УПРАВЛЕНИЕ ОБРАЗОВАНИЯ И НАУКИ ЛИПЕЦКОЙ ОБЛАСТИ
Государственное областное бюджетное профессиональное
образовательное учреждение
«Усманский многопрофильный колледж»

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

по учебной дисциплине МДК. 02.01 Информационные технологии и платформы разработки информационных систем

Программы подготовки специалистов среднего звена (ППССЗ)

по специальности: 09.02.04 Информационные системы (по отраслям)

по программе базовой подготовки

Усмань 2020

Методические указания по организации и проведению практических работ по учебной дисциплине МДК. 02.01 Информационные технологии и платформы разработки информационных систем по специальности 09.02.04 Информационные системы (по отраслям)

Организация-разработчик: Государственное областное бюджетное профессиональное образовательное учреждение «Усманский многопрофильный колледж»

Разработчики:
Мотин И.А. преподаватель информатики

Рассмотрены и утверждены на заседании предметно-цикловой комиссии естественнонаучных дисциплин

Протокол № 6 от 30.06.2020 г.

Председатель предметно-цикловой комиссии естественнонаучных дисциплин _____ Коровина Т.В.

УТВЕРЖДАЮ

Заместитель директора _____ Лаува О.А.

по учебно-методической работе

Введение

Практические занятия, как вид учебных занятий, направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений и составляют важную часть теоретической и профессиональной практической подготовки.

В процессе практического занятия обучающиеся выполняют одно или несколько практических заданий в соответствии с изучаемым содержанием учебного материала.

Содержание практических занятий по учебной дисциплине МДК. 02.01 Информационные технологии и платформы разработки информационных систем должно охватывать весь круг профессиональных умений, на подготовку к которым ориентирована данная дисциплина, а в совокупности охватывать всю профессиональную деятельность, к которой готовится специалист.

При разработке содержания практических занятий следует учитывать, что наряду с формированием умений и навыков в процессе практических занятий обобщаются, систематизируются, углубляются и конкретизируются теоретические знания, вырабатывается способность и готовность использовать теоретические знания на практике, развиваются интеллектуальные умения.

Выполнение обучающимися практических занятий проводится с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки обучающихся, установленными ФГОС и рабочей программой учебной дисциплины МДК. 02.01 Информационные технологии и платформы разработки информационных систем я по конкретным разделам и темам дисциплины;
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;

– совершенствования умений применять полученные знания на практике, реализации единства интеллектуальной и практической деятельности;
– развития интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;
– выработки таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива при решении поставленных задач при освоении общих и профессиональных компетенций.

Соответственно в процессе освоения учебной дисциплины МДК. 02.01 Информационные технологии и платформы разработки информационных систем обучающиеся должны овладеть:

умениями:

- осуществлять математическую и информационную постановку задач по обработке информации, использовать алгоритмы обработки информации для различных приложений;
- уметь решать прикладные вопросы интеллектуальных систем с использованием статических экспертных систем, экспертных систем реального времени;
- использовать языки структурного, объектно-ориентированного программирования и языка сценариев для создания независимых программ, разрабатывать графический интерфейс приложения;
- создавать проект по разработке приложения и формулировать его задачи, выполнять управление проектом с использованием инструментальных средств;

знаниями:

- основные виды и процедуры обработки информации, модели и методы решения задач обработки информации (генерация отчетов, поддержка принятия решений, анализ данных, искусственный интеллект, обработка изображений);
- сервисно ориентированные архитектуры, CRM-системы, ERP-системы;
- объектно-ориентированное программирование;

- спецификации языка, создание графического пользовательского интерфейса (GUI), файловый ввод-вывод, создание сетевого сервера и сетевого клиента;
- платформы для создания, исполнения и управления информационной системой;
- основные процессы управления проектом разработки.

Вышеперечисленные умения и знания направлены на формирование следующих профессиональных и общих компетенций студентов:

Профессиональные компетенции:

ПК 2.1	Участвовать в разработке технического задания.
ПК 2.2	Программировать в соответствии с требованиями технического задания.
ПК 2.3	Применять методики тестирования разрабатываемых приложений.
ПК 2.4	Формировать отчетную документацию по результатам работ.
ПК 2.5	Оформлять программную документацию в соответствии с принятыми стандартами.
ПК 2.6	Использовать критерии оценки качества и надежности функционирования информационной системы.

Общие компетенции:

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 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.

Данные методические указания по организации и проведению практических работ составлены в соответствии с содержанием рабочей программы учебной дисциплины МДК. 02.01 Информационные технологии и платформы разработки информационных систем специальности 09.02.04 Информационные системы (по отраслям) по программе базовой подготовки.

Учебная дисциплина МДК. 02.01 Информационные технологии и платформы разработки информационных систем изучается в течение трех семестров. Общий объем времени, отведенный на выполнение практической работы по учебной дисциплине МДК. 02.01 Информационные технологии и платформы разработки информационных систем, составляет в соответствии с учебным планом и рабочей программой– **63** часа.

Методические указания призваны помочь студентам правильно организовать работу и рационально использовать свое время при овладении содержанием учебной дисциплины МДК. 02.01 Информационные технологии и платформы разработки информационных систем, закреплении теоретических знаний и умений.

Распределение часов на выполнение практической работы студентов по разделам и темам учебной дисциплины МДК. 02.01 Информационные технологии и платформы разработки информационных систем

Наименование раздела, темы	Количество часов на ВСР
Раздел 1 Проектирование ИС	**60**
Тема 1.1. Архитектура информационных систем	8
Тема 1.2. Технологии разработки АИС	16
Тема 1.3. Аппаратно-программные платформы разработки ИС	4
Тема 1.5. Проектирование серверной части АИС	8
Тема 1.6. Проектирование клиентской части АИС	20
Тема 1.7. Тестирование приложений АИС	4
Раздел 2. Сбор и анализ информации для определения потребностей клиента при проектировании ИС	**3**
Тема 2.1. Технология сбора информации	1
Тема 2.2. Типовой состав документов на программный продукт	2

Перечень рекомендуемой литературы

Варфоломеева А.О., Коряковский А.В., Романов В.П. Информационные системы предприятия: учебное пособие / ИНФРА. – М. , 2019.

Федорова Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / КУРС, 2020.

Мелихова, Е. В. Обеспечение проектной деятельности: анализ и реализация. Ч. 2: Учебное пособие / Мелихова Е.В. - Волгоград:Волгоградский государственный аграрный университет, 2018. - 160 с

Ротер, М. Учитесь видеть бизнес-процессы. Практика построения карт потоков создания ценности / Ротер М., Шук Д., Пер.Муравьевой Г., - 5-е изд. - Москва :Альпина Пабл., 2017. - 136 с. ISBN 978-5-9614-6145-9.

Дополнительные источники:

Шишов, О. В. Технические средства автоматизации и управления : учеб. пособие / О.В. Шишов. — Москва : ИНФРА-М, 2019. — 396 с

Организация сетевого администрирования: Учебник / А.И. Баранчиков, П.А. Баранчиков, А.Ю. Громов, О.А. Ломтева. — Москва: КУРС: ИНФРА-М, 2020. — 384 с. - ISBN 978-5-16-104348-6.

Раздел 1. Проектирование ИС
Тема 1.1. Архитектура информационных систем (8 часов)

Практическая работа №1
«Проведение анализа информационного обеспечения ИС»

Задачи обучающегося:

1. Описать и проанализировать ИС.
2. Определить необходимые элементы КТС ИС и системного ПО ИС.

Опорные понятия: информационное обеспечение ИС.
Планируемый результат:
Студент должен
Уметь правильно осуществлять информационную постановку задач по обработке информации.
Необходимое оборудование: учебная литература, интернет

Алгоритм деятельности обучающегося:
Изучите теоретическую часть.

Проблемы управления программными проектами впервые появились в 60-х–начале70-хгодов прошлого века, когда провалились многие большие проекты по разработке программных продуктов. Были зафиксированы задержки в создании ПО, программное обеспечение было ненадежным, затраты на разработку в несколько раз превосходили первоначальные оценки и т.д. Провалы этих проектов обуславливались не только некомпетентностью руководителей и программистов. Напротив, в этих больших поисковых проектах принимали участие люди, уровень квалификации которых был явно выше среднего. Причины провалов коренились в тех подходах, которые использовались в управлении проектами. Применяемая методика была основана на опыте управления техническими проектами и оказалась неэффективной при разработке программных проектов.

Руководители программных проектов выполняют такую же работу, что и руководители технических проектов. Вместе с тем процесс разработки ПО существенно отличается от процессов реализации технических проектов. Ниже приведен небольшой список этих отличий.

1. ПРОГРАММНЫЙ ПРОДУКТ НЕМАТЕРИАЛЕН. Менеджер судостроительного проекта или проекта постройки здания видит результат выполнения своего проекта. Если реализация проекта отстает от графика,то это видно по незавершенности конструкции. В противоположность этому процент незавершенности программного проекта нельзя увидеть или потрогать. Менеджер программного проекта может полагаться только на документацию, которая фиксирует процесс разработки программного продукта.

2. НЕ СУЩЕСТВУЕТ СТАНДАРТНЫХ ПРОЦЕССОВ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. На сегодняшний день не существует четкой зависимости между процессом создания ПО и типом создаваемого программного продукта. Другие технические дисциплины имеют длительную историю, процессы разработки технических изделий многократно опробованы и проверены. Изучением же процессов создания ПО специалисты занимаются только последние несколько лет. Поэтому прока нельзя точно предсказать, на каком этапе процесса разработки ПО могут возникнуть проблемы, угрожающие всему проекту.

3. БОЛЬШИЕ ПРОГРАММНЫЕ ПРОЕКТЫ – ЭТО ЧАСТО ОДНОРАЗОВЫЕ ПРОЕКТЫ. Большие программные проекты, как правило, значительно отличаются от проектов, реализованных ранее. Поэтому, чтобы уменьшить неопределенность в планировании проекта, руководители проектов должны обладать очень большим практическим опытом. Но постоянные технологические изменения в компьютерной технике обесценивают предыдущий опыт. Перечисленные особенности могут привести к тому, что реализация проекта выйдет за рамки временного графика или бюджетных ассигнований. Об этом всегда нужно помнить.

Процессы управления программными проектами

Невозможно описать и стандартизировать все работы, выполняемые менеджером проекта по созданию ПО, но в большинстве случаев к ним относятся.

*Написание предложений по созданию ПО.

*Планирование и составление графика работ проекта.

*Оценивание стоимости проекта.

*Контроль процессов выполнения работ.

*Подбор персонала.

*Написание отчетов и представлений.

Время выполнения больших программных проектов может занимать несколько лет. В течение этого времени цели и намерения организации,

аказавшей программный проект, могут существенно измениться. Может оказаться, что разрабатываемый программный продукт стал уже ненужным либо исходные требования к ПО устарели и их нужно кардинально менять. В такой ситуации руководство организации-разработчика может принять решение о прекращении разработки ПО или об изменении проекта в целом.

Планирование проекта

Эффективное управление проектами напрямую зависит от правильного планирования работ. План, разработанный на начальном этапе проекта, рассматривается всеми его участниками как руководящий документ, выполнение которого должно привести к успешному завершению проекта. Этот первоначальный план должен максимально подробно описывать все этапы реализации проекта.

План проекта должен показать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. Детализация планов проектов очень разнится в зависимости от типа разрабатываемого программного продукта и организации-

разработчика. Но в любом случае большинство планов содержит следующие разделы.

1. Введение. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.).

2. Организация выполнения проекта. Описание способа подбора команды разработчиков и распределение обязанностей между членами команды.

3. Анализ рисков. Описание возможных проектных рисков, вероятность их проявления и стратегий, направленных на их уменьшение.

4. Аппаратные и программные ресурсы для реализации проекта. Перечень аппаратных средств и программного обеспечения, необходимого для разработки программного продукта.

5. Разбиение работ на этапы. Проект разбивается на отдельные процессы, определяются этапы выполнения проекта, приводится описание результатов каждого этапа и контрольные отметки.
6. График работ. В графике работ отображаются зависимости между отдельными этапами разработки по, оценки времени их выполнения и распределение членов команды проекта по отдельным этапам.
7. Механизмы контроля и мониторинга за ходом выполнения проекта. Описываются механизмы и сроки предоставления отчетов о ходе работ, а также механизмы мониторинга всего проекта.

При планировании проекта разработки ПО определяются контрольные точки – ***вехи***, отмечающие окончание определенного этапа работ. Для каждой вехи создается отчет, который предоставляется руководству проекта.

При определении контрольных точек весь процесс создания ПО должен быть разбит на отдельные этапы с указанным «выходом» (результатом) каждого этапа. Например, на рис. 1 показаны этапы разработки спецификации требований в случае, когда для ее проверки используется прототип системы.

Рис. 1 – Этапы процесса разработки спецификации

Информационный процесс - это осуществление всей совокупности следующих элементарных информационных актов: прием или создание информации, ее хранение, передача и использование. Информационная система - это совокупность механизмов, обеспечивающих полное осуществление информационного процесса.

Вне ИС информация может лишь сохраняться в виде записей на тех или иных физических носителях, но не может быть ни принятой, ни переданной, ни использованной.

Информационная система - организационно-техническая система, которая предназначена для выполнения информационно-вычислительных работ или предоставления информационно-вычислительных работ или предоставления информационно-вычислительных услуг, удовлетворяющих потребности системы управления и ее пользователей - управленческого персонала, внешних пользователей путем использования и/или создания информационных продуктов. Информационные системы существуют в рамках системы управления и полностью подчинены целям функционирования этих систем.

Информационно-вычислительная работа - деятельность, связанная с использованием информационных продуктов. Типичным примером информационной работы является поддержка информационных технологий управления.

Информационно-вычислительная услуга - это разовая информационно-вычислительная работа.Под информационным продуктом понимается вещественный или

нематериальный результат интеллектуального человеческого труда, обычно материализованный на определенном носителе, например разнообразных программных продуктов, выходной информации в виде документов управления, баз данных, хранилищ данных, баз знаний, проектов ИС и ИТ.

Методологическую основу изучения ИС составляет системный подход, в соответствии с которым любая система представляет собой совокупность взаимосвязанных объектов, функционирующих совместно для достижения общей цели.

