работы необходимо в обработчике события OnDestroy формы написать код: `ADOTable1.Active:=false;`
16. Сохраните приложение и запустите его. В результате должна появиться пустая таблица с теми названиями столбцов, которые были заданы в Microsoft Access. Все поля не помещаются при выводе, должен автоматически появиться ползунок горизонтальной прокрутки.



17. Можно заметить, что поле Мемо не просматривается в режиме таблицы. Для заполнения и просмотра данных удобнее использовать отдельные поля (режим формы заполнения). Для этого необходимо выбрать ADOTable1, щелкнуть по нему правой клавишей и из выпадающего меню выбрать Fields Editor.
18. В открывшемся окне необходимо также щелкнуть правой клавишей и выбрать пункт Add all fields – будут добавлены все поля базы данных.
19. Затем необходимо выделить все поля – щелкнуть правой клавишей и выбрать Select all. Затем, удерживая нажатой левую клавишу мыши, перетащить поля в область формы. Будут добавлены соответствующие

метки и поля Edit. Их следует разместить в соответствии с рисунком (некоторые поля будут находиться гораздо ниже, надо прокрутить ползунок вертикальной прокрутки формы).

The screenshot shows a Windows application window titled "Form1". At the top left, there are two "ADO" icons and a navigation icon. Below them is a table with two columns and two rows. A horizontal toolbar with various navigation icons is positioned below the table. The form contains several data entry fields: "№ студента" (DBEdit1) and "ФИО" (DBEdit2) are on the same line; "Адрес" (DBEdit3) and "Год рождения" (DBEdit4) are on the next line; "Национальность" (DBEdit5) and "Пол" (DBEdit) are on the following line. Below these is a "Характеристика" section with a large text area (DBMemo1). To the right of this area are two more fields: "Количество пропусков" (DBEdit7) and "Средний балл" (DBEdit8).

20. После расстановки объектов сохраните проект и запустите его. Внесите несколько записей с помощью формы заполнения и навигатора. Убедитесь, что после занесения эти же данные отображаются в соответствующих полях таблицы.

Form1

№студента	ФИО	Адрес	Год рождения	Национальность	Пол
*	Казиев Абузар Керимович	пр. Шамиля, 22, кв. 93	1989	аварец	муж

№студента  ФИО   
 Адрес  Год рождения   
 Национальность  Пол   
 Характеристика

Количество пропусков   
 Средний балл

## Часть 2.

1. Запустите Delphi и откройте созданный ранее проект curator.
2. Добавьте новую форму, на которой будут располагаться новые компоненты: File → New → Form. Сохраните её по имени unit\_z.
3. Свяжите формы между собой: выберите первую форму, затем File → Use unit ... и в появившемся списке выберите модуль unit\_z. Аналогично, выбрав вторую форму надо добавить unit\_k.
4. В первой форме добавьте 2 кнопки BitBtn (вкладка Additional). У первой кнопки свойство Kind сделайте равным bkOk, а в обработчике нажатия введите команды:
 

```
form1.Hide; form2.Show;
```
5. У второй кнопки свойство Kind сделайте равным bkClose.
6. Первая форма будет загружаться при запуске программы (основная форма), из неё можно попасть во вторую форму. Выходить из программы следует только из первой формы.

7. Вся дальнейшая работа будет происходить на второй форме. Перейдите на вторую форму и поместите на неё кнопку BitBtn. В обработчике нажатия кнопки введите следующую команду:

```
form2.Hide; form1.Show;
```

8. Сохраните проект и запустите, проверьте правильность выполнения переходов.

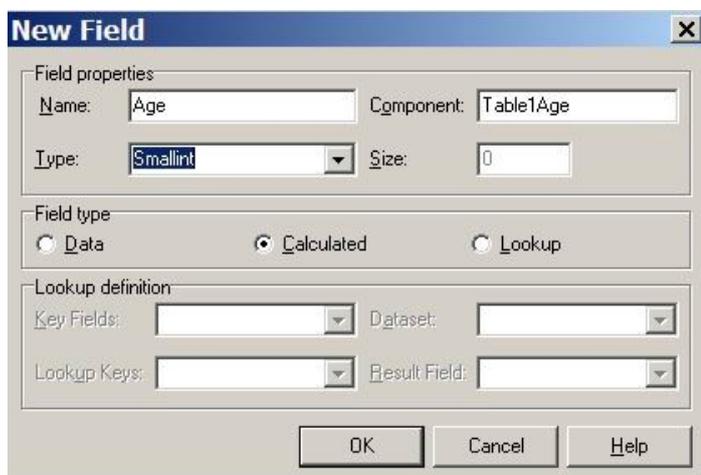
9. Добавьте компонент DBGrid для отображения данных из таблицы. Свяжем этот компонент с уже готовым DataSource1 на первой форме: надо просто выбрать этот компонент в выпадающем списке свойства DataSource.

10. Нам нужны не все данные, поэтому нажмите правой клавишей на компоненте DBGrid1, выбираем Columns Editor ... В выпавшем списке полей надо удалить ненужные поля. Должны ОСТАТЬСЯ: ФИО, Год рождения, Национальность, Пол, Средний балл.

11. Чтобы невозможно было испортить данные в DBGrid, выберите его свойство Options, нажмите плюс, стоящий справа от него. Раскроется список подсвойств. Выберите dgEditing и сделайте его false. Это не позволит вносить изменения в DBGrid.

12. Теперь создадим вычисляемое поле – его нет в исходной таблицы, но для получения значений в нём используются табличные записи. Добавим в приложении поле, отображающее возраст студента. Перейдите на первую форму, сделайте двойной щелчок на Table1, появится Редактор Полей. Щелкните в Редакторе Полей правой кнопкой мыши и в выпавшем меню выберите New field (Новое поле). Появится окно добавления поля. В разделе Field Properties (свойства поля) укажите имя поле в строке Name – Age, в строке Type (тип данных) – SmallInt. В следующем разделе Field Type переключите радиокнопки в позицию Calculated.

Щелкните Ok, вы вернётесь в окно Инспектора Полей, в котором появится новое поле Age. Задайте для него в Инспекторе Объектов значение "Возраст" для свойства **DisplayLabel**.



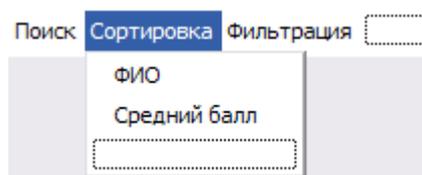
13. Теперь напишите обработчик, вычисляющий значение в созданном поле. Для этого выберите Table1 и в Инспекторе Объектов на странице событий щелкните на событии OnCalcFeilds. В появившейся заготовке процедуры напишите следующий код:

```
procedure TForm1.ADOTable1CalcFields(DataSet: TDataSet);
var Year1, Year2, Month, Day: Word;
begin
    DecodeDate(Date, Year1, Month, Day);
    Year2:=ADOTable1DSDesigner4.Value;
    ADOTable1Age.Value:=Year1-Year2;
end;
```

14. Перейдите на вторую форму, щелкните на DBGrid правой клавишей, выберите Columns Editor ... В открывшемся окне щелкните правой клавишей и выберите Add. Добавиться новое поле. В Object TreeView выберите добавленное поле, и в свойстве fieldName из выпадающего списка выберите Age. Закройте Columns Editor.

15. Сохраните и запустите приложение, убедитесь, что возраст рассчитывается верно.

16. Теперь осуществим поиск определённой записи по введённому значению поля ФИО. Для этого предварительно создадим меню: с вкладки Standard добавьте компонент MainMenu. Щелчком правой клавишей мыши по MainMenu вызовите контекстное меню и выберите Menu Designer ... Затем создайте такую структуру меню:



Закройте Menu Designer.