Информационная система представляет собой совокупность функциональной структуры, информационного, математического, технического, организационного и кадрового обеспечения, которые объединены в единую системы в целях сбора, хранения, обработки и выдачи необходимой информации для выполнения функций управления. Она обеспечивает информационные потоки:

i-1 - информационный поток из внешней среды в систему управления, который, с одной стороны, представляет собой поток нормативной информации, создаваемый государственными учреждениями в части законодательства, а с другой стороны - поток информации о конъюнктуре рынка, создаваемый конкурентами, потребителями, поставщиками;

i-2 - информационный поток из системы управления во внешнюю среду (отчетная информация, прежде всего финансовая в государственные органы, инвесторам, кредиторам, потребителям; маркетинговая информация потенциальным потребителям);

i-3 - информационный поток из системы управления на объект, представляет собой совокупность плановой, нормативной и распорядительной информации для осуществления хозяйственных процессов;

i-4 - информационный поток от объекта в систему управления, который отражает учетную информацию о состоянии объекта управления экономической системой (сырья, материалов, денежных, энергетических, трудовых ресурсов, готовой продукции и выполненных услугах) в результате выполнения хозяйственных процессов.

Задачи информационных систем

Корпоративные системы позволяют решить следующие задачи:

- гарантировать требуемое качество управления предприятием;
- повысить оперативность и эффективность взаимодействия между подразделениями;
- обеспечить управляемость качеством выпускаемой продукции;
- увеличить экономическую эффективность деятельности предприятия;
- создать систему статистического учета на предприятии;
- осуществлять прогноз развития предприятия;
- создать систему стратегического и оперативного планирования, систему прогнозирования.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Выберите предметную область, соответствующую порядковому номеру списка группы.
2. Выберите название ИС в рамках предметной области.
3. Определите цель ИС
4. Проведите анализ осуществимости ИС

4.1. Что произойдет с организацией, если система не будет введена в эксплуатацию?

4.2. Какие текущие проблемы существуют в организации и как новая система поможет их решить?

4.3. Каким образом (и будет ли) ИС способствовать целям бизнеса?

4.4. Требует ли разработка ИС технологии, которая до этого раньше не использовалась в организации?
5. Где будет размещена ИС? Кто является пользователем ИС?
6. Комплекс технических средств ИТ
6.1. Какие средства компьютерной техники необходимы для ИС?
6.2. Какие средства коммуникационной техники необходимы для ИС?
6.3. Какие средства организационной техники необходимы для ИС?
6.4. Какие средства оперативной полиграфии необходимы для ИС?
7. Опишите системное ПО ИТ.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Расскажите про процессы управления программными проектами.
2. Расскажите про планирование проекта.
3. Представьте этапы процесса разработки спецификации.
4. Что такое информационный процесс?
5. Что такое информационная система?
6. Что такое информационно-вычислительная работа?
7. Что такое информационно-вычислительная услуга?
8. Что представляет собой информационная система?
9. Какие информационные потоки обеспечивает ИС?
10. Перечислите задачи информационных систем.

Раздел 1. Проектирование ИС
Тема 1.1. Архитектура информационных систем (8 часов)

Практическая работа №2
«Проведение анализа технического и программного обеспечения ИС»

Задачи обучающегося:

1. Описать и проанализировать ИС.
2. Определить необходимые элементы КТС ИС и системного ПО ИС.

Опорные понятия: информационное обеспечение ИС.
Планируемый результат:
Студент должен
Уметь правильно осуществлять информационную постановку задач по обработке информации.
Необходимое оборудование: учебная литература, интернет

Алгоритм деятельности обучающегося:

Выберите предметную область, соответствующую порядковому номеру списка группы.
- Железнодорожный вокзал.

Выберите название ИС в рамках предметной области.
- ИС для заказа билетов на поезд.

Определите цель ИС
- Целью данной ИС является обеспечение заказов билетов на поезд в сети Интернет.

Проведите анализ осуществимости ИС

Что произойдет с организацией, если система не будет введена в эксплуатацию?

Не будет обеспечена максимальная скорость передачи информации клиенту, а так же востребованность железнодорожного транспорта будет не высока, что ведет к упадку прибыли.

Какие текущие проблемы существуют в организации и как новая система поможет их решить?

К текущим проблемам организации относятся: не эффективное распределение и систематизирование больших потоков информации, не удобный способ заказов билетов на поезда. Система поможет эффективно распределять информацию в соответствие с запросами клиента, обеспечивать удобный и быстрый способ заказов билетов на поезда в сети Интернет.

Каким образом (и будет ли) ИС способствовать целям бизнеса?

Можно продать данную ИС Железнодорожным вокзалам, для увеличения прибыли в данной сфере

Требует ли разработка ИС технологии, которая до этого раньше не использовалась в организации?

Да, требует, для точного определения данных о станциях, маршрутах, времени отправления поездов, стоимости билетов на поезд.

Где будет размещена ИС? Кто является пользователем ИС?

Данная ИС будет размещена на сервере организации, так и в сети Интернет, для обеспечения доступа к ИС клиенту. Пользователями данной ИС являются: работники организации, пассажира.

Комплекс технических средств ИТ

Какие средства компьютерной техники необходимы для ИС?

Монитор, системный блок, клавиатура, мышь.

Какие средства коммуникационной техники необходимы для ИС?

Локальная, глобальная сети.

Какие средства организационной техники необходимы для ИС?

Средства составления и изготовления документов, средства обработки документов.

Какие средства оперативной полиграфии необходимы для ИС?

Принтер.

Опишите системное ПО ИТ.

Операционные системы Windows 7/8/8.1/10, Linux, браузер Интернета, офисные пакеты приложений Microsoft Office, Apache Openffice, антивирус Kaspersky Anti-Virus.

Ответы на контрольные вопросы:

Что такое информационный процесс?

Что такое информационная система?

Что такое информационно-вычислительная работа?

Что такое информационно-вычислительная услуга?

Что представляет собой информационная система?

Раздел 1. Проектирование ИС
Тема 1.2. Технологии разработки АИС (16 часов)

Практическая работа №3
«Моделирование бизнес-процессов средствами BPwin»

Задачи обучающегося:

1. Ознакомиться с возможностями инструментального средства системного анализа бизнес-процессов BPwin и получение с его помощью практических навыков моделирования бизнес-процессов предприятия

Опорные понятия: бизнес-процессы.
Планируемый результат:
Студент должен
Уметь правильно осуществлять информационную постановку задач по обработке информации.
Создавать контекстные диаграммы IDEF0
Необходимое оборудование: учебная литература, специализированное ПО

Алгоритм деятельности обучающегося:

На рисунке 2 представлена контекстная диаграмма «Прием отрудника на работу».

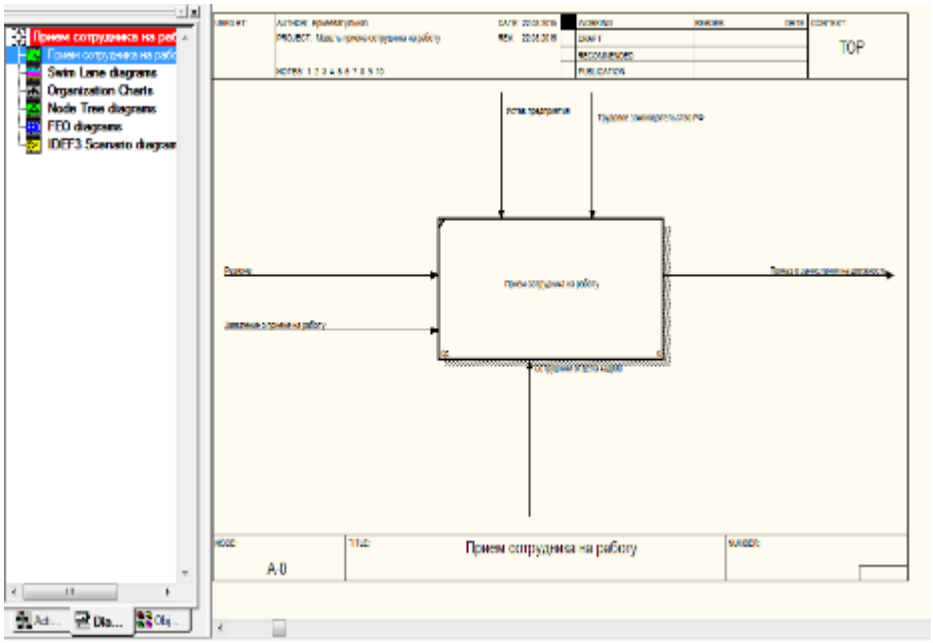


Рисунок 2 Контекстная диаграмма Прием сотрудника на работу в стандарте IDEF0

В контекстной диаграмме входной информацией являются данные: заявление о приеме на работу, резюме. Выходная информация – приказ о зачислении. Механизмами являются сотрудники отдела кадров. Управляющие стрелки – устав предприятия, трудовое законодательство РФ.

Отчет по модели представлен на рисунке 3 (Меню Tools/Reports/Model Report).

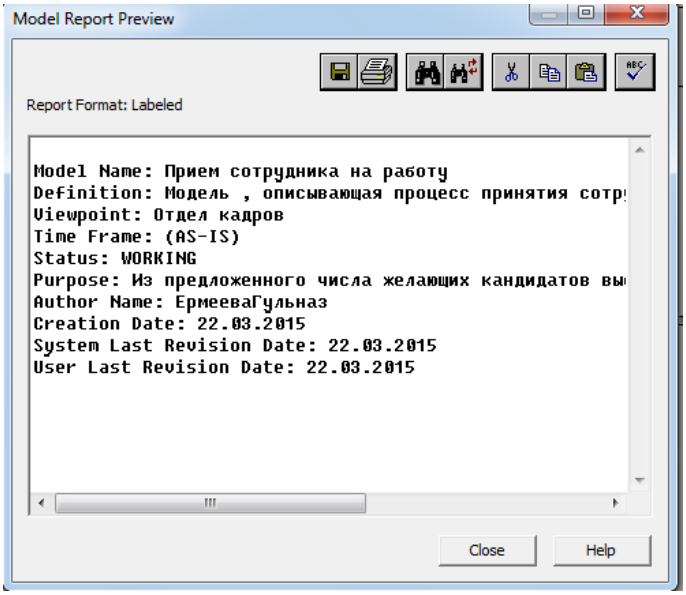


Рисунок 3 Отчет Model Report

После описания системы в целом производится разбиение ее на крупные фрагменты. На рисунке 4 представлена декомпозиция контекстной диаграммы рисунка 2.

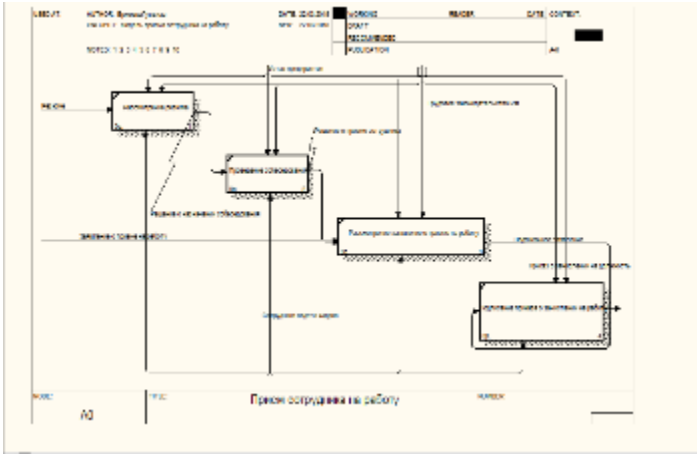


Рисунок 4 Декомпозиция контекстной диаграммы Прием сотрудника на работу

Процесс состоит из четырех работ: рассмотрение резюме, проведение собеседования, рассмотрение заявления о приеме на работу, подписание приказа о зачислении.

Далее поочередно декомпозируем контекстной диаграммы Рассмотрение резюме, Рассмотрение заявления о приеме на работу, Проведение собеседования, Подписание приказа о зачислении с целью достижения более подробного описания процесса приема сотрудника на работу.

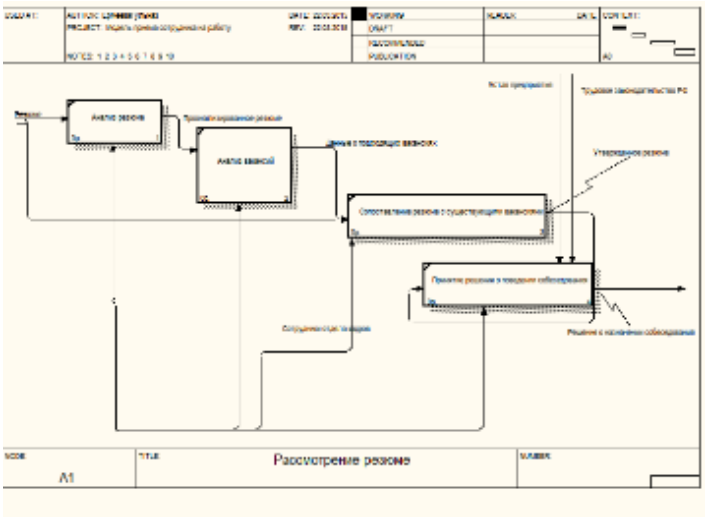


Рисунок 5 Декомпозиция контекстной диаграммы Рассмотрение резюме

В диаграмме процесса «Рассмотрение резюме» входной информацией является резюме. Процесс состоит из четырех работ: анализ резюме, анализ вакансий, сопоставление резюме с существующими вакансиями, принятие решения о проведении собеседования.

Выходная информация – решение о назначении собеседования.

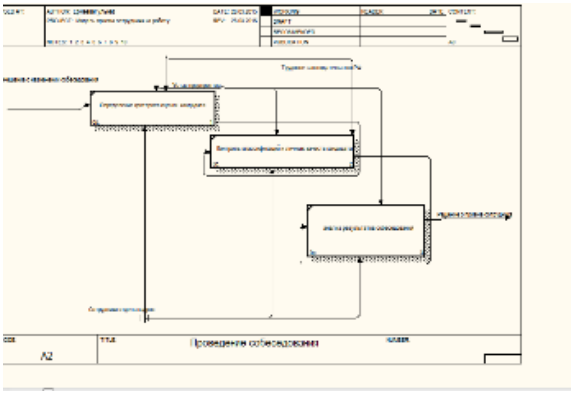


Рисунок 6 Декомпозиция контекстной диаграммы Проведение собрания

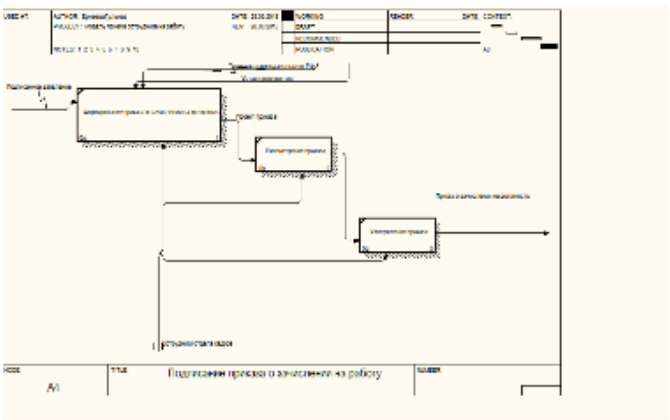


Рисунок 7 Декомпозиция контекстной диаграммы Подписание приказа о зачислении

В процессе выполнения работы было изучено моделирование бизнес-процессов с использованием инструментального средства системного анализа бизнес-процессов BPwin. В ходе работы ознакомились с возможностями пакета BPwin и получили с его помощью практические навыки моделирования бизнес-процессов предприятия.

В качестве предметной области для моделирования бизнес-процессов был выбран процесс Прием сотрудника на работу.

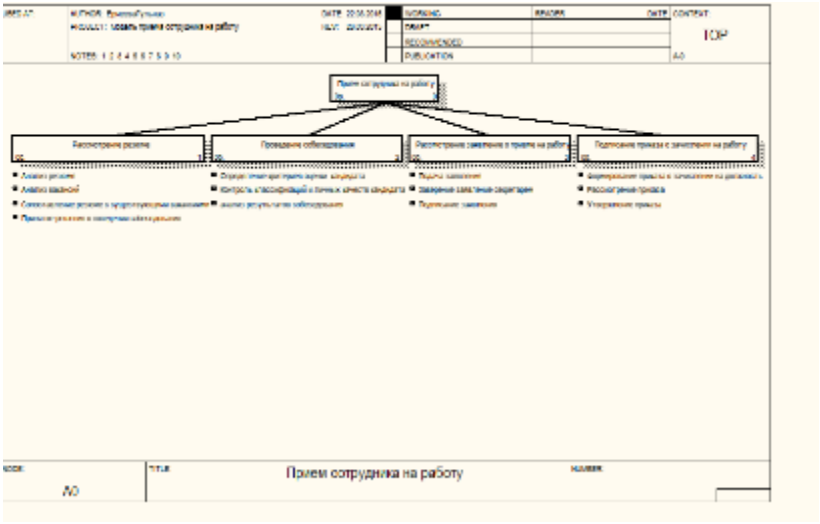


Рисунок 9 Диаграмма дерево узлов процесса «Прием сотрудника на работу»

Порядок выполнения работы:

1.Исходя из выбранной ранее темы создать диаграмму IDEF0

2.Оформить отчет о проделанной работе.

Раздел 1. Проектирование ИС
Тема 1.2. Технологии разработки АИС (16 часов)

Практическая работа №4
«Отображение модели данных в инструментальном средстве ERwin»

Задачи обучающегося:

1. Ознакомиться с возможностями инструментального средства системного анализа бизнес-процессов BPwin и получение с его помощью практических навыков моделирования бизнес-процессов предприятия

Опорные понятия: модели данных.
Планируемый результат:
Студент должен
Уметь правильно осуществлять информационную постановку задач по обработке информации.
Создавать пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, специализированное ПО

Алгоритм деятельности обучающегося:

ERwin имеет два уровня представления модели – логический и физический. Логический уровень — это абстрактный взгляд на данные, когда данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире. Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами.

Логическая модель данных разрабатывается на основе существующих моделей данных (например, реляционной), но никак не связана с конкретной реализацией системы управления базы данных (СУБД) и прочих физических условий реализации. Она может быть построена на основе другой логической модели, например, на основе модели потоков данных или процессов.

Логическая модель данных является источником информации для фазы физического проектирования. Она предоставляет разработчику физической модели данных средства проведения всестороннего анализа различных аспектов работы с данными, что имеет исключительное значение для выбора действительно эффективного проектного решения.

Физическая модель данных, напротив, зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей.

На физическом уровне объекты БД должны называться так, как того требуют ограничения СУБД. На логическом уровне можно этим объектам дать синонимы — имена более понятные неспециалистам, в том числе на кириллице и с использованием специальных символов. Такое соответствие позволяет лучше документировать модель и дает возможность обсуждать структуру данных с экспертами предметной области.

После описания логической модели проектировщик может выбрать необходимую СУБД, и ERwin автоматически создаст соответствующую физическую модель. На основе физической модели ERwin может сгенерировать системный каталог СУБД или соответствующий SQL-скрипт. Этот процесс называется прямым проектированием (Forward Engineering). Тем самым достигается масштабируемость — создав одну логическую модель данных, можно сгенерировать физические модели под любую поддерживаемую ERwin СУБД.

С другой стороны, ERwin способен по содержимому системного каталога или SQL-скрипту воссоздать физическую и логическую модель данных (Reverse Engineering). На основе полученной логической модели данных можно сгенерировать физическую модель для другой СУБД и затем создать ее системный каталог. Следовательно, ERwin позволяет решить задачу по переносу структуры данных с одного сервера на другой. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах — таблицах, колонках, индексах, процедурах и т.д.

Интерфейс ERwin выполнен в стиле Windows-приложений, достаточно прост и интуитивно понятен (рис. 2.1).

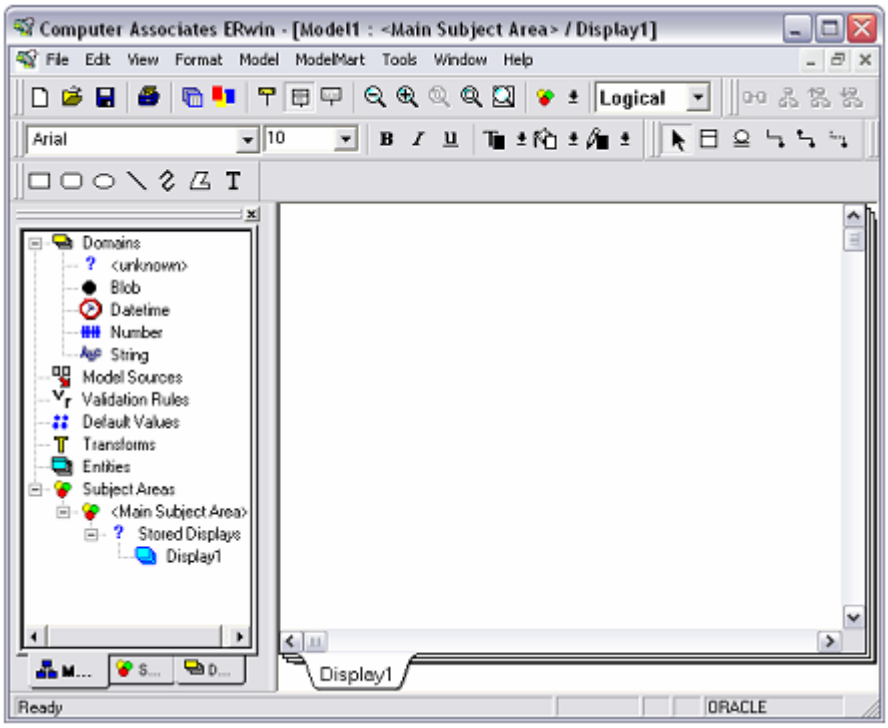


Рис. 2.1. Окно отображение логической модели

Рассмотрим кратко значения кнопок панели инструментов (табл. 2.1).

Таблица 2.1. Основная панель инструментов

Кнопки	Назначение кнопок
	Создание, открытие, сохранение и печать модели
	Изменение уровня просмотра модели: уровень сущностей, уровень атрибутов и уровень определений
	Изменение масштаба просмотра модели
	Переключение между областями модели - Subject Area
	Диалоги для генерации отчетов по модели
	Палитра инструментов
Arial	Панель инструментов Font and Color Toolbar
	Панель Суперкласс – подкласс
	Панель для рисования графических объектов

Для создания типов сущностей модели и связывания их между собой используются палитра инструментов, показанная на рис. 2.2. Палитра инструментов выглядит различно на разных уровнях отображения модели. На логическом уровне палитра инструментов имеет следующие значения кнопок (табл. 2.2).

Рис. 2.2. Палитра инструментов

Таблица 2.2. Палитра инструментов

Кнопки	Назначение кнопок	Описание
	Указатель	кнопка указателя (режим мыши) - в этом режиме можно установит фокус на каком-либо объекте модели
	Сущность	кнопка внесения сущности - для внесения сущности нужно щелкнуть левой кнопкой мыши по кнопке внесения сущности и один раз по свободному пространству на модели. Для редактирования сущностей или других объектов модели необходимо перейти в режим указателя
	Категория	категория, или категориальная связь, - специальный тип связи между сущностями, которая будет рассмотрена ниже. Для установления категориальной связи нужно щелкнуть левой кнопкой мыши по кнопке категории, затем один раз щелкнуть по сущности - родовому предку, затем - по сущности-потомку
	Идентифицирующая связь	связь между независимой и зависимой сущностями (более подробно описана ниже по тексту)
	Связь "Многие-ко-многим"	экземпляр одной сущности может быть связан со многими экземплярами другой сущности и наоборот (возможна только на уровне логической модели)
	Неидентифицирующая связь	связь между независимыми сущностями (более подробно описана ниже по тексту)

На физическом уровне палитра инструментов имеет:

- вместо кнопки категорий — кнопку внесения представлений (view);
- вместо кнопки связи "многие-ко-многим" — кнопку связей представлений.

- Запустите Computer Associates ERwin. Если появится окно ModelMart Connection Manager, закройте его нажатием на кнопку Cancel.
- Если появляется диалог Computer Associates ERwin, выберите пункт Create a new model и нажмите OK (рис. 2.3).

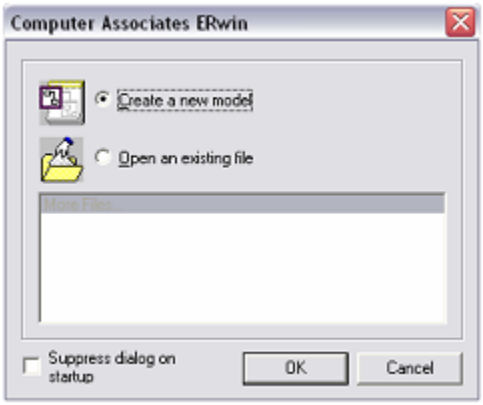


Рис. 2.3. Диалоговое окно Computer Associates ERwin

• В появившемся диалоговом окне Create Model – Select Template выберите пункт Logical/Physical и нажмите OK (рис. 2.4).

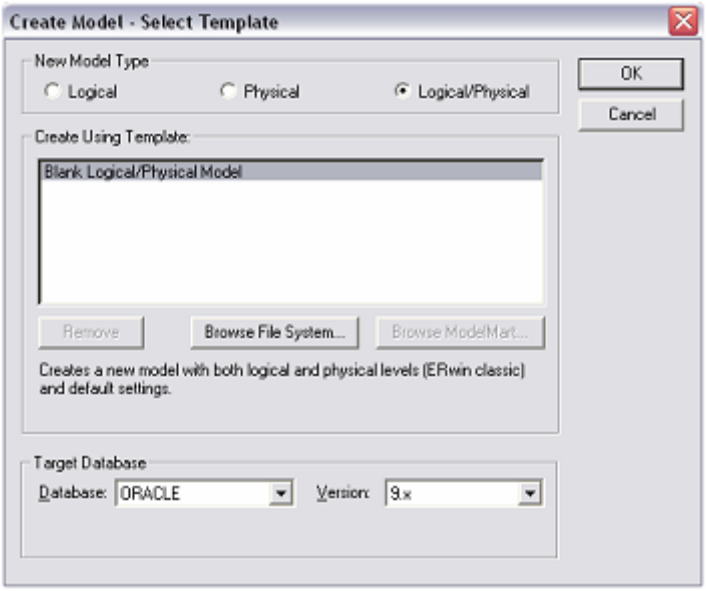


Рис. 2.4. Диалоговое окно Create Model – Select Template

• Если вам не понятно, как выполнить то или иное действие, вы можете вызвать помощь – клавиша F1 или меню Help.

Порядок выполнения работы:

1.Исходя из выбранной ранее темы создать две модели представления данных

2.Оформить отчет о проделанной работе.

Раздел 1. Проектирование ИС
Тема 1.2. Технологии разработки АИС (16 часов)

Практическая работа №5
«Создание концептуальной модели системы»

Задачи обучающегося:

1. Ознакомиться с возможностями инструментального средства системного анализа бизнес-процессов BPwin и получение с его помощью практических навыков моделирования бизнес-процессов предприятия

Опорные понятия: модели данных.
Планируемый результат:
Студент должен
Уметь правильно осуществлять информационную постановку задач по обработке информации.
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, специализированное ПО

Алгоритм деятельности обучающегося:

Различают три уровня логической модели, отличающихся по глубине представления информации о данных:

- диаграмма сущность-связь (Entity Relationship Diagram, ERD);
- модель данных, основанная на ключах (Key Based model, KB);
- полная атрибутивная модель (Fully Attributed model, FA).

Диаграмма сущность-связь представляет собой модель данных верхнего уровня. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные сущности и связи между ними, которые удовлетворяют основным требованиям, предъявляемым к ИС. Диаграмма сущность-связь может включать связи "многие-ко-многим" и не включать описание ключей. Как правило, ERD используется для презентаций и обсуждения структуры данных с экспертами предметной области.

Модель данных, основанная на ключах — более подробное представление данных. Она включает описание всех сущностей и первичных ключей и предназначена для представления структуры данных и ключей, которые соответствуют предметной области.

Полная атрибутивная модель — наиболее детальное представление структуры данных: представляет данные в третьей нормальной форме и включает все сущности, атрибуты и связи.

ERwin имеет несколько уровней отображения диаграммы: уровень сущностей, уровень атрибутов, уровень определений, уровень первичных ключей и уровень иконок. Переключиться между первыми тремя уровнями можно с использованием кнопок панели инструментов. Переключиться на другие уровни отображения можно при помощи контекстного меню, которое появляется, если нажать правую кнопку мыши на любом месте диаграммы, не занятой объектами модели. В контекстном меню следует выбрать пункт Display Level и затем необходимый уровень отображения.