17. Также на форму дополнительно поместите поле Edit с вкладки Standard.

18. Теперь в Object TreeView выберите объект пункт меню Поиск и для его события OnClick напишите следующий код:

```
procedure TForm2.N1Click(Sender: TObject);
begin
  If not Form1.ADOTable1.Locate
    ('ФИО',Edit1.Text,[])
  then ShowMessage('Значений не найдено');
end;
```

В результате выполнения этого пункта меню будет осуществляться поиск строки, введённой в поле Edit, по полю ФИО до первого полного совпадения. Если строки отличаются хотя бы регистром, то будет выведено сообщение: "Значений не найдено".

19. Перейдём к сортировке. Сначала подготовим базу данных для сортировки по полю ФИО. Для этого сохраните проект и закройте Delphi. Запустите Microsoft Access и откройте свою базу данных Куратор. Если будет выдаваться предупреждение системы безопасности "Заблокировать небезопасные выражение?" – нажмите НЕТ, а в следующем окне нажмите ОТКРЫТЬ. Выберите свою таблицу

Студенты и откройте её в конструкторе. Выделите поле ФИО. Поменяйте его свойства как показано на рисунке:

Обязательное поле	Да
Пустые строки	Нет
Индексированное поле	Да (Допускаются совпадения)

20. Закройте Microsoft Access и снова запустите Delphi. Вызовите свой проект.

21. Для обработчика кнопок ФИО меню Сортировка напишите следующий код:

```
procedure TForm2.N3Click(Sender: TObject);
begin
    Form1.ADOTable1.IndexFieldNames:='ФИО';
end;
```

22. Теперь реализуем фильтрацию по полю Национальность. Для соответствующего пункта меню напишите обработчик:

```
procedure TForm2.N5Click(Sender: TObject);
begin
    Form1.ADOTable1.Filter:=
'Национальность="'+Edit1.Text+'";
    Form1.ADOTable1.Filtered:=true;
end;
```

При выполнении будут отображаться только те записи о студентах, национальность которых полностью совпадает с введенной в текстовое поле, т.е. например "аварец" и "аварка" – разные национальности с точки зрения поисковой команды.

**Задание:** Создайте сортировку по полю Средний балл. Задайте фильтрацию по полю Пол и возраст. Измените соответствующим образом меню программы.

## **Лабораторная работа №12. Индивидуальный проект**

**Цель работы:** Создать базу данных на основе технического задания и собственное приложение, обрабатывающее эту базу данных.

### **Общие положения.**

Предлагаемые студентам задания являются краткими техническими заданиями, на основе которых должны быть составлены полноценные базы данных. Работа студентов предполагает выполнение следующих этапов:

1. Сбор фактического материала по теме задания.
2. Уточнение технического задания по фактическому собранному материалу.
3. Составление концептуальной схемы на основе уточнённого технического задания.
4. Реализация структуры базы данных по концептуальной схеме.
5. Наполнение таблиц базы данных фактическими данными, собранными ранее.
6. Оформление работы базы данных, то есть создание форм ввода данных, необходимых запросов и отчётов.
7. Защита своего проекта.

На основе выполнения и защиты проекта студент аттестуется и получает допуск к сдаче экзамена.

### **Материалы итогового тестирования по дисциплине.**

Во всех предлагаемых ниже вопросах правильный вариант ответа стоит первым, для того чтобы не вводить дополнительные текстовые обозначения или таблицу правильных ответов.

1. База данных – это ...
  - a) именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области;
  - b) набор связанных между собой таблиц с данными;

- c) набор языковых и программных средств, предназначенных для ведения документации предприятия;
  - d) система сбора, обработки и хранения информации.
2. Единое вместительное хранилище разнообразных данных и описаний их структур, которое после своего определения, осуществляемого отдельно и независимо от приложений, используется одновременно многими приложениями – это ...
- a) База данных;
  - b) Система управления базами данных;
  - c) Файл базы данных;
  - d) Таблица с метаданными.
3. СУБД – это ...
- a) совокупность языковых и программных средств, предназначенная для создания, ведения и совместного использования БД многими пользователями;
  - b) совокупность связанных между собой баз данных, доступных для многих пользователей;
  - c) совокупность связанных между собой данных, распределённых в системе;
  - d) совокупность управляющих баз данных.
4. Что не является свойством базы данных?
- a) Простота доступа;
  - b) Безопасность;
  - c) Целостность;
  - d) Эффективность.
5. Что означает целостность базы данных?

- a) В каждый момент сведения, содержащиеся в базе являются непротиворечивыми;
  - b) Все связанные данные находятся целиком в одной таблице;
  - c) Данные в базе данных являются защищёнными от несанкционированного доступа и преднамеренного изменения;
  - d) База данных реагирует на запросы времени за минимально возможное время, т.е. имеет минимизированное время отклика.
6. Что не является преимуществом использования СУБД?
- a) Требуемая высокая квалификация работников;
  - b) Минимизация избыточности данных;
  - c) Независимость прикладных программ от данных;
  - d) Непротиворечивость данных и контроль их целостности.
7. Что не является компонентом СУБД?
- a) Операционная система;
  - b) Пользователи;
  - c) Аппаратное обеспечение;
  - d) Программное обеспечение.
8. Администраторы баз данных отвечают за ...
- a) физическое проектирование и реализацию, обеспечение целостности и безопасности;
  - b) отбор оптимального варианта воплощения логической схемы;
  - c) создание приложений, обеспечивающих реализацию дополнительных возможностей;
  - d) занесение данных в базу, целенаправленное и регулярное обновление.
9. Какой уровень не входит в трёхуровневую архитектуру данных
- a) Физический;

- b) Внешний;
- c) Внутренний;
- d) Концептуальный.

10. Основное назначение трёхуровневой архитектуры данных

- a) обеспечение логической и физической независимости данных;
- b) управление данными во внешней памяти;
- c) поддержка целостности данных;
- d) поддержка языков баз данных.

11. Что такое транзакция?

- a) последовательность операций над базой данных, рассматриваемых СУБД как единое целое;
- b) словарь данных или системный каталог;
- c) функция управления параллельным доступом;
- d) вспомогательная функция поддержки обмена данными.

12. Что не относится к функциям СУБД?

- a) поддержка функционирования локальной сети в реляционной базе данных;
- b) поддержка обмена данными;
- c) поддержка целостности данных;
- d) поддержка независимости данных от фактической структуры данных.

13. Логически в современной реляционной СУБД можно выделить: ядро

СУБД, компилятор языка БД, подсистему поддержки времени выполнения, контроллер доступа к данным, набор утилит. Какой компонент в этом определении лишний?

- a) контроллер доступа к данным;
- b) ядро СУБД;

- с) компилятор языка БД;
- д) подсистему поддержки времени выполнения.

14. ... – манипулирует предназначенными для хранения данных файлами и отвечает за распределение доступного дискового пространства.

- а) контроллер файлов;
- б) процессор запросов;
- в) контроллер словаря;
- г) контроллер базы.

15. Среди языков, поддерживаемых современными СУБД, описательным является

- а) язык определения данных;
- б) язык управления данными;
- в) язык манипулирования данными;
- г) язык изменения данных.

16. Для модификации существующей схемы данных используется ...

- а) язык определения данных;
- б) язык управления данными;
- в) язык манипулирования данными;
- г) язык изменения данных.

17. Какая операция не относится к операциям языка манипулирования данными?

- а) описание данных;
- б) селекция данных;
- в) модификация данных;
- г) включение данных.