2.2.2. Сущности и атрибуты

Основные компоненты диаграммы ERwin – это сущности, атрибуты и связи. Сущность можно определить как объект, событие или концепцию, информация о которой должна сохраняться. Сущности должны иметь наименование с четким смысловым значением, фактически это имя ее экземпляра. Каждый экземпляр индивидуален и должен отличаться от всех остальных экземпляров.

Атрибут выражает определенное свойство объекта. С точки зрения БД (физическая модель) сущности соответствует таблица, экземпляру сущности — строка в таблице, а атрибуту — колонка таблицы.

Entity Editor в контекстном меню для сущности позволяет определить имя, описание, комментарии, иконку. Для описания атрибутов сущности выбирается пункт Attribute Editor. Здесь можно указать имя нового атрибута и домен, который будет использоваться при определении типа колонки на уровне физической модели. Атрибуты должны именоваться в единственном числе, иметь четкое смысловое значение и быть достаточно важными для того, чтобы их моделировать. Именование сущности в единственном числе облегчает в дальнейшем чтение модели. Каждый атрибут должен быть определен (закладка Definition), при этом следует избегать циклических определений и производных атрибутов. Для внесения дополнительных комментариев и определений к сущности служат свойства, определенные пользователем (UDP).

Соблюдение этого правила позволяет частично решить проблему нормализации данных уже на этапе определения атрибутов.

Каждый атрибут хранит информацию об определенном свойстве сущности, а каждый экземпляр сущности должен быть уникальным. Атрибут или группа атрибутов, которые однозначно идентифицируют экземпляр сущность, называется первичным ключом. Атрибуты первичного ключа на диаграмме не требуют специального обозначения — это те атрибуты, которые находятся в списке атрибутов выше горизонтальной линии.

Продолжим рассмотрение практического моделирования на примере деятельности вымышленной компании, которая занимается в основном сборкой карманных персональных компьютеров (КПК) и установкой на них программного обеспечения (ПО). Компания не производит компоненты и программы самостоятельно, а только собирает КПК и устанавливает ПО.

Основные процедуры компании, следующие:

- продавцы принимают заказы клиентов;
- операторы группируют заказы по типу работ;
- операторы собирают КПК;
- операторы устанавливают ПО на КПК;
- операторы упаковывают КПК согласно заказам;
- кладовщик отгружает клиентам заказы.

Для начала представим модель базы данных компании.

Модель должна содержать 8 сущностей: Заказ, Заказчик, Оператор, КПК, ПО, Кладовщик, Готовый товар, Продавец. Каждая из представленных сущностей имеет свой набор атрибутов.

Щелкните по кнопке и на рабочем поле программы. Появится окошко сущности с названием E/1. Окошко сущности разделено на две части. В верхнюю часть входят атрибуты, которые являются ключами сущности, а в нижнюю – не ключевые атрибуты (см. рис. 2.5).

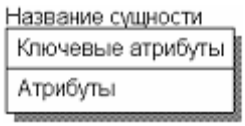


Рис. 2.5. Сущность

Двойной щелчок по сущности открывает диалоговое окно Attributes (рис. 2.6).

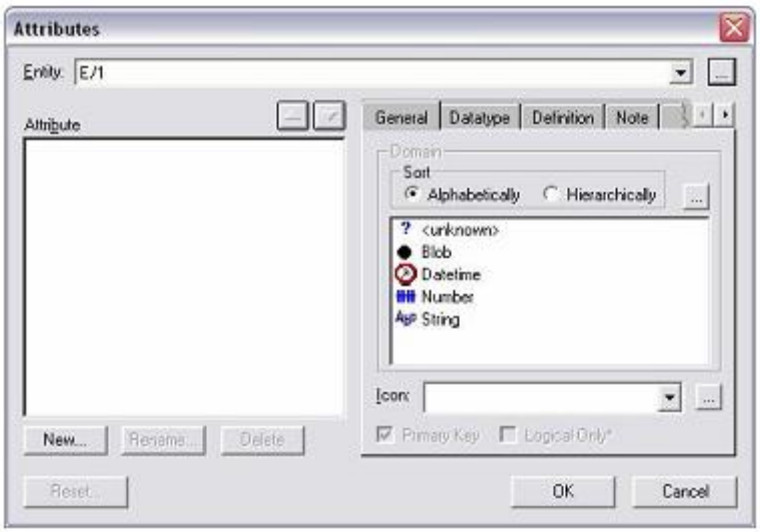


Рис. 2.6. Диалоговое окно Attributes

В верхнем правом углу окна расположена кнопка, которая открывает диалоговое окно Entities (рис. 2.7), где в поле Name имеется возможность изменить имя сущности. Измените имя сущности на «Заказ». После завершения ввода нажмите кнопку OK.

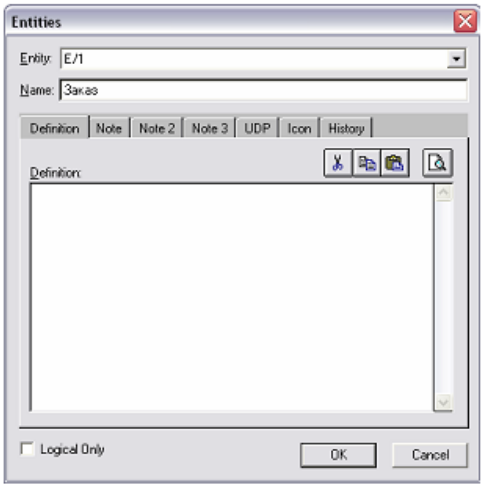


Рис. 2.7. Диалоговое окно Entities

Нажатием на кнопку New... в левом нижнем углу окна открывается диалоговое окно New Attribute создания нового атрибута сущности (рис. 2.8).

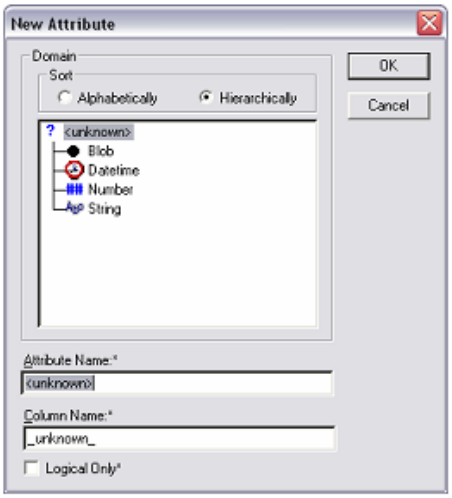


Рис. 2.8. Диалоговое окно New Attribute

В поле Attribute Name:* введите название атрибута «Код_заказа», в окошке Domain выберите тип вводимых данных. Для завершения процедуры ввода нажмите OK. Для задания первичного ключа в окне Attributes для выделенного атрибута «Код_заказа» установите галочку Primary Key.

Возможен так же и другой способ ввода атрибутов в поля сущности. Чтобы заполнить окошко сущности атрибутами, нужно выбрать сущность левой кнопкой мыши и нажать клавишу «Tab». После ввода названия атрибута надо нажать «Enter», если необходимо остаться в верхней части и продолжить заполнять ключевые атрибуты, или «Tab» если необходимо приступить к заполнению других атрибутов. Клавиша «Tab» перемещает курсор между тремя полями ввода: названием сущности, ключевыми атрибутами и не ключевыми атрибутами. Задайте еще три атрибута в сущности Заказ: «Стоимость», «Дата_заказа» и «Дата_выдачи». Результат показан на рис. 2.9.

Рис. 2.9. Сущность Заказ

Одним из предложенных способов создайте и заполните все необходимые сущности компании, описанные в таблице

Сущности	Атрибуты
Клиент	- Код_клиента (PK) - Фамилия - Имя - Отчество - Адрес - Номер_счета
Заказ	- Код заказа (PK) - Дата_заказа - Дата_выдачи - Стоимость
Сотрудник	- Код_сотрудника (PK) - Фамилия - Имя - Отчество - Дата_рождения - Должность - Зарплата
КПК	- Код_КПК (PK) - Процессор - Память - Дисплей - Модем - Интерфейс
ПО	- Код_ПО (PK) - ОС - Мультимедиа - Интернет - Антивирус - Игры
Товар	- Код_товара (PK)
Продавец	- Код_продавца (PK)
Кладовщик	- Код_кладовщика (PK)
Оператор сборки	- Код_оператора_сборки (PK)
Оператор установки	- Код_оператора_установки (PK)

Порядок выполнения работы:

1.Исходя из выбранной ранее темы создать концептуальную модель системы

2.Оформить отчет о проделанной работе.

Раздел 1. Проектирование ИС
Тема 1.2. Технологии разработки АИС (16 часов)

Практическая работа №6
«Создание инфологической модели системы».

Задачи обучающегося:
1.Приобрести навыки построения инфологической модели для заданной предметной области на основе ER-модели
Опорные понятия: инфологическая модель.
Планируемый результат:
Студент должен
Уметь правильно осуществлять информационную постановку задач по обработке информации.
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, специализированное ПО

Алгоритм деятельности обучающегося:

Инфологическая модель позволяет представить предметную область в формализованном виде. При этом используются наиболее естественные для человека формы представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому для построения инфологической модели данных используют различные виды семантических моделей: семантические сети, модель «сущность-отношение» и др. В лабораторной работе используется модель «сущность-отношение» (ER-модель). Основными элементами этой модели являются сущности, их свойства (атрибуты) и связи между сущностями.

Сущность определяет некоторый объект предметной области, информацию о котором необходимо хранить в БД. При этом объекты могут быть как реальными (например, студент), так и абстрактными, (например, зачет). Объекты предметной области образуют классы объектов, имеющих одинаковые свойства. Такие классы определяют *тип сущности* (в дальнейшем, просто сущность). Понятие тип сущности относится к набору однородных личностей, предметов (напр. сущность Студент, характеризующаяся фамилией, номером группы, номером зачетной книжки). Конкретный объект из этого класса задает *экземпляр сущности*. Экземпляр сущности определяет конкретный объект в наборе. Напр., конкретного студента, для которого указанные характеристики имеют конкретные значения: Семенов Гр1 1232009. Характеристики сущности называются атрибутами. Так, фамилия, номер группы, номер зачетной книжки являются атрибутами сущности Студент. Имена атрибутов должны быть уникальным для конкретного типа сущности. Для разных типов сущностей имена атрибутов могут повторяться (напр., атрибут Фамилия может быть определен для многих сущностей: преподаватель, студент, автор книги и т.п.). Атрибуты определяют структуру информации, которая должна быть собрана о сущности.

Среди всех атрибутов сущности особое положение занимают атрибуты, образующие ключ. *Ключ* − это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Различают сильные и слабые типы сущностей. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных.

Другим элементом ER-модели является связь. *Связь* определяет зависимость двух или более сущностей. С помощью связей отражаются семантические соотношения между сущностями. Наличие множества связей и определяет сложность инфологических моделей. При построении ER-модели используются бинарные связи, т.е. связи между двумя сущностями. Различают следующие типы связей:

Связь ОДИН-К-ОДНОМУ (1:1). Такая связь определяет то, что в каждый момент времени каждому экземпляру сущности А соответствует 1 или 0 экземпляров сущности В (рис.1.1): студент занимается в одной аудитории

Рис. 1.1.Связь 1:1

Связь ОДИН-КО-МНОГИМ (1:М). Она означает, что одному экземпляру сущности А соответствуют несколько экземпляров сущности В (рис.1.2.): Преподаватель обучает многих студентов

Рис. 2. Связь 1:М

Связь МНОГИЕ-КО-МНОГИМ (M:N). Она означает, что многим экземплярам сущности А соответствует много экземпляров сущности (рис. 1.3): Преподаватели обучают студентов.

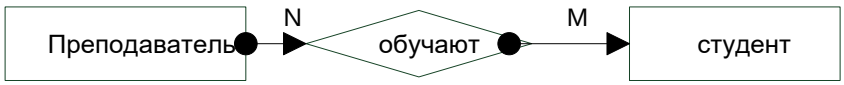


Рис. 1.3. Связь N:M

Все связи требуют описания, которое включает:

идентификатор связи;

формулировку имен связи с точки зрения связываемых сущностей;

тип связи.

Особое внимание при построении модели БД необходимо уделять обеспечению целостности данных. *Целостность* − свойство БД, которое понимается, как способность БД поддерживать правильность данных в любой момент времени. Поддержание целостности может рассматриваться как защита данных от неверных изменений или разрушений. Значительная часть правил, определяющих целостность БД, зависит от смысловых правил работы с данными в конкретной предметной области. Поэтому при проектировании ER-модели необходимо подробно определить все смысловые ограничения и правила, связанные с обработкой данных.

Средства разработки инфологической модели в AllFusion ERwin Data Modeler

Для построения ER-модели предметной области используется CASE- средство AllFusion ERwin Data Modeler (далее ERwin). Это CASE-средство используется для проектирования и документирования БД. ERwin позволяет наглядно отображать сложные структуры данных и имеет удобный графический интерфейс. ER-модель строится в виде логической модели используемого CASE-средства.

Основными компонентами диаграммы являются сущности, атрибуты и связи. Построение модели предполагает определение того, какая информация должна отображаться в каждой сущности и в атрибутах конкретной сущности. Каждая сущность должна иметь уникальное имя, которое отражает её смысловое значение. Кроме того, для каждой сущности необходимо задать её подробное описание и особенности взаимодействия с другими сущностями.

Сущности в ERwin отображаются прямоугольниками (рис. 1.4.). Название сущности отображается над ним.

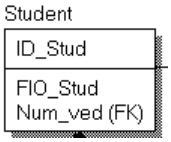


Рис. 1. 4. Пример сущности в ERwin

Сущность характеризуется своими атрибутами, для которых в ERwin необходимо определить имя, тип принимаемого значения, краткое описание, смысловые правила использования значений атрибутов. Имена заданных для сущности атрибутов отображаются внутри прямоугольника. В верхней части прямоугольника, задающего сущность, отражаются атрибуты, образующие ключ, в нижней − остальные атрибуты. Имена атрибутов должны иметь четкие смысловые значения и быть уникальными в рамках свей модели, а не только описываемой сущности.

В ERwin можно определить следующие типы данных для атрибутов:

Integer −целый,

Real − вещественный,

Date − дата,

Varchar − строка символов,

Money − денежный.