18. Какой язык, из поддерживаемых СУБД, используется для занесения данных, изменения или выборки данных?
- а) язык манипулирования данными;
  - б) язык управления данными;
  - в) язык определения данных;
  - г) язык изменения данных.
19. Жизненный цикл базы данных включает следующие этапы: планирование разработки БД – определение требований к системе – сбор и анализ требований пользователей – проектирование базы данных – разработка языка запросов – разработка приложений – реализация – загрузка данных – тестирование – эксплуатация и сопровождение. Уберите лишнее.
- а) разработка языка запросов;
  - б) определение требований к системе;
  - в) разработка приложений;
  - г) тестирование.
20. Какой модели данных не существует?
- а) проектная модель;
  - б) даталогическая модель;
  - в) инфологическая модель;
  - г) концептуальная модель.
21. По структуре представления данных выделяют следующие виды моделей. Уберите лишнюю.
- а) логико-предметная модель;
  - б) сетевая модель;
  - в) иерархическая модель;
  - г) реляционная модель.

22. Отдельные объекты, которые входят в данный объект-тип, называются ...
- а) экземплярами объекта;
  - б) реализациями объекта;
  - в) единичными объектами;
  - г) объектами данных.
23. Множество значений атрибута называется ...
- а) доменом;
  - б) многозначным атрибутом;
  - в) характеристикой объекта;
  - г) базовым атрибутом.
24. Поименованная характеристика объекта, с помощью которой задаются его свойства, называется ...
- а) атрибутом;
  - б) доменом;
  - в) объектом;
  - г) отношением.
25. Выберите верное утверждение.
- а) атрибуты, с помощью которых можно идентифицировать экземпляр объекта, называют ключами;
  - б) атрибуты, являющиеся уникальными для базы данных, образуют ключевую связь;
  - в) набор уникальных атрибутов образует первичный ключ;
  - г) из множества первичных ключей выбирается уникальных ключ.
26. Что такое мощность связи?
- а) максимальное количество экземпляров одного объекта, связанного с одним экземпляром другого объекта;

- б) количество связанных отношений, участвующих в выбранной связи;
- в) мощность ключевого поля (первичное, кандидат, уникальное), по которому осуществляется связь;
- г) количество ключевых полей, задействованных в связи.

27. Что такое генерализация?

- а) процесс сведения различий между объектами к минимуму путём выделения их общих характеристик;
- б) процесс увеличения различий между отдельными экземплярами объекта за счёт выделения их отличительных характеристик;
- в) процесс уменьшения атрибутов объекта путём их объединения в комбинированные атрибуты;
- г) процесс увеличения атрибутов объекта за счёт объединения разных объектов отношения.

28. Различают следующие главные составляющие любой модели данных. Уберите лишнее.

- а) концептуальная часть;
- б) структурная часть;
- в) управляющая часть;
- г) классы ограничений целостности.

29. Модель данных – это ...

- а) некоторая абстракция, которая отражает самые важные аспекты функционирования рассматриваемой предметной области;
- б) правила задания видов структур и их взаимодействия между собой;
- в) набор определений различных операций над структурами базы данных;
- г) набор правил реализации представления данных и отношений между ними по правилам реляционной базы данных.

30. Выделяют следующие категории моделей данных. Уберите лишнюю.
- а) трёхуровневые модели;
  - б) объектные модели;
  - в) модели на основе записей;
  - г) физические модели.
31. Какая математическая структура лежит в основе сетевой модели?
- а) направленный граф;
  - б) интегральная схема;
  - в) дифференциальная схема;
  - г) базисный граф.
32. В реляционной базе данных отношение – это
- а) двумерная таблица;
  - б) связь между таблицами;
  - в) связь между атрибутами;
  - г) вложенные атрибуты.
33. В чём состоит правило поддержки недействительных значений?
- а) Независимо от типа данных должна быть возможность представления отсутствующих данных;
  - б) Недействительные данные должны удаляться в автоматическом режиме;
  - в) Недействительное значение атрибута делает недействительным весь кортеж, содержащий атрибут;
  - г) Недействительное значение ключевого атрибута делает недействительным весь кортеж.
34. В чём состоит правило независимости распространения.
- а) Реляционная СУБД не должна зависеть от потребностей конкретного клиента;

- б) Любая реляционная база должна распространяться независимо от системы управления базами данных;
- в) Любая реляционная база данных не зависит от установленной операционной системы и систем связи;
- г) Реляционная СУБД не должна зависеть от файловой системы.

35. Любое отношение обладает следующими свойствами. Уберите лишнее.

- а) Набор атрибутов строго ограничен;
- б) Отношение имеет имя, отличное от имён других отношений;
- в) Каждый компонент кортежа является простым, он не состоит из группы значений;
- г) Все кортежи должны быть различны.

36. Что означает "категорная целостность"?

- а) ключевые атрибуты обязательно должны иметь определённые значения (не NULL);
- б) совокупность категорий базы данных должна принадлежать одному отношению;
- в) ключевые атрибуты образуют целостную категорию отношения;
- г) категории должны принадлежать атрибутам со значениям не NULL.

37. Выберите неверное утверждение.

- а) Нормализация – это приведение реляционной каждой базы к пятой нормальной форме, делая последовательные преобразования от первой нормальной формы и выполняя процедуру Бойса-Кодда;
- б) Нормализация – это выявление различного рода зависимостей между атрибутами и устранение нежелательных из них;
- в) Цель процедуры нормализации – устранить нежелательные функциональные зависимости и аномалии (включения, удаления и модификации), а также минимизировать избыточность;

г) Процесс нормализации – это формальный метод анализа отношений на основе первичных и потенциальных ключей и существующих функциональных зависимостей.

38. Какая команда не является командой редактирования содержимого таблицы?

- а) Browse;
- б) Append;
- в) Edit;
- г) Change;

39. Какая команда системного меню позволяет удалять помеченные записи?

- а) Remove Deleted Records;
- б) Delete Selected Records;
- в) Delete Selective Records;
- г) Restructure Deleted Records.

40. Укажите правильную последовательность вхождения.

- а) Проект – База данных – Таблица – Запись;
- б) База данных – Проект – Таблица – Запись;
- в) База данных – Проект – Запись – Таблица;
- г) База данных – Запись – Проект – Таблица.

41. Какой тип индексов может содержать повторяющиеся значения?

- а) Regular;
- б) Unique;
- в) Candidate;
- г) Primary.

42. Какая опция не позволяет изменять значения полей первичного ключа или ключа-кандидата в родительской таблице, если в дочерней таблице имеется хотя бы одна запись, содержащая ссылку на изменяемую запись
- а) Restrict;
  - б) Cascade;
  - в) Ignore;
  - г) Abort.
43. Наглядным средством представления информации, обеспечивающим удобство ввода, являются ...
- а) формы;
  - б) таблицы;
  - в) отчёты;
  - г) запросы.
44. Во внешней памяти СУБД хранит следующие разновидности объектов (Уберите лишний):
- а) Наборы исключаяющих индексов;
  - б) Строки отношений;
  - в) Управляющие структуры;
  - г) Журнальную информацию.
45. С точки зрения СУБД база данных выглядит как ...
- а) набор записей;
  - б) набор страниц;
  - в) набор данных;
  - г) набор индексов.
46. Как называется технология быстрого прямого доступа к хранимой записи на основе значения ключевого поля

- а) Хеширование;
- б) Технология запросов;
- в) Технология прямого обращения;
- г) Администрирование.

47. Основной файл, содержащий данные отношения – ...

- а) индексированный файл;
- б) индексный файл;
- в) файл прямого доступа;
- г) первичный файл.

48. Описательный язык, с помощью которого описывается предметная область, называется ...

- а) язык определения данных;
- б) язык манипулирования данными;
- в) язык запросов;
- г) декларативный язык.

49. Какую операцию не реализует язык манипулирования данными?

- а) определения;
- б) селекции;
- в) удаления;
- г) модификации.

50. Какой параметр инструкции SELECT определяет, из каких таблиц или запросов должны извлекаться данные?

- а) FROM;
- б) SELECT;
- в) WHERE;
- г) HAVING.

51. Какой параметр инструкции SELECT определяет, какие поля таблицы должны быть обработаны?

- a) SELECT;
- б) WHERE;
- в) FROM;
- г) INNER JOIN.

52. Какой параметр инструкции SELECT определяет условие отбора данных?

- a) WHERE;
- б) FROM;
- в) ORDER BY;
- г) SELECT.

53. Какой параметр инструкции SELECT позволяет отсортировать данные в выходном наборе.

- a) ORDER BY;
- б) HAVING BY;
- в) SORTING BY;
- г) GROUP BY.

54. Назовите оператор ввода данных.

- a) INSERT;
- б) INPUT;
- в) READ;
- г) UPDATE.

55. Транзакция – это

- а) неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными, рассматриваемая СУБД как единое целое;
- б) последовательность операторов, воздействующая на всю базу данных единым образом;
- в) последовательность операторов, представленных в виде процедуры, воздействующих на неделимые данные;
- г) последовательность операторов, одновременно воздействующих на базу, таким образом, действие на базу является неделимым.

56. Для чего предназначен оператор COMMIT в теле транзакции?

- а) Фиксирует изменения в базе данных;
- б) Отменяет внесённые изменения при сбое транзакции;
- в) Отменяет внесённые изменения по требованию пользователя;
- г) Фиксирует состояние перед выполнением транзакции.

57. Что не является основным свойством транзакции

- а) Безопасность;
- б) Атомарность;
- в) Согласованность;
- г) Изолированность.

58. Как называется специальный тип хранимой процедуры, которая автоматически выполняется при каждой попытке изменить защищаемые ею данные?

- а) Триггер;
- б) Транзакция;
- в) Селектор;
- г) Сейвмейкер.

59. Последовательность компилированных операторов SQL, хранящихся в системной базе СУБД – это
- а) Хранимая процедура;
  - б) Управляющая информация;
  - в) Системный протокол;
  - г) Программа-администратор.
60. Что не относится к прямым обязанностям администратора БД?
- а) Наполнение БД реальными данными;
  - б) Резервное копирование;
  - в) Мониторинг производительности;
  - г) Создание и удаление баз данных.
61. Что такое "роль"?
- а) Поименованный набор полномочий;
  - б) Набор функций, доступных администратору;
  - в) Набор ключей, хранящийся в системном реестре;
  - г) Поименованный набор стандартных функций.
62. Хранилища данных, ориентированные на долговременное хранение информации и работу с нестандартными запросами называются ...
- а) Информационные системы;
  - б) Оперативные системы;
  - в) Интеллектуальные системы;
  - г) Корпоративные системы.

**Задания для самостоятельной работы студентов.**

**Перечень вопросов по дисциплине для самостоятельного изучения.**

1. XML-серверы.
2. Проектирование на физическом уровне.

3. Гипертекстовые базы данных.
4. Мультимедийные базы данных.
5. Управление складами данных.
6. Распределённые базы данных.
7. Коммерческие базы данных.
8. OLAP-технологии.

### **Тематика рефератов и творческих работ студентов.**

1. Базы данных в Internet и управление ими.
2. Система управления базами данных Microsoft Access.
3. Database Desktop – возможности и настройка.
4. Базы знаний.
5. Инфологическое моделирование баз данных.
6. Организация доступа к удалённым базам данных.
7. Методика разработки базы данных по определенной области.
8. Реляционные базы данных – правила построения отношений.

### **Задачи для самостоятельного решения.**

В данном пункте приведены примерные технические задания, которые выносятся на итоговые лабораторные занятия в качестве индивидуальных проектов.

### **Зачисление абитуриентов.**

**В базе должны быть представлены данные:**

1. Анкетные данные абитуриента: фамилия, имя, отчество, адрес прописки и реальный адрес, дата рождения и возраст, какую школу окончил, национальность.
2. Совокупность оценок на вступительных экзаменах.
3. Готовность учиться на договорной (платной) основе.

**В базе данных должны быть реализованы возможности:**

1. Выбор для зачисления заданного количества абитуриентов. Рассмотреть вариант, когда количество студентов набравших проходной балл больше количества приёмных мест.
2. Формирование для собеседования списка тех, кто набрал предельный проходной балл, но не может платить за образование.

### **Справочник потребителя (служба быта).**

**В базе должны быть представлены данные:**

1. Данные о предприятиях бытового обслуживания города: название, разряд, адрес и телефоны, специализация, перечень оказываемых услуг, форма собственности, часы и дни работы.

**В базе данных должны быть реализованы возможности:**

1. Подбор предприятий по заданной услуге.
2. Поиск предприятия по названию, телефону, специализации.
3. Сводные данные о предприятиях одной специализации.

### **Склад.**

**В базе должны быть представлены данные:**

1. Данные о товарах, хранящихся на складе: наименование, единицы измерения, цена единицы, количество, дата последнего завоза.
2. Данные о поставщиках: название фирмы, адрес, контактный телефон, список поставляемых товаров.

**В базе данных должны быть реализованы возможности:**

1. Регистрация поступления товара (формирование приходной накладной).
2. Регистрация отгрузки товара (формирование расходной накладной).

3. Вывод инвентарной ведомости (наличие и количество различных видов товаров от различных поставщиков).

### **Личная библиотека.**

**В базе должны быть представлены данные:**

1. Выходные данные книги (авторы, название, издательство, год издания, количество страниц).
2. Раздел библиотеки (специальная литература, хобби, домашнее хозяйство, беллетристика и так далее).
3. Происхождение и наличие книги в данный момент, субъективная оценка книги.

**В базе данных должны быть реализованы возможности:**

1. Выбор книг по произвольному запросу: по автору, по названию, по издательству, по разделу, по субъективной оценке.
2. Инвентаризация библиотеки: данные обо всех книгах по разделам и наличию.

### **Справочник абитуриента.**

**В базе должны быть представлены данные:**

1. Данные о ВУзах: наименование, адрес, перечень специальностей, конкурс прошлого года по каждой специальности (дневной, вечерней, заочной форм обучения), размер оплаты при договорном обучении.

**В базе данных должны быть реализованы возможности:**

1. Выборки всех данных по данной специальности.
2. Выборки всех данных по данному ВУЗу.
3. Поиск минимального конкурса по данной специальности. Поиск минимального конкурса вообще.

## **Справочник владельца видеотеки.**

**В базе должны быть представлены данные:**

1. Данные видеофильмов: название, студия, жанр, год выпуска, режиссер, исполнители главных ролей, краткое содержание, субъективная оценка фильма. Факт наличия фильма в видеотеке.
2. Данные о носителях: кассета, CD, DVD. Входит сборник на диске или отдельно.

**В базе данных должны быть реализованы возможности:**

1. Оформление выдачи и возврата кассеты.
2. Поиск фильма по названию, по режиссёру, по актёрам, по жанру.
3. Инвентаризация видеотеки: данные о всех фильмах и наличие.

## **Крылатые фразы.**

**В базе должны быть представлены данные:**

1. Справочник пословиц, поговорок, афоризмов, каламбуров, других словесных курьёзов.

**В базе данных должны быть реализованы возможности:**

1. Классификация по авторам и источникам.
2. Поиск по темам и ключевым словам.

## **Ресторан.**

**В базе должны быть представлены данные:**

1. Рецепты блюд: раскладка, рецепт приготовления, разряд (салат, десерт и т.д.).
2. Данные о стоимости продуктов: единица измерения (в рецептах), стоимость.