В ERwin между сущностями можно установить связи нескольких типов. В CASE-средстве различают несколько типов связей:

идентифицирующую связь "один-ко-многим",

связь "многие-ко-многим",

неидентифицирующую связь "один-ко-многим",

категориальную связь, которая связывает сущности отношением типа тип-подтип.

В зависимости от используемых связей между сущностями в AllFusion ERwin Data Modeler различают зависимые и независимые сущности. Идентифицирующая связь устанавливается между независимой (родительский конец связи) и зависимой (дочерний конец связи) сущностями. При использовании идентифицирующей связи дочерняя сущность автоматически преобразуется в зависимую, которая изображается прямоугольником со скругленными углами. Экземпляр зависимой сущности определяется только через отношение к родительской сущности, т.е. он не может существовать, если для него нет экземпляра родительской сущности. При установлении идентифицирующей связи атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Эта операция дополнения атрибутов дочерней сущности при создании связи называется миграцией атрибутов. В дочерней сущности новые атрибуты помечаются как внешний ключ − FK.

Для каждой связи задается мощность и имя связи. Имя связи − это фраза, характеризующая отношение между родительской и дочерней сущностями. Мощность связей служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней. Мощность задается одним из четырех типов:

общий случай, когда одному экземпляру родительской сущности соответствуют 0, 1 или много экземпляров дочерней сущности; такая связь не помечается каким-либо символом;

символом P помечается случай, когда одному экземпляру родительской сущности соответствуют 1 или много экземпляров дочерней сущности (исключено нулевое значение);

символом Z помечается случай, когда одному экземпляру родительской сущности соответствуют 0 или 1 экземпляр дочерней сущности (исключены множественные значения);

цифрой помечается случай точного соответствия, когда одному экземпляру родительской сущности соответствует заранее заданное число экземпляров дочерней сущности.

Различают несколько видов зависимых сущностей:

Характеристическая − связана только с одной родительской сущностью и хранит информацию о её характеристиках;

Ассоциативная − с несколькими родительскими сущностями и отражает особенности взаимосвязи этих сущностей;

Категориальная − отражает иерархию наследования сущностей, которая представляет собой особый тип объединения сущностей, имеющих общие характеристики.

При установлении идентифицирующих связей между сущностями в дочерней сущности автоматически создаются внешние ключи. Атрибут внешнего ключа обозначается символом (FK).

Последовательность выполнения работы:

Запустить программу AllFusion ERwin Data Modeler и ознакомиться с её интерфейсом для работы с логической моделью.

Выделить необходимый набор сущностей, отражающих заданную предметную область и информационные потребности пользователей в ней. Варианты заданий для выбора предметной области приведены в Приложении I.

Средствами AllFusion ERwin Data Modeler построить и описать ER-модель для заданной предметной области. Для каждой выделенной сущности необходимо задать её полное описание.

Для каждой сущности определить необходимый набор атрибутов и задать их полное описание.

Определить смысловые ограничения и правила обработки каждого атрибута.

Выделить ключевые атрибуты для каждой сущности.

Определить сущности вида подтип/супертип, где это необходимо.

На основе анализа взаимодействия сущностей в предметной области определить типы связей между ними.

Описать каждую связь.

Построить отчет по созданной модели, используя генератор отчетов Report Builder.

Оформить отчет по лабораторной работе.

1. Требования к оформлению отчета

Отчет по лабораторной работе должен включать следующие разделы:

1. Титульный лист. Формат титульного листа приведен в Приложении II.
2. Краткое описание выбранной предметной области
3. Графическая ER-модель
4. Документирование ER-модели, созданное средствами Report Builder в виде отчета, которое должно содержать следующие описания:
    - описание сущностей: имя, краткое описание (Definition), ключ
    - описание атрибутов: имя, тип, краткое описание (Definition), смысловые ограничения на значения (Note)
    - Описание связей: имя, мощность

Контрольные вопросы

1. Каковы задачи, решаемые на этапе инфологического проектирования?
2. В чем состоит отличие понятия типа сущности и элемента сущности?
3. Какие типы сущностей различают в CASE-средстве AllFusion ERwin Data Modeler?
4. Назовите основные описатели атрибута в AllFusion ERwin Data Modeler?
5. Назовите основные типы связей в AllFusion ERwin Data Modeler?
6. Что такое внешний ключ?
7. Как формализуется связь 1:1?
8. Как формализуется связь 1:M?
9. Как формализуется связь M:N?
10. Определите основные шаги формирования отчета средствами AllFusion ERwin Data Modeler.

Раздел 1. Проектирование ИС

Тема 1.3. Аппаратно-программные платформы разработки ИС (4 часа)

Практическая работа №7

«Оптимизация выбора состава программного обеспечения ИС для определенной предметной области».

Задачи обучающегося:

1. Составить и проанализировать требования к информационной системе, оформить техническое задание на разработку программного обеспечения

Опорные понятия: выбор ПО.

Планируемый результат:

Студент должен

Уметь правильно составить и проанализировать требования к информационной системе, оформить техническое задание на разработку программного обеспечения
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, специализированное ПО

1.Общие сведения

1.1. Наименование системы

«Отслеживание почтовых отправлений»

1.2. Назначение и цели создания системы

Удобное отслеживание почтовых отправлений по идентификатору.

Цели:

- Поиск информации по почтовому отправлению с помощью идентификатора или отслеживающему номеру;
- Хранение данных о почтовом отправлении;
- Внесение новых почтовых отправлений;
- Корректировка и обработка почтовых отправлений.

2.Характеристика объектов информатизации

2.1. Краткое описание работы системы «Отслеживание почтовых отправлений»

Отслеживание почтовых отправлений по указанному идентификатору, предоставляя информацию о этом отправлении.

2.2. Описание объектов информатизации

Данные о новом почтовом отправлении вносятся в базу данных, где имеется специальный идентификатор, по которому найдет данные отправления.

3.Требования к информационной системе

3.1. Базовые принципы разработки подсистем

При проектировании и разработке подсистем должны использоваться следующие базовые принципы:

- Требуемое время реакции системы на запрос;
- Простота в эксплуатации и поддержки системы.

Система должна содержать:

- Средства поиска информации;
- Окно для показа времени и даты.

3.2. Требования к архитектуре системы.

Разрабатываемая система имеет архитектуру клиент-сервер. В качестве клиентского приложения выступает стандартный веб-браузер.

3.3. Требования к способам и средствам связи для информационного обмена между компонентами (модулями) системы

Система «Отслеживание почтовых отправлений» работает в пределах сети Интернет.

3.4. Требования к режимам функционирования системы

Разрабатываемая система функционирует 24 часа в сутки.

3.5. Требования к пользователям

Разрабатываемая система не требует специальных знаний.

3.6. Требования к численности и квалификации персонала системы и режиму его работы.

Разрабатываемая система поддерживает 2 вида пользователей: сотрудника «Почты» для внесения информации о почтовом отправлении и обычный пользователь, которому необходимо найти информацию о почтовом отправлении.

3.7. Требования к защите информации от несанкционированного доступа.

Особых требований к защите информации не требуется.

3.8. Внесение корректировок в программный продукт, связанных с ошибками в Системе

Время для исправления ошибок в системе не более одной недели.

Порядок выполнения работы:

1.По образцу выше составьте техническое задание согласно выбранной темы

2.Оформите отчет

Ответы на контрольные вопросы:

1. Что такое пользовательские требования?
2. Что такое системные требования?
3. Расскажите о разработке требований
4. Представьте процесс разработки требований
5. Расскажите о формировании и анализе требований
6. Расскажите о VORD и основных этапах метода.
7. Что такое аттестация требований?

Раздел 1. Проектирование ИС
Тема 1.5. Проектирование серверной части АИС (8 часов)

Практическая работа №8
«Установка серверного ПО ИС на аппаратные сервера и его дальнейшее сопровождение».

Задачи обучающегося:
1. Научиться устанавливать сервер БД и среду для разработки приложений
Опорные понятия: установка ПО.
Планируемый результат:
Студент должен
Уметь устанавливать и настраивать серверное ПО для создания и функционирования ИС
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, специализированное ПО, интернет

Порядок выполнения работы:

1.Перед началом выполнения изучите видеоурок, который доступен по ссылке

https://www.youtube.com/watch?v=YXcdh0j4UHY&feature=emb_title

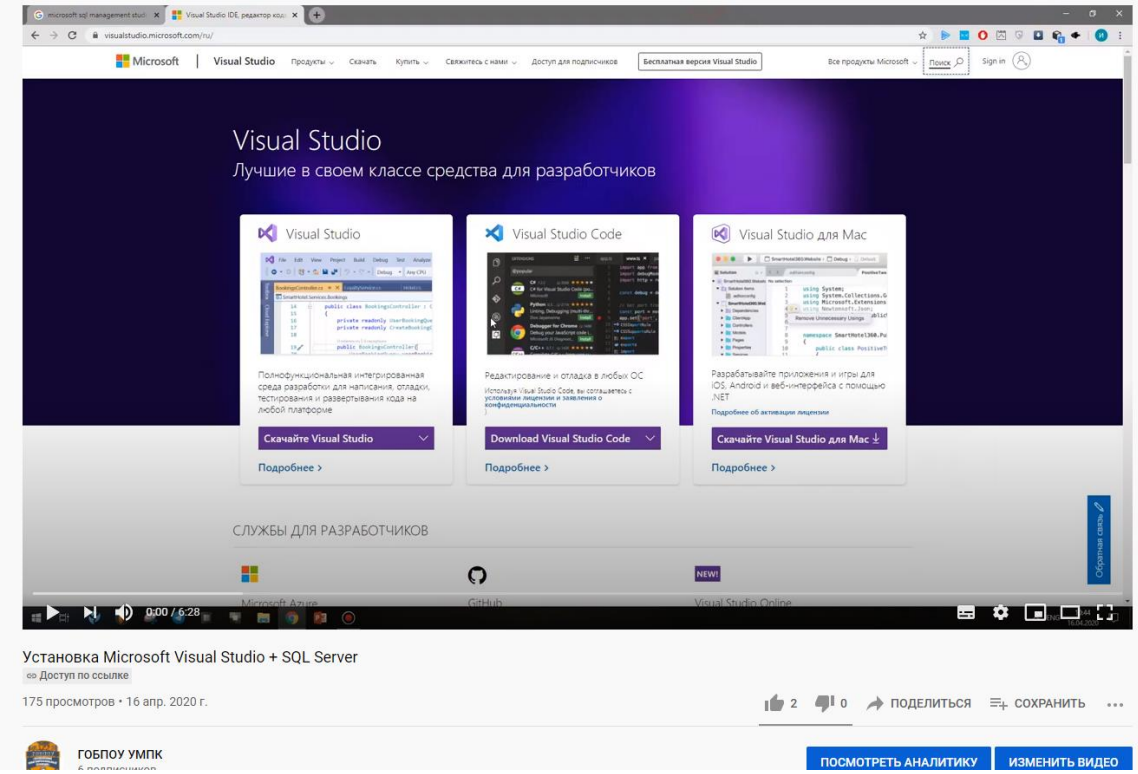


Рис.1 – Установка ПО

2.Повторите все выполненные действия, параллельно фиксируя результаты в отчет. Должно быть описание каждого этапа со скриншотами

Контрольные вопросы:

1.Чем отличается БД от СУБД

2.Как расчитать характеристики сервера в зависимости от нагрузки

Раздел 1. Проектирование ИС
Тема 1.5. Проектирование серверной части АИС (8 часов)

Практическая работа №9

«Организация работы ПО ИС в локальных сетях. Особенности настройки и сопровождения».

Задачи обучающегося:

1. Научиться устанавливать и выполнять первоначальную настройку серверной ОС семейства Windows

Опорные понятия: установка ПО.

Планируемый результат:

Студент должен

Уметь устанавливать и настраивать серверное ПО для создания и функционирования ИС

Пользоваться инструментальными средствами программного обеспечения

Необходимое оборудование: учебная литература, VirtualBox, ПК
Порядок выполнения работы:

Операционная система (ОС) – это совокупность программных средств, осуществляющая управление ресурсами компьютера, запуск прикладных программ и их взаимодействие с внешними устройствами и другими программами, а также обеспечивающая диалог пользователя с ЭВМ. Ресурсом является любой компонент ЭВМ и предоставляемые им *возможности*: центральный процессор, оперативная и внешняя память, внешнее устройство, программа и т.д. ОС загружается в оперативную память при включении компьютера и предоставляет пользователю удобный способ общения (*интерфейс*) с вычислительной системой.

Назначение серверной операционной системы - это управление приложениями, обслуживающими всех пользователей корпоративной сети, а нередко и внешних пользователей. К таким приложениям относятся современные системы управления базами данных, средства управления сетями и анализа событий в сети, службы каталогов, средства обмена сообщениями и групповой работы, Web-серверы, почтовые серверы, корпоративные брандмауэры, серверы приложений самого разнообразного назначения, серверные части бизнес-приложений. Требования к производительности и надежности указанных операционных систем очень высоки; нередко сюда входят и поддержка кластеров (набора ряда однотипных компьютеров, выполняющих одну и ту же задачу и делящих между собой нагрузку), и возможности дублирования и резервирования, и переконфигурации программного и аппаратного обеспечения без перезагрузки операционной системы.

Выбор серверной операционной системы и аппаратной платформы для нее в первую очередь определяется тем, какие приложения под ее управлением должны выполняться (как минимум, выбранные приложения должны существовать в версии для данной платформы) и какие требования предъявляются к ее производительности, надежности и доступности.
Windows

Серверные версии операционной системы **Windows (*Windows NT,Windows 2000, Windows Server 2003*)** сегодня применяются довольно широко - благодаря удобству администрирования и невысокой совокупной стоимости владения.

Windows NT

Первая полностью 32-разрядная операционная система этого семейства, появилась вскоре после выпуска **Windows 95**. Самой популярной стала версия **Windows NT Server 4.0**, существовавшая в варианте не только для *Intel-совместимых компьютеров*, но и для *RISC-систем*. Данная операционная система обладала привычным пользовательским интерфейсом **Windows 95**, удобными средствами администрирования, встроенным Web-сервером, средствами диагностики сети, управления процессами и задачами, интеграции с другими операционными системами (например, с ***Novell NetWare***), а также утилитами и службами управления рабочими станциями. Чуть позже для этой операционной системы появились такие сервисы, как монитор транзакций и сервер приложений **Microsoft Transaction Server**, сервер управления очередями сообщений**Microsoft Message Queue Server**, а также ряд коммерческих продуктов, в том числе серверные СУБД, средства групповой работы и обмена сообщениями, серверы приложений как от компании **Microsoft**, так и от других производителей.