**В базе данных должны быть реализованы возможности:**

1. Формирование меню на определённое количество персон: первое, второе, салат, десерт, – и определение его стоимости.
2. По заданной стоимости подобрать блюда на заданное количество персон.

### **Магазин компьютерных дисков.**

**В базе должны быть представлены данные:**

1. Данные о дисках: название (английский и русский вариант), сфера применения (системный, утилиты, игровой по жанрам), разработчик, издатель, цена.

**В базе данных должны быть реализованы возможности:**

1. Поиск дисков по названию и ключевым словам, по издателю и разработчику.
2. Заказ дисков (формирование расходной накладной), получение дисков (формирование приходной накладной).

### **Справочник владельца компьютера.**

**В базе должны быть представлены данные:**

1. Данные о комплектующих: наименование, вид (процессор, винчестер и т.д.), цена.

**В базе данных должны быть реализованы возможности:**

1. Подбор конфигурации, определение цены.
2. Подбор списка комплектующих на заданную стоимость.

### **Телепрограмма.**

**В базе должны быть представлены данные:**

1. Программа телепередач нескольких каналов на неделю по дням и часам.
2. Разные жанры телепередач: новости, спорт, фильмы, сериалы и т.п.

**В базе данных должны быть реализованы возможности:**

1. Составление программы просмотра по определённому вкусу.
2. Вывод программы определённого канала в заданном промежутке времени.

### **Справочник лекаря.**

**В базе должны быть представлены данные:**

1. Данные о болезнях: название, симптомы, лечебные процедуры, список рекомендуемых лекарств с указанием требуемого количества, противопоказания.
2. Данные о лекарствах: название, действие, показания и противопоказания.

**В базе данных должны быть реализованы возможности:**

1. Формирование рецепта: диагноз и выписанные лекарства.
2. Спецификация лекарств по болезням, их взаимозаменяемость.

## **Раздел 4. Разработка и эксплуатация удаленных баз данных.**

1. Технологические методы защиты информации.
2. Программы для создания зашифрованной области на жестком диске DriveCrypt Plus Pack3.
3. Установка привилегий доступа к данным.
4. Назначение сервера, рабочей станции.
5. Классификация SQL-серверов и особенности их применения.
6. Преимущества использования компонентов построения запросов по сравнению с компонентами прямого доступа к таблице.
7. Назначение и правила использования элементов базы данных.

8. Уровни представления данных: уровень сущностей и уровень атрибутов.
9. Приемы графического проектирования структуры базы данных.
10. Технология автоматического создания на сервере файла базы данных по спроектированной модели.
11. Состав параметров при установлении привилегий.
12. Значение резервного копирования для сохранности данных.
13. Оптимизация базы данных путем создания и немедленного восстановления.
14. Различные методы сортировки, поиска и фильтрации данных.
15. Понятие исключительной ситуации при работе клиентского приложения с удаленной базой данных.
16. Мягкий и жесткий выход из исключительной ситуации.
17. Способы перехвата исключительных ситуаций.
18. Отличие исключительных ситуаций, возникающих на сервере и в клиентском приложении.
19. Архитектуры удаленных баз данных.
20. Принципы разработки и эксплуатации систем управления удаленными базами данных.
21. Технологии доступа к удаленным базам данных.
22. Защита информации и управления доступом к данным.
23. Восстановление данных в критических ситуациях.

### *Задания лабораторных занятий*

**Тема: Разработка программ управления удаленными базами данных с применением операторов SQL.**

1. Внедрение операторов SQL в прикладные программы.
2. Выполнение однострочных и многострочных запросов с помощью внедренных операторов SQL и курсоров.
3. Модификация таблиц баз данных с помощью курсоров.

**Тема: Программные инструментальные средства автоматизации проектирования систем.**

1. Создание модели процессов в BPWin (IDEF0).
2. Создание модели данных с помощью ERWin.
3. Создание объектной модели с помощью Rational Rose.

**Тема: Проектирование серверной части приложения.**

1. Применение СУБД Access для разработки проекта удаленных баз данных.
2. Создание серверного приложения преобразованием проекта базы данных формата Microsoft Access в формат SQL Server.

3. Проектирование и модификация таблиц командами SQL.
4. Создание пользовательских представлений.
5. Разработка хранимых процедур.
6. Разработка триггеров.

**Тема: Общие принципы проектирования клиентской части баз данных.**

1. Основные требования к разработке пользовательского интерфейса.
2. Разработка пользовательского интерфейса средствами визуального проектирования MS Access.
3. Технологии разработки форм пользовательских приложений.

**Тема: Ориентация развития СУБД на расширенную реляционную модель.**

1. Основные направления совершенствования реляционных баз данных.
2. Генерация систем баз данных, ориентированных на приложения.
3. Оптимизация запросов, управляемых правилами.
4. Поддержка динамической информации и темпоральных запросов.

**Тема: Web-технологии в разработке удаленных баз данных.**

1. Статические и динамические Web-страницы.
2. Требования к интеграции удаленных баз данных со средой Web.
3. Методы интеграции удаленных баз данных в среду Web.
4. Генерация Web-страниц визуальными средствами Microsoft Access.

**Тема: Работа с СУБД MySQL.**

1. Администрирование базы данных.
2. Интерфейсы для работы с MySQL.
3. Создание нового пользователя.

**Тема: Язык запросов СУБД MySQL.**

1. Операторы определения данных.
2. Операторы манипулирования данными.
3. Выполнение запросов к базе данных.

**Тема: Индексы и производительность баз данных.**

1. Принципы работы индексов.
2. Создание индексов.
3. Недостатки индексов.

**Тема: Технологические методы защиты информации.**

4. Авторизация пользователей.
5. Резервное копирование и восстановление.
6. Шифрование.
7. Программы для создания зашифрованной области на жестком диске DriveCrypt Plus Pack3.

**Задания для итогового контроля остаточных знаний по дисциплине.**

***Архитектура БД – это...***

- а) организация взаимодействия аппаратных средств;
- б) схемы, характеризующие базы данных с разных сторон с целью определить оптимальное построение информационной системы;
- в) внутренняя структура СУБД, обеспечивающая доступ ко всем компонентам БД;
- г) открытый интерфейс доступа к базам данных, т.е. взаимодействие процессора (ядра) базы данных Jet с внешними источниками данных.

***1. Модели данных – это...***

- а) организация взаимодействия аппаратных средств;
- б) схемы, характеризующие базы данных с разных сторон с целью определить оптимальное построение информационной системы;
- в) внутренняя структура СУБД, обеспечивающая доступ ко всем компонентам БД;
- г) открытый интерфейс доступа к базам данных, т.е. взаимодействие процессора (ядра) базы данных Jet с внешними источниками данных.

***2. Основной принцип модели клиент-сервер применительно к технологии управления БД заключается в разделении функций стандартного интерактивного приложения на пять групп. Какие из следующих вариантов не входят в эти группы?***

- а) Функции вывода данных;
- б) Прикладные функции, определяющие основные алгоритмы решения задач приложения (Business Logic);
- в) Функции обработки данных внутри приложения (Data Base Logic);
- г) Служебные функции, играющие роль связок между функциями первых четырех групп.

***3. Клиентская часть не включает в себя:***

- а) Презентационную логику;
- б) Бизнес-логику;
- в) Логика обработки данных;
- г) Систему транзакций.

***4. Какое из следующих свойств не характерно для модели файлового сервера?***

- а) Презентационная логика и бизнес-логика располагаются на клиентской части.
- б) На сервере располагаются файлы с данными и поддерживается доступ к этим файлам.
- в) Недостатком данной модели является очень большая загрузка сервера.
- г) Каждая файловая команда вызывает перекачку блока информации на компьютер клиента, а СУБД анализирует полученную информацию.

**5. Какая из следующих моделей относится к трехуровневой?**

- а) Модель файлового сервера;
- б) Модель сервера баз данных;
- в) Модель сервера приложений;
- г) Модель файлового приложения.

**6. CALS-технологии – это...**

- а) совокупность локальных баз данных, содержащих сведения об изделиях, производственной среде, ресурсах и процессах предприятия;
- б) современное направление развития информационного обеспечения производственных и бизнес-процессов, направленное на создание единого информационного пространства;
- в) процесс создания, преобразования и передачи информации;
- г) программные средства управления данными об изделиях.

**7. В соблюдении каких условий заключаются принципы разработки многопользовательских БД?**

- а) Системный подход;
- б) Информатизация;
- в) Стандартизация;
- г) Систематизация.

**8. Планирование разработки – это...**

- а) подготовительные работы, включающие в себя нахождение методов и средств оптимального решения задачи, поставленной перед информационной системой;
- б) трудоемкий процесс сбора и анализа информации от каждого потенциального пользователя БД;
- в) разработка и исследование концептуальных и логических моделей БД;
- г) разработка и исследование физических моделей БД.

**9. Физическое моделирование – это...**

- а) описание способов хранения БД на запоминающих устройствах;
- б) разработка и установление связей между таблицами БД, а также их модификация на основе принципов нормализации;
- в) определение необходимого состава таблиц БД, исходя из установленного состава пользователей;
- г) модификация на основе принципов нормализации.

**10. Логическое моделирование – это...**

- а) описание способов хранения БД на запоминающих устройствах;
- б) разработка и установление связей между таблицами БД, а также их модификация на основе принципов нормализации;
- в) определение необходимого состава таблиц БД, исходя из установленного состава пользователей;
- г) модификация на основе принципов нормализации.

**11. Этап реализации СУУБД не включает:**

- а) Формирование серверной части системы;
- б) Формирование клиентской части системы;
- в) Доработку программных модулей управления БД;
- г) Проверка разработанных информационных моделей БД.

**12. Для выполнения тестирования СУУБД...**

- а) производится наблюдение за разработанной системой;
- б) все таблицы БД должны быть заполнены соответствующей информацией;
- в) осуществляется выбор рациональных инструментальных средств разработки БД;
- г) производится экспертная оценка осуществимости проектов.

**13. Выберите верные правила записи операторов SQL:**

- а) каждая конструкция в операторе должна начинаться с новой строки, начало каждой конструкции оператора должно обозначаться одним и тем же отступом.
- б) для записи зарезервированных слов должны использоваться строчные буквы;
- в) для записи слов, определяемых пользователем, должны использоваться прописные буквы;
- г) квадратные скобки определяют обязательный элемент, например [a].

**14. Выберите верные правила записи операторов SQL:**

- а) фигурные скобки определяют необязательный элемент, например {a};
- б) многоточие (...) используется для указания многократного (необязательного) повторения выполнения группы операторов. {a | b} [, с ...] означает, что после a или b могут несколько раз повторяться символы с, разделенные запятыми;
- в) для записи слов, определяемых пользователем, должны использоваться прописные буквы;
- г) квадратные скобки определяют обязательный элемент, например [a].

**16. В языке SQL используются следующие целочисленные типы данных:**

- а) TINYINT, INT, BIGINT;
- б) MEDIUMINT, DECIMAL, INT;
- в) NUMERIC, INT, BIGINT;
- г) BLOB, NUMERIC, DECIMAL.

**17. В языке SQL используются следующие строковые типы данных:**

- а) TINYBLOB, LONGBLOB, MEDIUMBLOB;
- б) MEDIUMINT, TINYTEXT, LONGTEXT;
- в) MEDIUMTEXT, LONGTEXT, VARCHAR;
- г) BLOB, NUMERIC, LONGTEXT.

**18. Какое из утверждений о мониторе обработки транзакций неверно:**

- а) это программа, управляющая обменом данных между клиентами и серверами;
- б) обеспечивает создание единообразной вычислительной среды, отвечающей современным концепциям управления производством посредством создания единого информационного пространства;
- в) повышает эффективность управления базами данных за счет маршрутизации транзакций;
- г) обычно применяется в среде с большим объемом транзакций, в которой он не имеет возможности снять часть нагрузки с сервера СУБД.

**19. Архитектура ODBC состоит из:**

- а) приложения, диспетчера драйверов, агентов баз данных;
- б) драйверов баз данных, источников данных и приложения;
- в) 4 компонентов;
- г) приложения и источников данных.

**20. Основу структуры ADO.NET составляют два основных модуля:**

- а) приложение и резидентная реляционная база данных;
- б) провайдер данных и резидентная реляционная база данных;
- в) провайдер данных и приложение;
- г) приложение и источники данных.

## **Задания для самостоятельной работы студентов**

### **1.1 Перечень вопросов по дисциплине для самостоятельного изучения.**

- 24. Технологические методы защиты информации.
- 25. Программы для создания зашифрованной области на жестком диске DriveCrypt Plus Pack3.
- 26. Установка привилегий доступа к данным.
- 27. Назначение сервера, рабочей станции.
- 28. Классификация SQL-серверов и особенности их применения.
- 29. Преимущества использования компонентов построения запросов по сравнению с компонентами прямого доступа к таблице.
- 30. Назначение и правила использования элементов базы данных.
- 31. Уровни представления данных: уровень сущностей и уровень атрибутов.
- 32. Приемы графического проектирования структуры базы данных.
- 33. Технология автоматического создания на сервере файла базы данных по спроектированной модели.
- 34. Состав параметров при установлении привилегий.
- 35. Значение резервного копирования для сохранности данных.
- 36. Оптимизация базы данных путем создания и немедленного восстановления.
- 37. Различные методы сортировки, поиска и фильтрации данных.
- 38. Понятие исключительной ситуации при работе клиентского приложения с удаленной базой данных.
- 39. Мягкий и жесткий выход из исключительной ситуации.
- 40. Способы перехвата исключительных ситуаций.
- 41. Отличие исключительных ситуаций, возникающих на сервере и в клиентском приложении.

## **ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ**

### **Экзаменационные вопросы:**

#### **Раздел 1. Основы алгоритмизации и программирования.**

- 1. Этапы разработки программы.
- 2. Понятие алгоритма. Свойства и формы записи алгоритмов.

3. Виды алгоритмов: линейные, разветвляющиеся, циклические.
4. Классификация языков программирования.
5. Основные понятия алгоритмических языков программирования.
6. Алфавит языка Паскаль (используемые символы, служебные слова, комментарии).
7. Структура программы на языке Pascal. Объекты, указываемые в блоке описания переменных.
8. Типы данных в языке ВР. Порядковые типы данных.
9. Типы данных в языке ВР. Вещественные типы данных.
10. Ввод/вывод данных в языке Паскаль. Форматный вывод данных.
11. Условный и составной операторы. Общий вид, способ выполнения, примеры.
12. Оператор выбора CASE. Общий вид, способ выполнения, примеры.
13. Счетный оператор цикла FOR. Общий вид, способ выполнения, примеры.
14. Оператор цикла типа REPEAT. Общий вид, способ выполнения, примеры.
15. Оператор цикла типа WHILE. Общий вид, способ выполнения, примеры.
16. Понятие циклов. Вложенные циклы. Процедуры управления циклами.
17. Описание массивов. Ввод и вывод массивов. Обращение к элементу массива.
18. Методы сортировки одномерных массивов.
19. Двумерные массивы. Типовые алгоритмы обработки матриц.
20. Множества. Описание множеств, операции и процедуры, применяемые к множествам, примеры.
21. Записи. Описание, понятие полей записи, примеры.
22. Строки. Описание строк, процедуры и функции для работы со строками, примеры.
23. Подпрограммы. Использование подпрограмм, общая структура процедур и функций, примеры.
24. Параметры подпрограмм. Фактические и формальные параметры, параметры-переменные и параметры-значения, примеры.
25. Рекурсивные процедуры и функции. Примеры.