Windows 2000

На данный момент является самой популярной операционной системой **Microsoft** в корпоративном секторе. К серверным операционным системам этого семейства относятся **Windows 2000 Server** - универсальная сетевая операционная система для серверов рабочих групп и отделов, **Windows 2000 Advanced Server** - операционная система для эксплуатации бизнес-приложений и приложений для электронной коммерции и **Windows**

2000 Datacenter Server - ОС для наиболее ответственных приложений, осуществляющих обработку данных. В состав **Windows 2000 Server**, по сравнению с **Windows NT 4.0**, включены и дополнительные службы, облегчающие управление серверами, сетями и рабочими станциями, например службы каталогов ***Active Directory***, позволяющие создать единое хранилище учетных записей пользователей, клиентов, серверов и приложений **Windows**, дополнительные средства конфигурирования сетей и подключения удаленных пользователей, терминальные службы для удаленного управления компьютерами. Кроме того, в данную операционную систему были добавлены службы компонентов, являющиеся дальнейшим развитием ***Microsoft Transaction Server***, что позволило создавать для этой ОС корпоративные приложения, обладающие масштабируемостью и надежностью.

Windows Server 2003

Создание семейства **Windows Server 2003** стало следующим шагом в развитии операционных систем **Windows 2000**. Основными особенностями данного семейства операционных систем являются наличие в их составе платформы **Microsoft .NET Framework**, а также поддержка **Web-сервисов XML**. **Windows Server 2003** существует в четырех редакциях:

- ***Windows Server 2003 Web Edition*** - операционная система для развертывания и обслуживания *Web-приложений* и *Web-сервисов*, включая приложения **.ASP .NET**;
- ***Windows Server 2003 Standard Edition*** - сетевая операционная система для выполнения серверной части бизнес-решений и рассчитанная на применение в небольших компаниях и подразделениях. Здесь имеются средства совместного использования ресурсов и централизованного развертывания приложений для настольных компьютеров, а также реализована поддержка до 4 Гбайт оперативной памяти и симметричной многопроцессорной обработки с использованием двух процессоров;
- ***Windows Server 2003 Enterprise Edition*** - ОС, которая прежде всего предназначена для средних и крупных компаний. Она поддерживает серверы на базе 64-разрядных процессоров (до восьми штук) и объем оперативной памяти до 64 Гбайт и выпускается в версиях для 32- и 64-разрядных платформ;
- ***Windows Server 2003 Datacenter Edition*** - операционная система, которая служит для создания критически важных технических решений с высокими требованиями к масштабируемости и доступности. К таким решениям относятся приложения для обработки транзакций в режиме реального времени, а также решения, основанные на интеграции нескольких серверных продуктов. В данной ОС реализована поддержка симметричной многопроцессорной обработки с использованием до 32 процессоров, а также имеются службы балансировки нагрузки и создания кластеров, состоящих из восьми узлов. Эта ОС доступна для 32- и 64-разрядных платформ.

Процесс установки серверной ОС семейства **Windows** практически ничем не отличается от установки других ОС указанного семейства, кроме необходимости указания в процессе установки количества одновременных подключений.

UNIX

Операционная система ***UNIX*** относится к «долгожителям» рынка серверных операционных систем - она была создана в конце 60-х годов в **Bell Laboratories** фирмы **AT&T**. Отличительной особенностью этой ОС, обусловившей ее «живучесть» и популярность, было то, что ядро операционной системы, написанной на ассемблере, было невелико, тогда как вся оставшаяся часть операционной системы была написана на языке С. Такой подход делал легко переносимой на самые разнообразные аппаратные платформы и саму операционную систему, и созданные для нее приложения. Важным достоинством ***UNIX*** стала ее открытость, позволившая одновременно существовать как коммерческим, так и некоммерческим версиям ***UNIX***.

Общими для всех версий ***UNIX*** особенностями являются многопользовательский режим со средствами защиты данных от несанкционированного доступа, реализация мультипрограммной обработки в режиме разделения времени, использование механизмов виртуальной памяти и свопинга, унификация операций ввода-вывода, иерархическая файловая система, разнообразные средства взаимодействия процессов, в том числе межсетевого.

Solaris (Sun Microsystems)

Операционная система ***Sun Solaris*** сегодня входит в число самых известных коммерческих версий ***UNIX***. Эта ОС обладает развитыми средствами поддержки сетевого взаимодействия и представляет собой одну из самых популярных платформ для разработки корпоративных решений - для нее существует около 12 тыс. различных приложений, в том числе серверов приложений и СУБД почти от всех ведущих производителей.***Solaris*** соответствует многим промышленным стандартам и характеризуется высокой масштабируемостью. Для подавляющего большинства приложений эта операционная система обеспечивает практически линейный рост производительности при увеличении числа процессоров за счет симметричных многопроцессорных вычислений. В настоящее время Solaris поддерживает процессоры **SPARC** и **Intel x86**. Из *особенностей* ***Solaris 9***следует отметить поддержку до 1 млн. одновременно работающих процессов, до 128 процессоров в одной системе и до 848 процессоров в кластере, до 576 Гбайт физической оперативной памяти, поддержку файловых систем размером до 252 Тбайт, наличие средств управления конфигурациями и изменениями, встроенную совместимость с ***Linux***. Операционная система ***Solaris 9*** представляет собой основу открытой сетевой среды **Sun Open Net Environment (Sun ONE)**. В комплект поставки ***Solaris 9*** входят ключевые приложения **Sun ONE:** *Application Server, Directory Server, Integration Server, Message Queue, Portal Server, Web Server*.

HP-UX (Hewlett-Packard)

Операционная система ***HP-UX***, разработанная в компании **Hewlett-Packard**, является потомком **AT&T System V**. Ее последняя версия, **HP-UX 11i**, доступна для двух аппаратных платформ - **PA_RISC** и**Itanium** - и ориентирована главным образом на серверы производства **Hewlett-Packard**. Из *особенностей* ***HP-UX 11i*** нужно назвать средства интеграции с ***Windows*** и ***Linux***, в том числе средства переноса *Java- приложений*, разработанных для этих платформ, а также средства повышения производительности *Java-приложений*. Кроме того, ***HP-UX 11i*** поддерживает ***Linux API***, что гарантирует перенос приложений между ***HP-UX*** и ***Linux***. Отметим, что приложения для ***HP-UX 11i*** переносятся между двумя поддерживаемыми ею аппаратными платформами без изменений и перекомпиляции.

Говоря о производительности и масштабируемости ***HP-UX 11i***, следует отметить, что одна копия операционной системы поддерживает до 256 процессоров; поддерживаются также кластеры размером до 128 узлов. К тому же данная платформа поддерживает подключение и отключение дополнительных процессоров, замену аппаратного обеспечения, динамическую настройку и обновление операционной системы без необходимости перезагрузки, резервное копирование в режиме *on-line* и дефрагментацию дисков без выключения системы.

Выбор программного обеспечения для данной операционной системы весьма широк - это и серверы приложений ведущих производителей, и Web- и WAP-серверы, и поисковые серверы, и средства кэширования, и службы каталогов.

Linux и FreeBSD

Операционная система ***Linux*** - это некоммерческий продукт категории **Open Source** для платформы **Intel**, который в течение десяти лет создавали тысячи энтузиастов. Список серверных продуктов для ***Linux***, пожалуй, не менее внушителен, чем для ***Solaris, HP-***

UX и AIX, и включает такие популярные продукты, как **Web-сервер Apache**, серверные СУБД и серверы приложений практически от всех производителей.

Одним из серьезных преимуществ ***Linux*** является низкая стоимость ее приобретения (хотя сама операционная система является некоммерческим продуктом, сертифицированные дистрибутивы ***Linux*** - обычно продукты коммерческие). Кроме того, ряд компаний, в частности **IBM**, вкладывают значительные средства в развитие ***Linux*** как серверной платформы, одновременно стремясь реализовать совместимость с ***Linux*** в своих коммерческих версиях ***UNIX*** в расчете на возможный переход с ***Linux*** на указанные операционные системы.

NetWare (Novell)

В начале и середине 90-х годов ***Novell NetWare*** была доминирующей сетевой операционной системой. Хотя в настоящее время снизилась доля серверов, управляемых ***NetWare***, как и количество создаваемых для нее приложений и инфраструктурного программного обеспечения, эта операционная система по-прежнему популярна благодаря надежности, масштабируемости, способности управлять большим количеством рабочих станций.

Основными особенностями последней версии данной операционной системы, ***Novell NetWare 6.5***, являются возможность создания географически распределенных кластеров, наличие средств поддержки мобильных и удаленных пользователей, инструментов управления удаленными сетевыми ресурсами, а также средств синхронизации информации о пользователях и приведения в соответствие между собой каталогов в смешанных средах. Защита данных в ***Novell NetWare 6.5*** осуществляется с помощью служб каталогов **NDS eDirectory**.

В состав ***Novell NetWare 6.5*** входят известные **OpenSource**-продукты, а именно: ***Web-сервер Apache, СУБД MySQL, сервер приложений Apache Tomcat***. Кроме того, в ***NetWare 6.5*** включены сертифицированный на соответствие спецификации **J2EE 1.3** сервер приложений и среда разработки **Novell exteNd** и так называемый виртуальный офис, позволяющий через Web-интерфейс обращаться к бизнес-ресурсам пользователя, включая файлы, электронную почту, средства календарного планирования.

Mac OS X (Apple)

Операционная система ***Mac OS X***, созданная компанией **Apple** совместно с рядом университетских ученых, основана на ***BSD UNIX***. В 1999 году версия ***Mac OS X Server*** была выпущена в виде продукта **Open Source**, что позволило разработчикам адаптировать ***Mac OS X*** для конкретных заказчиков, а также привлечь их к дальнейшему развитию этой операционной системы.

Mac OS X характеризуется наличием менеджера виртуальной памяти, возможностью полной изоляции приложений друг от друга, поддержкой многозадачности, сравнимой с аналогичной поддержкой в**Windows**. В ***Mac OS X*** имеются эмулятор предыдущих версий ***Mac OS***, средства редактирования графических изображений, встроенная поддержка **OpenGL**, почтовый клиент, средства управления паролями для доступа к Web-ресурсам. В целом ***Mac OS X*** представляется многообещающей серверной операционной системой, и для нее уже начали выпускаться серверные СУБД и иное инфраструктурное программное обеспечение, хотя корпоративные пользователи пока относятся к ней достаточно осторожно.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Установите Windows 2003 server.

1. Запустите приложение ВМ ***VirtualBox*** и подключите к созданной ранее виртуальной машине **VM-2** образ установочного диска **Windows 2003 Server**.
2. Запустите ВМ и начните установку ОС.

3. Ознакомьтесь с информацией программы установки и нажмите ***Enter***.

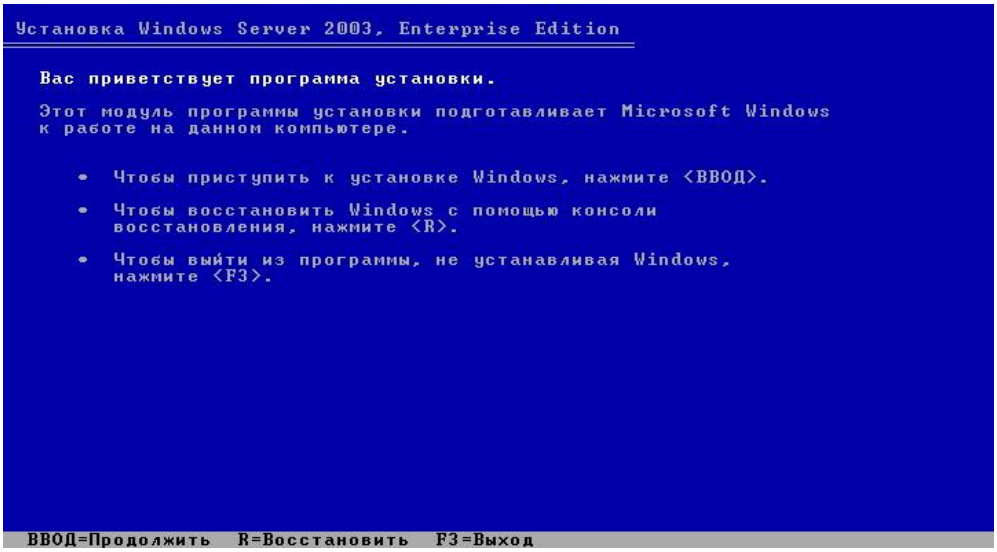


Рис.7. Программа установки.

4. Ознакомьтесь с лицензионным соглашением и согласитесь с ним (клавиша **F8**).
5. Создайте раздел для ОС на всем жестком диске клавишей ***ENTER***.

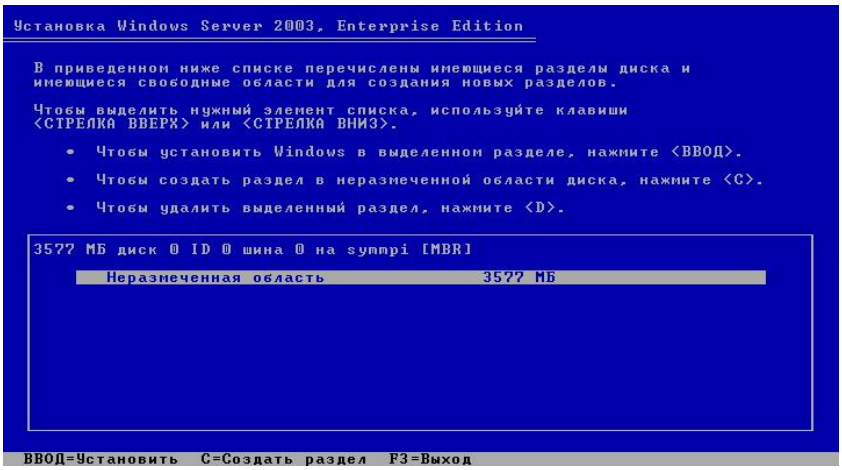


Рис. 8. Создание раздела на жестком диске.