26. Понятие файлов. Доступ к файлам. Процедуры и функции для работы с файлами.
27. Типизированные файлы. Процедуры и функции для работы с типизированными файлами, примеры.
28. Нетипизированные файлы. Процедуры и функции для работы с нетипизированными файлами, примеры.
29. Текстовые файлы. Процедуры и функции для работы с текстовыми файлами, примеры.

## **Раздел 2. Основы web-программирования.**

1. Предмет Интернет-программирования, его особенности. Различия в программировании на стороне клиента и сервера. Инструменты и технологии программирования.
2. Основные понятия Интернет: World Wide Web, Веб-серверы, Веб-страницы, Протоколы, Браузеры, Адресация в сети Интернет.
3. Глобальные компьютерные сети: основные понятия, принципы функционирования. Каталоги ресурсов. Поисковые системы.
4. Понятие технологии клиент- сервер, Web-сервер, сервер БД, почтовый сервер, файловый сервер.
5. Хостинг. Бесплатный хостинг. FTP. Размещение Интернет-ресурса на сервере провайдера. Регистрация Интернет-ресурса в каталогах и поисковых системах.
6. Классификация веб-сайтов.
7. Протокол HTTP. Структура запросов и ответов.
8. Протокол SMTP (соединение, посылка письма).
9. Протокол IMAP4 (соединение, выбор ящика, проверка флагов)
10. Протокол POP3 (соединение, получение списка писем, статистики, получение письма).
11. Протокол IMAP4 (чтение удаление, перемещение писем).
12. Протокол NNTP (соединение, получение списка групп, списка новостей, посылка новости).
13. Протокол FTP (соединение, подготовка второго соединения)
14. Протокол FTP (просмотр содержимого папки, получение файлов).

15. Протокол HTTP (соединение, метод GET).
16. Протокол HTTP (метод POST, HTTP 1.1).
17. Протокол ICMP (ping).
18. Общие понятия языка разметки гипертекста – HTML.
19. Структура HTML-документа.
20. Ввод и оформление текста в HTML.
21. Создание списков в HTML.
22. Создание ссылок в HTML.
23. Встраивание изображений на веб-страницу средствами HTML.
24. Добавление мультимедийных файлов на веб-страницу средствами HTML.
25. Вставка фоновых изображений и фоновой музыки на веб-страницу средствами HTML.
26. Создание таблиц средствами языка HTML.
27. Создание фреймов в HTML.
28. Конструирование форм в языке HTML.
29. Технология создания карты графических ссылок.
30. Основные правила Web-дизайна.
31. Понимание роли графики и анимации в Web-дизайне.
32. Общие подходы к дизайну сайта. Разработка макета страницы.
33. Введение в таблицы стилей и язык CSS.
34. Использование стиля при оформлении сайта. Возможности CSS.
35. Применение таблиц стилей CSS в форматировании текста.
36. Структурное форматирование в CSS.
37. Свойства для форматирования таблиц в CSS.
38. Пользовательские формы в CSS.
39. Позиционирование объектов в CSS.
40. Основы синтаксиса Java Script и объектно – ориентированное программирование.
41. Внедрение Java Script в код HTML страницы.

42. Выражения и операторы в Java Script.
43. Создание объектов и функций в Java Script.
44. Создание динамических веб-страниц средствами Java Script.
45. Особенности языка PHP.
46. Переменные. Константы. Операторы в PHP. Циклы. Массивы. Работа со строками.
47. PHP. Операторы INCLUDE и REQUIRE. Особенности написания функций.
48. PHP. Работа с классами.
49. PHP. Регулярные выражения.
50. PHP. Работа с текстовыми файлами.
51. PHP. Функции управления сессиями.
52. PHP. Обработка входных данных.
53. PHP. Доступ к базам данных.
54. Функции в PHP. Встроенные функции.
55. Работа с датой и временем в PHP.
56. Подключение к базе данных из PHP файла. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу. Передача параметров в запрос.
57. Методы передачи параметров между страницами (GET, POST). Обработка действий пользователя при помощи форм.
58. Аутентификация пользователей средствами PHP.
59. Аутентификация пользователей средствами Web-сервера.
60. PHP. Способы управления сессиями. Работа с теньскими посылками.
61. Создание HTML-страниц средствами PHP.
62. Связь PHP и HTML.
63. Работа с базой данных MySQL.
64. Архитектура базы данных MySQL (таблицы, связи, триггеры).
65. Принципы хранения информации в базах данных MySQL.
66. Активные серверные страницы (ASP). Особенности синтаксиса.

67. Объекты ASP: Application, Session, Server, Request и Response.

### **Раздел 3. Базы данных.**

1. Современное состояние технологий баз данных.
2. Основные свойства баз данных.
3. Классификация баз данных.
4. Возможности СУБД.
5. Основные достоинства и недостатки СУБД.
6. Типовая структура современной СУБД.
7. Трёхуровневая архитектура данных.
8. Жизненный цикл базы данных.
9. Этапы проектирования. Концептуальное проектирование.
10. ER-модель, ERR-модель. Специализация, генерализация, категоризация.
11. Сетевая и иерархическая модели данных.
12. Правила реляционной базы данных.
13. Проектирование реляционной базы данных. Понятие нормализации.
14. Нормальные формы. Правила нормализации.
15. Организация доступа к данным. Индексные файлы.
16. Базы данных на инвертированных файлах.
17. Языки определения данных.
18. Языки манипулирования данными.
19. Структура языка SQL.
20. Язык запросов SQL, команда Select.
21. Операторы манипулирования данными и операторы определения данных в языке SQL.
22. Модель и свойства транзакций.
23. Триггеры и хранимые процедуры.
24. Функции администратора базы данных.

25. Методы обеспечения безопасности базы данных.
26. Методика резервного копирования и восстановления базы данных.
27. Объектно-ориентированные и объектно-реляционные базы данных.
28. Оперативные и информационные системы. Хранилища данных.
29. Банки данных. Основные функции.
30. Основные преимущества и недостатки использования банков данных.

#### **Раздел 4. Разработка и эксплуатация удаленных баз данных.**

1. Удаленные и распределенные базы данных.
2. Основные свойства распределенных баз данных.
  1. Архитектура клиент-сервер в технологии управления удаленными базами данных.
  2. Двухуровневые модели управления базами данных.
  3. Трехуровневые модели управления базами данных.
  4. CALS-технологии – основная концепция разработки удаленных баз данных.
  5. Принципы разработки многопользовательских информационных систем.
  6. Организация многопользовательских систем управления базами данных в локальных вычислительных сетях.
  7. Этапы проектирования многопользовательских баз данных.
  8. Администрирование баз данных.
  9. Назначение языка SQL.
  10. Основные правила записи операторов в языке SQL.
  11. Операторы манипулирования данными SQL.
  12. Операторы определения данных SQL.
  13. Структура организации доступа к данным в трехуровневой архитектуре.
  14. Объектные модели доступа к удаленным базам данных.
  15. Монитор обработки транзакций.
  16. Универсальная стратегия доступа к данным ODBC.
  17. Технологии доступа к данным COM, ADO.NET, .NET, .NET Framework, CORBA.
  18. Основные проблемы и способы защиты баз данных.
  19. Дисковое хранилище с системой уничтожения данных.
  20. Организационные рекомендации по обеспечению безопасности эксплуатации удаленных баз данных.
  21. Восстановление базы данных.
  22. Объектно-ориентированные модели данных.