6. Выполните форматирование созданного раздела в файловой системе **NTFS** - нажмите ***ENTER***. *Дождитесь окончания форматирования раздела, и копирования файлов установки на него. В процессе копирования компьютер перезагрузится и продолжит установку автоматически.*
7. Самостоятельно укажите параметры языка и раскладки клавиатуры и перейдите к следующему шагу кнопкой ***Далее***.

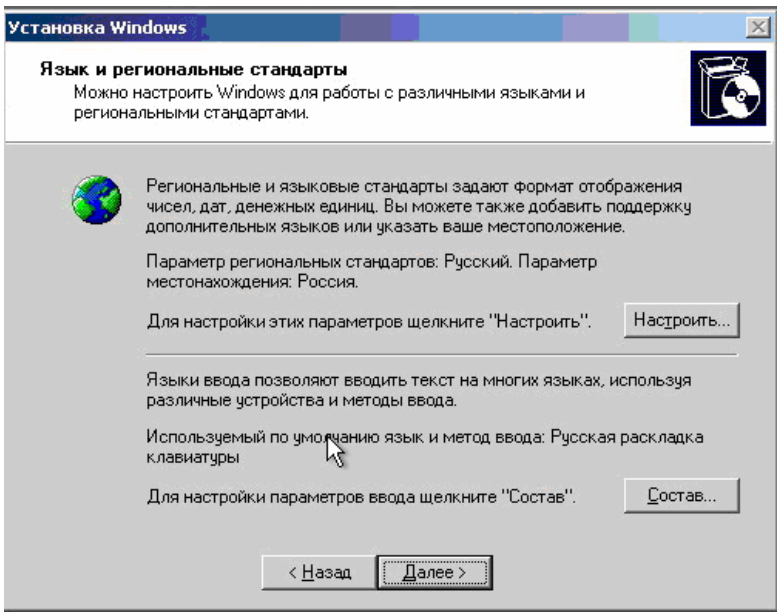


Рис.9. Язык и региональные стандарты.

8. Укажите регистрационные данные:
   - ведите в поле **Имя** – *USER*
   - ведите в поле **Организация** – *MSPU*
   - завершите ввод кнопкой ***Далее***.

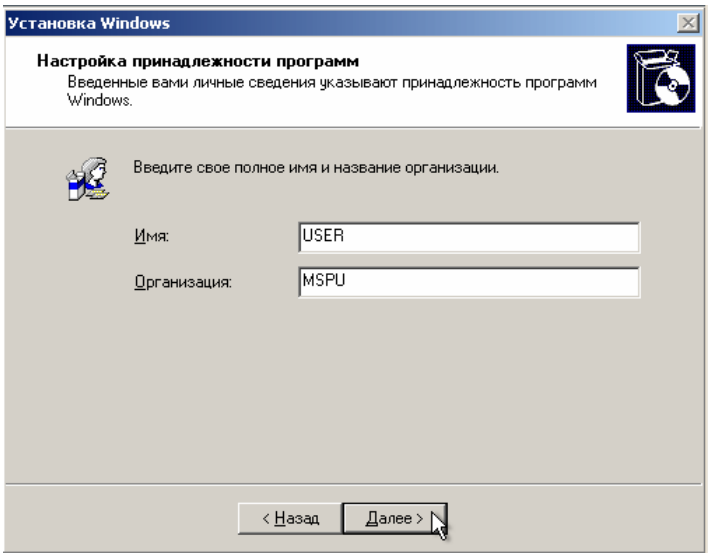


Рис.10. Имя пользователя и название организации

9. Введите в поле **Ключ продукта** лицензионный ключ и щелкните ***Далее***.
10. Укажите вариант лицензирования при котором для каждого подключения требуется отдельная лицензия:
    - установите радиокнопку *На сервере*;
    - введите в текстовое поле количество одновременных подключений, например *10*;
    - подтвердите параметры кнопкой ***Далее***.

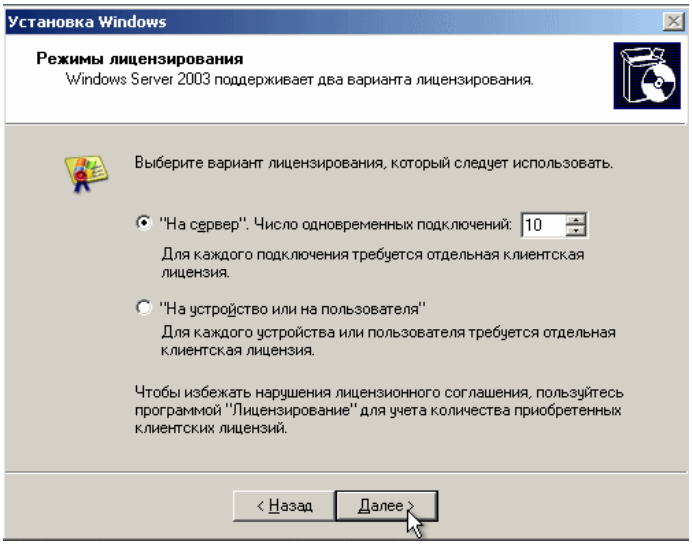


Рис.11. Вариант лицензирования.

11. Укажите имя компьютера и пароль администратора:
    - Введите в поле **Имя компьютера** – *WIN2003*;
    - Введите в поле **Пароль администратора** – *123456*;
    - Введите в поле **Подтверждение** - *123456*.
    - Подтвердите сделанные изменения кнопкой ***Далее***. *Появится диалоговое окно сообщающее о том что пароль слишком простой.*
    - Ознакомьтесь с информацией о том что вы указали простой пароль и продолжите установку кнопкой ***Да***.

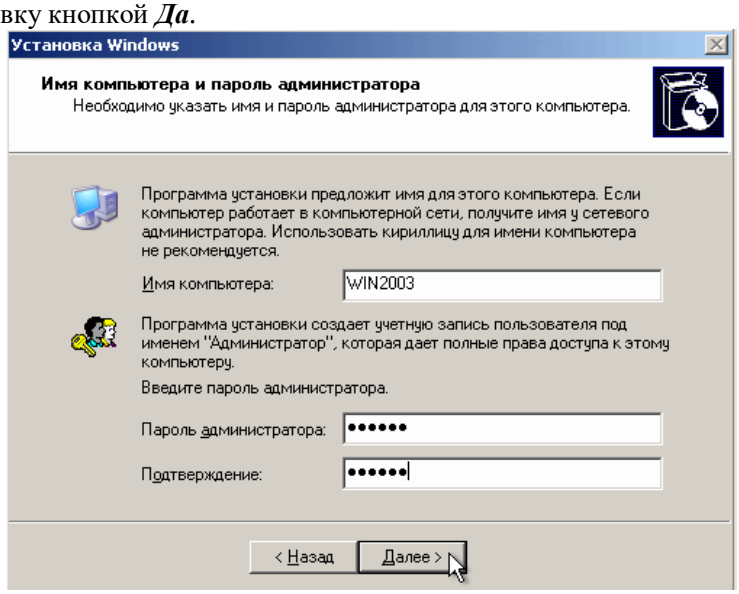


Рис.12. Имя компьютера и пароль администратора.

12. Укажите *дату и время* и щелкните ***Далее***.
13. Установите сетевые параметры для использования статического IP-адреса:
    - выберите радиокнопку *Обычные параметры* и щелкните ***Далее***;
14. Укажите сетевую группу, например *Workgroup* и щелкните ***Далее***.
15. Дождитесь окончания выполнения установки ОС.
    По окончании установки компьютер перезагрузится. После этого загрузится операционная система ***Windows 2003 Server***.

Задание 2. Выполните первоначальную настройку ОС:

1. Выполните вход в систему на сервере:

- вызовите в ВМ событие *нажатие комбинации клавиш CTRL+ALT+DEL*, для этого нажмите **RCTRL+DEL**;
- введите **пароль администратора** – *123456*;
- подтвердите введенные данные кнопкой ***ОК***.

2. установите дополнительные компоненты со 2-го диска:
   - откройте менеджер виртуальных дисков (***команда меню ВМ Устройства/Подключить CD/DVD-ROM/Образ CD/DVD-ROM***);
   - добавьте образ:
     - откройте окно добавления кнопкой ***Добавить***;
     - перейдите в каталог с образами ОС и выберите файл ***win2003-2.iso***;
   - подтвердите выбор образа ***Выбрать***;
   - ознакомьтесь с информацией мастера установки и щелкните ***Далее***;
   - ознакомьтесь с лицензионным соглашением и примите его (***Далее***).
   - активизируйте копирование файлов кнопкой ***Далее***;
   - завершите работу мастера установки кнопкой ***Готово***.
3. Ознакомьтесь с информацией диалогового окна **Послеустановочные обновления безопасности Windows Server** (***Готово***).

   *После нажатия кнопки **Готово** появится окно, предупреждающее о том, что после этого будет разрешено подключение по сети к вашему серверу.*
4. Подтвердите закрытие диалогового окна кнопкой ***Да***.
5. Ознакомьтесь с информацией в окне ***Управление данным сервером*** и закройте его.
6. Выполните настройку сетевого интерфейса:
   - откройте диалоговое окно **Состояние подключения по локальной сети (*Пуск/Панель управления/Сетевые подключения/Подключение по локальной сети*)**;
   - откройте диалоговое окно свойств подключения по локальной сети кнопкой ***Свойства***;
   - откройте диалоговое окно **Параметры протокола Интернет TCP/IP** двойным щелчком;
   - включите использования статического адреса соответствующей радиокнопкой и укажите следующие данные:
     - **IP-адрес** – *192.168.1.2*;
     - **Маска подсети** – *192.168.255.255*;
     - **Основной шлюз** – *192.168.1.1*;
     - завершите ввод параметров кнопкой ***ОК***;
   - закройте диалоговое окно свойств подключения по локальной сети кнопкой ***ОК***;
   - закройте диалоговое окно **Состояние подключения по локальной сети** кнопкой ***Закрыть***.
7. Установите дополнения гостевой ОС:
   - запустите **Мастер дополнений гостевой ОС (*Устройства/Установить дополнение гостевой ОС*)**;
   - ознакомьтесь с информацией мастера и щелкните ***Next***;
   - ознакомьтесь с лицензионным соглашением и согласитесь с ним (***I accept the terms in License Agreement/Next***)
   - подтвердите путь установки дополнений по умолчанию кнопкой ***Install***;
   - подтвердите установку графического адаптера **VirtualBox Graphic Adapter** кнопкой ***Продолжить***;
   - разрешите ОС доверять устанавливаемому драйверу графического адаптера кнопкой ***ДА***;
   - завершите установку дополнений кнопкой ***Finish***.

После этого компьютер перезагрузится

8. Самостоятельно установите в ВМ более высокое разрешение экрана, например **1024x768**.

9. Остановите виртуальную машину.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое операционная система (ОС)?
2. Назначение серверной операционной системы.
3. Расскажите о Windows NT
4. Расскажите о Windows 2000
5. Расскажите о Windows Server 2003
6. Расскажите о Windows NT
7. Расскажите о UNIX
8. Расскажите о Solaris (Sun Microsystems)
9. Расскажите о HP-UX (Hewlett-Packard)
10. Расскажите о Linux и FreeBSD
11. Расскажите о NetWare (Novell)
12. Расскажите о Mac OS X (Apple)

Раздел 1. Проектирование ИС
Тема 1.6. Проектирование клиентской части АИС (20 часов)

Практическая работа №10
«Клиентская часть: размещение не визуальных компонентов, соединение с БД».

Задачи обучающегося:
1. Научиться устанавливать и ознакомится с компонентами ADOConnection, ADOTable и DataSource для связи БД
Опорные понятия: подключение базы данных.
Планируемый результат:
Студент должен
Уметь устанавливать и настраивать соединение с базой данных и подключать ее к клиентскому приложению
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, IDE, ПК

Создать клиентское приложение для работы с БД «Товар». Приложение имеет главную форму (окно), две рабочие формы (окна) и модуля данных. Главное окно содержит меню вызова таблиц БД (рисунок 1). Выбор пункта меню вызывает соответствующее окно для работы с тремя таблицами. Таблицы можно редактировать и добавлять новыми записями.

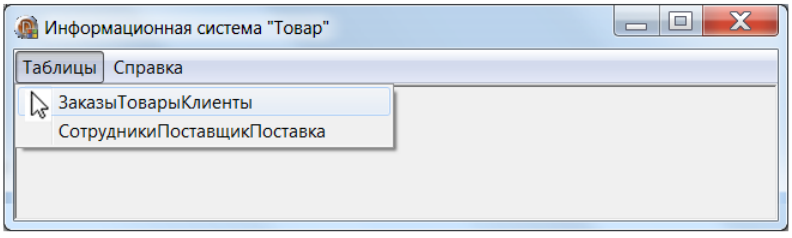


Рисунок 1 – Начальное окно главной формы приложения

В модуле данных размещаются компоненты связи приложения с БД.

Порядок выполнения работы:

1 Создаем главную форму приложения. Размещаем на ней компонент Меню для выбора нужных таблиц.

2 Связываем БД «Товар» с нашим приложением.

3 Создаем две подчиненные формы. На первой будут таблицы «Заказы», «Товары» и «Клиенты». На второй – «Сотрудники», «Поставщик» и «Поставка».

Необходимые компоненты

Data Module – специальная форма, предназначенная исключительно для размещения на ней невизуальных компонент для доступа к данным.

ADOConnection – используется для подключения приложения к БД (закладка dbGO или в поздний версиях ADO).

ADOTable (таблица ADO)– связывается с конкретной таблицей БД (закладка dbGO или в поздний версиях ADO);

DataSource (источник данных) – компонент, используется как связка данных из таблиц, с отображающими и управляющими компонентами Delphi (закладка Data Access);

DBGrid – таблица, позволяющая вывести содержимое таблицы БД на пользовательскую форму (закладка Data Controls);

DBNavigator – кнопочная панель, которая обеспечивает перемещение указателя текущей записи таблицы, активизацию режима редактирования, добавление и удаление записей (закладка Data Controls).

1 Создание нового приложения

1.1 Запускаем Delphi. Выбираем **File -> New-> VCL Forms Application**. Сохраняем модуль **Unit1** и проект приложения под именем **Tovar.dpr** в своей папке. В эту же папку помещаем файл БД Товар.accdb.

1.2 Сделаем нашу форму главной MDI формой (многодокументный интерфейс). Для этого в инспекторе объектов в свойствах Form1 свойство **FormStyle** установим в **fsMDIForm**. Форма стала стартовой.