23. Языки программирования объектно-ориентированных баз данных.
24. Управление буферами базы данных.
25. Механизм резервного копирования.
26. Структура базы данных.
27. Активные базы данных.
28. Дедуктивные базы данных.
29. Интегрированная информационная среда предприятия.
30. Структура и состав интегрированной информационной среды предприятия.
31. Управление интегрированной информационной средой предприятия.
32. Управление качеством.
33. Управление потоками работ.

**КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ  
ПРИ ЗАЧЕТЕ / ЭКЗАМЕНЕ**

Баллы	Оценка /зачет	критерии оценивания
85 – 100	«отлично» / зачтено	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, владеет разносторонними навыками и приемами выполнения практических задач по формированию общепрофессиональных компетенций.
75 - 84	«хорошо» / зачтено	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, а также имеет достаточно полное представление о значимости знаний по дисциплине.
51 – 74	«удовлетворительно» / зачтено	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает сложности при выполнении практических работ и затрудняется связать

		теорию вопроса с практикой.
менее 51	«неудовлетворительно»/незачтено	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по данной дисциплине.

## ТЕМАТИКА КУРСОВЫХ РАБОТ / ПРОЕКТОВ

Не предусмотрено

### КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ КУРСОВЫХ РАБОТ / ПРОЕКТОВ

Баллы	Оценка /зачет	критерии оценивания
85 – 100	«отлично»	Во введении приводится обоснование выбора конкретной темы, полностью раскрыта актуальность её в научной отрасли, чётко определены грамотно поставлены задачи и цель курсовой работы. Основная часть работы демонстрирует большое количество прочитанных автором работ. В ней содержатся основные термины и они адекватно использованы. Критически оценены источники: вся необходимая информация проанализирована, вычленена, логически структурирована. Присутствуют выводы и грамотные обобщения. В заключении сделаны логичные выводы, а собственное отношение выражено чётко. Автор <b>курсовой работы</b> грамотно демонстрирует осознание возможности применения исследуемых теорий, методов на практике. Приложение содержит цитаты и таблицы, иллюстрации и диаграммы. <b>Курсовая работа</b> написана в стиле академического письма (использован научный стиль изложения материала). Автор адекватно применял терминологию, правильно оформил ссылки. Оформление работы соответствует требованиям ГОСТ, библиография, приложения оформлены на отличном уровне. Объём работы соответствует требованиям.
75 – 84	«хорошо»	Во введении содержит некоторую нечёткость формулировок. В основной её части не всегда проводится критический анализ, отсутствует авторское отношение к изученному материалу. В заключении неадекватно использована терминология, наблюдаются незначительные ошибки в стиле, многие цитаты грамотно оформлены. Допущены незначительные неточности в оформлении библиографии, приложений .
51 – 74	«удовлетворительно»	во введении содержит лишь попытку обоснования выбора темы и актуальности, отсутствуют чёткие формулировки. Расплывчато определены задачи и цели. Основное содержание - пересказ чужих идей, нарушена логика изложения, автор попытался сформулировать выводы. В заключении автор попытался сделать обобщения, собственного отношения к работе практически не проявил. В

		приложении допущено несколько грубых ошибок. Не выдержан стиль требуемого академического письма по проекту в целом, часто неверно употребляются научные термины, ссылки оформлены неграмотно, наблюдается плагиат.
менее 51	«неудовлетворительно»	Во введении не содержит обоснования темы, нет актуализации темы. Не обозначены цели, задачи проекта. Скупое основное содержание указывает на недостаточное число прочитанной литературы. Внутренняя логика всего изложения работы слабая. Нет критического осмысления прочитанного, как и собственного мнения. Нет обобщений, выводов. Заключение таковым не является. В нём не приведены грамотные выводы. Приложения либо вовсе нет, либо оно недостаточно. В работе наблюдается отсутствие ссылок, плагиат, не выдержан стиль, неадекватное использование терминологии. По оформлению наблюдается ряд недочётов: не соблюдены основные требования ГОСТ, а библиография с приложениями содержит много ошибок.

#### **IV. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИЕ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ**

Процедура оценивания – порядок действий при подготовке и проведении аттестационных испытаний и формировании оценки.

Процедура промежуточной аттестации проходит в соответствии с Положением о промежуточной (рубежной) аттестации знаний студентов и учащихся ДГУНХ.

- Аттестационные испытания проводятся преподавателем (или комиссией преподавателей – в случае модульной дисциплины), ведущим лекционные занятия по данной дисциплине, или преподавателями, ведущими практические и лабораторные занятия (кроме устного экзамена). Присутствие посторонних лиц в ходе проведения аттестационных испытаний без разрешения ректора или проректора не допускается (за исключением работников университета, выполняющих контролируемые функции в соответствии со своими должностными обязанностями). В случае отсутствия ведущего преподавателя аттестационные испытания проводятся преподавателем, назначенным письменным распоряжением по кафедре .

- Инвалиды и лица с ограниченными возможностями здоровья, имеющие нарушения опорно-двигательного аппарата, допускаются на аттестационные испытания в сопровождении ассистентов-сопровождающих.

- Во время аттестационных испытаний обучающиеся могут пользоваться программой учебной дисциплины, а также с разрешения преподавателя справочной и нормативной литературой, непрограммируемыми калькуляторами.

- Время подготовки ответа при сдаче зачета/экзамена в устной форме должно составлять не менее 40 минут (по желанию обучающегося ответ может быть досрочным). Время ответа – не более 15 минут.

- При подготовке к устному экзамену экзаменуемый, как правило, ведет записи в листе устного ответа, который затем (по окончании экзамена) сдается экзаменатору.

- При проведении устного экзамена экзаменационный билет выбирает сам экзаменуемый в случайном порядке.

- Экзаменатору предоставляется право задавать обучающимся дополнительные вопросы в рамках программы дисциплины текущего семестра, а также, помимо теоретических вопросов, давать задачи, которые изучались на практических занятиях.

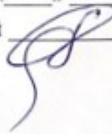
- Оценка результатов устного аттестационного испытания объявляется обучающимся в день его проведения. При проведении письменных аттестационных испытаний или компьютерного тестирования – в день их проведения или не позднее следующего рабочего дня после их проведения.
- Результаты выполнения аттестационных испытаний, проводимых в письменной форме, форме итоговой контрольной работы или компьютерного тестирования, должны быть объявлены обучающимся и выставлены в зачётные книжки не позднее следующего рабочего дня после их проведения.

Порядок подготовки и проведения промежуточной аттестации в форме зачета/экзамена

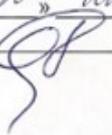
действие	сроки	методика	ответственный
выдача вопросов для промежуточной аттестации	1 неделя семестра	на лекционных /практических и др.занятиях, на офиц.сайте вуза и др.	ведущий преподаватель
консультации	последняя неделя семестра/период сессии	на групповой консультации	ведущий преподаватель
промежуточная аттестация	в период сессии	устно, письменно, тестирование бланочное или компьютерное, по билетам, с практическими заданиями	ведущий преподаватель, комиссия
формирование оценки	на аттестации		ведущий преподаватель, комиссия

Лист актуализации фонда оценочных средств по междисциплинарному курсу  
«Разработка, внедрение и адаптация программного обеспечения отраслевой направленности»

Фонд оценочных средств пересмотрен,  
обсужден и одобрен на заседании кафедры

Протокол от «25» мая 2019 г. № 9  
Зав. кафедрой 

Фонд оценочных средств пересмотрен,  
обсужден и одобрен на заседании кафедры

Протокол от «30» июня 2020 г. № 10  
Зав. кафедрой 

Фонд оценочных средств пересмотрен,  
обсужден и одобрен на заседании кафедры

Протокол от «26» мая 2021 г. № 9  
Зав. кафедрой 