1.3 В свойстве **Caption** главной формы пишем «Информационная система Товар».

1.4 Добавим на форму компонент TMainMenu из вкладки Standard. Щелкнем дважды мышкой на компоненте MainMenu1 и увидим окно для создания меню.

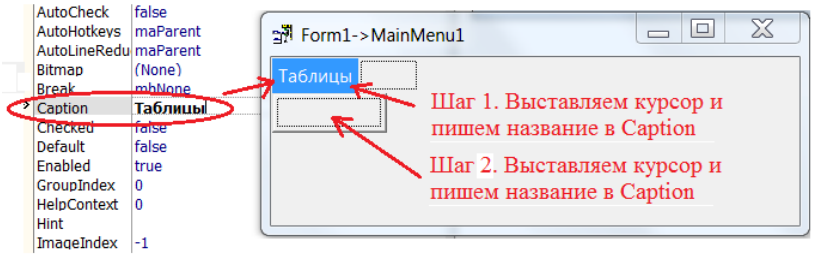


Рисунок 2 – Настройка компоненты MainMenu1

Выставляем курсор на «прямоугольник» и в появившемся инспекторе свойств пишем Таблицы (рисунок 2), нажимаем Enter. Перемещаем курсор на «прямоугольник» ниже «Таблицы». В появившемся инспекторе свойств пишем «ЗаказыТоварыКлиенты» и нажимаем Enter, а потом – «СотрудникиПоставщикПоставка». Перемещаем курсор на «прямоугольник» правее «Таблицы» и пишем «Справка».

1.4 Компилируем и запускаем программу (F9).

2 Привязка БД «Товар.accdb» к приложению

2.1 Командой File / New / Data Module (в категории Delphi Files) добавьте в проект новый модуль данных. Дайте ему имя **dm** (для краткости) и сохраните его как **DatMod.pas** в папку приложения.

2.2 Положите на модуль **dm** компонент **ADOConnection** (в закладке dbGo). Компонент ADOConnection1 позволит привязать БД **Товар.accdb** к приложению.

Настраивается компонент **ADOConnection.** Дважды щелкните на этом компоненте. В открывшемся окне установите источник связи, выбрав **Use Connection String** (рисунок 3).

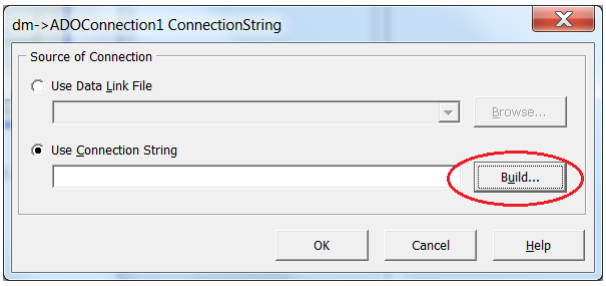


Рисунок 3 – Окно модуля данных

Нажмите кнопку Build..., перейдите на закладку «Поставщик данных» и выберите **Microsoft Office 12.0 Access Database Engine Provider** (рисунок 4). Нажмите кнопку **Далее**.

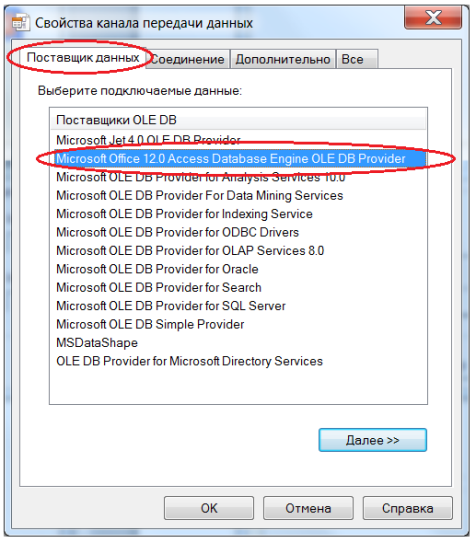


Рисунок 4 – Окно свойств связи с данными

Перейдите за закладку **Соединение**, введите в строку **Источник данных** имя нашей БД, как показано на рисунке 5. Если для доступа к БД необходим пароль и идентификатор пользователя, то их надо указать (по умолчанию к БД, созданной в MS Access, доступ есть у пользователя Admin, но пароль не нужен). После этого нажмите кнопку **Проверить подключение,** чтобы проверить подключается ли БД Товар.accdb. Если будет дан ответ «Проверка подключения выполнена», то щелкая по кнопке ОК закрыть все окна.

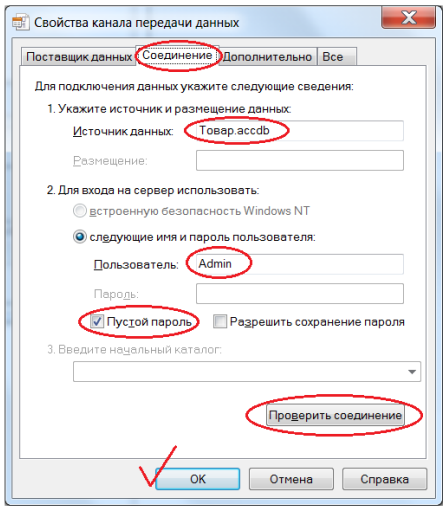


Рисунок 5 – Подключение БД Товар.accdb к приложению

2.3 Установим свойства **ADOConnection** в **Инспекторе объектов**:

- **LoginPrompt = False** – запрос имени и пароля пользователя отключен.
- **Mode** = **cmShareDenyNone**.
- **Connected** = **True** – подключение к базе активировано.

3 Связывание таблиц БД с приложением

3.1 Положите на модуль данных **dm** компоненты **ADOTable** (закладка dbGo) и **DataSource** (закладка DataAccess).
3.2 Для удобства работы дайте им имена **TabZakaz** и **dsZakaz**, соответственно. Для этого в инспекторе объектов **ADOTable1** в свойство **Name** записываем **TabZakaz**, а в инспекторе объектов **DataSource1** в свойство **Name** записываем **dsZakaz.**
3.3 Подключим таблицу **TabZakaz** к компоненту **ADOConnection1** и к одноименной таблице **«Заказы»** нашей БД (рисунок 6).
В инспекторе объектов таблицы **TabZakaz** устанавливаем свойства:

- **Connection = ADOConnection1.**
- **TableName = Заказы.**
- **Active = True**.

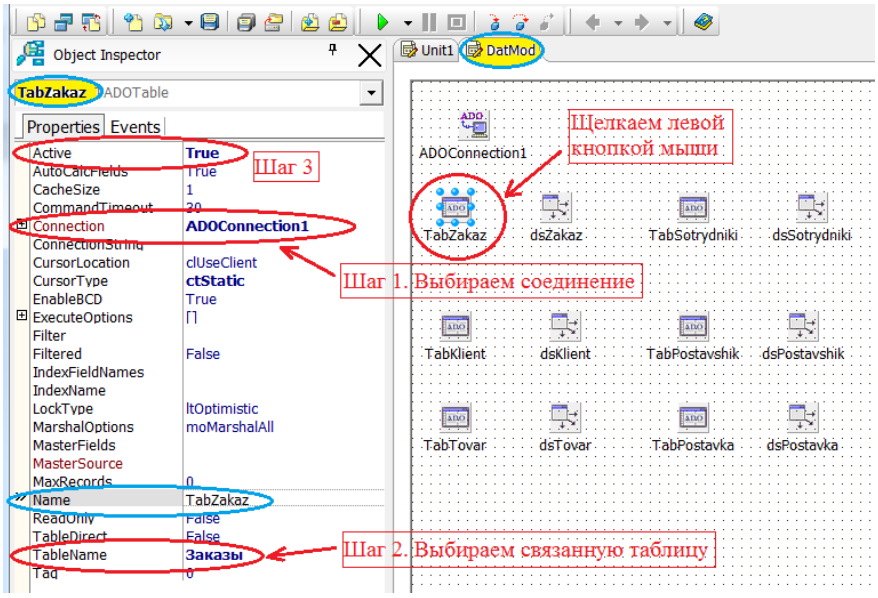


Рисунок 6 – Установка свойств компонента TabZakaz (ADOTable1)

3.4 В инспекторе объектов компонента **dsZakaz** (бывшая DataSource1) установите свойство **DataSet** = **TabZakaz** (рисунок 7).

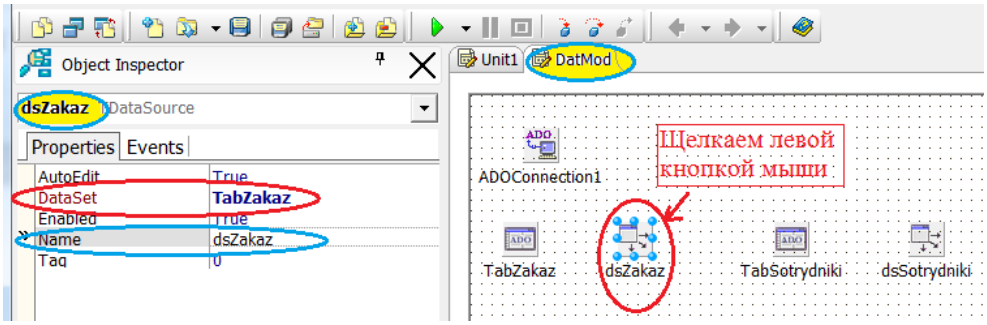


Рисунок 7 – Установка свойств компонента dsZakaz (DataSource1)

После определения свойств исчезнут красные знаки вопроса слева от компонентов в окне **Stucture**, что говорит о готовности компонента к работе.

3.5 Двойным щелчком на компоненте **TabZakaz** откроем **Окно редактора полей**, щелкнем в окне правой кнопкой мыши и в контекстном меню выберем команду **Add all fields**. Окно

редактора заполнится списком всех полей таблицы **Заказы**. Если щелкнуть на любом поле в окне редактора полей, то в окне инспектора объектов станут доступными свойства объекта-поля.

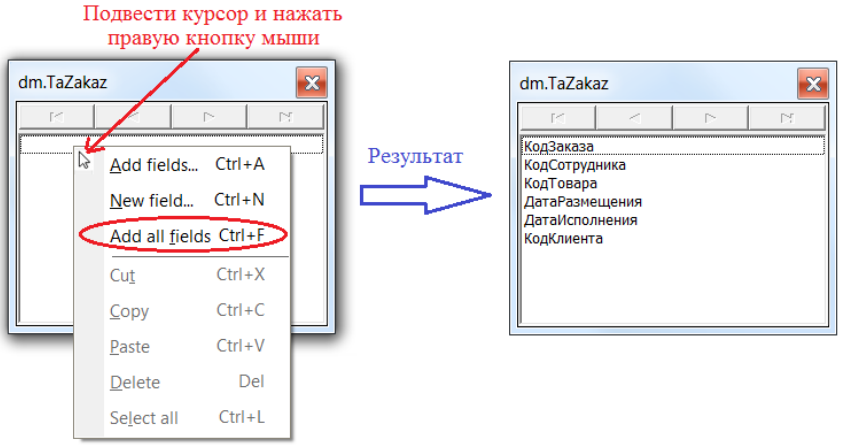


Рисунок 8 – Добавление полей в компонент TabZakaz

3.6 Поступая аналогичным образом, установите в модуль **dm** соответствующие компоненты для таблиц «Товары» (компоненты TabTovar, dsTovar), «Клиенты» (компоненты TabKlient, dsKlient), «Сотрудники» (компоненты TabSotrydniki, dsSotrydniki), «Поставщик» (компоненты TabPostavshik, dsPostavshik), «Поставка» (компоненты TabPostavka, dsPostavka).

3.7 Сохраняем проект – File -> Save All.

База данных Товар.accdb подключена к клиентскому приложению, и все таблицы привязаны к компонентам Delphi.

Раздел 1. Проектирование ИС
Тема 1.6. Проектирование клиентской части АИС АИС (20 часов)

Практическая работа №11

«Клиентская часть: размещение визуальных компонентов, отображение таблиц».

Задачи обучающегося:
1. Научиться устанавливать и ознакомится с компонентами ADOConnection, ADOTable и DataSource для связи БД
Опорные понятия: подключение базы данных.
Планируемый результат:
Студент должен
Уметь создавать формы для ввода данных в клиентском приложении
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, IDE, ПК

1. Создаем новую форму File->New->Form-Delphi. Сохраняем модуль с именем **Unit2** (предлагается по умолчанию).
2 В инспекторе объектов для Unit2 устанавливаем свойство:
Caption = «Заказы» «Товары» «Клиенты»;
Visible = False // При запуске приложения – окно невидимо.
3 Знакомим Form1 и Form2.
3.1 Делаем активной Form1 (щелкнем по Unit1, а затем по форме). Далее **File->Use Unit..** (Использовать модуль), где в контекстном меню выбираем **Unit2** и жмем **OK** (рисунок 1).

Теперь форма Form1 знает о существовании Form2, в коде **Unit1** в разделе **implementation** прописалась строка `uses Unit1;` (проверьте это нажав F12).

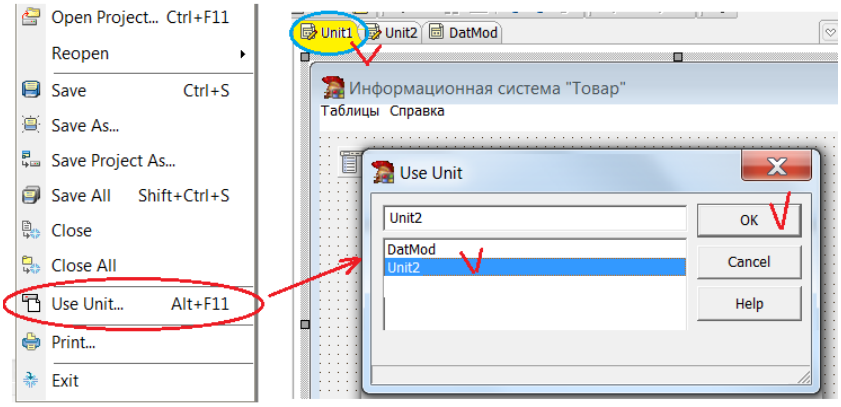


Рисунок 1 – Знакомство форм Form1 с Form2

3.2 Делаем активной Form2. Далее, **File->Use Unit..** и в контекстном меню выбираем **Unit1** и жмем **OK**. Форма Form2 знает о Form1 (проверьте).

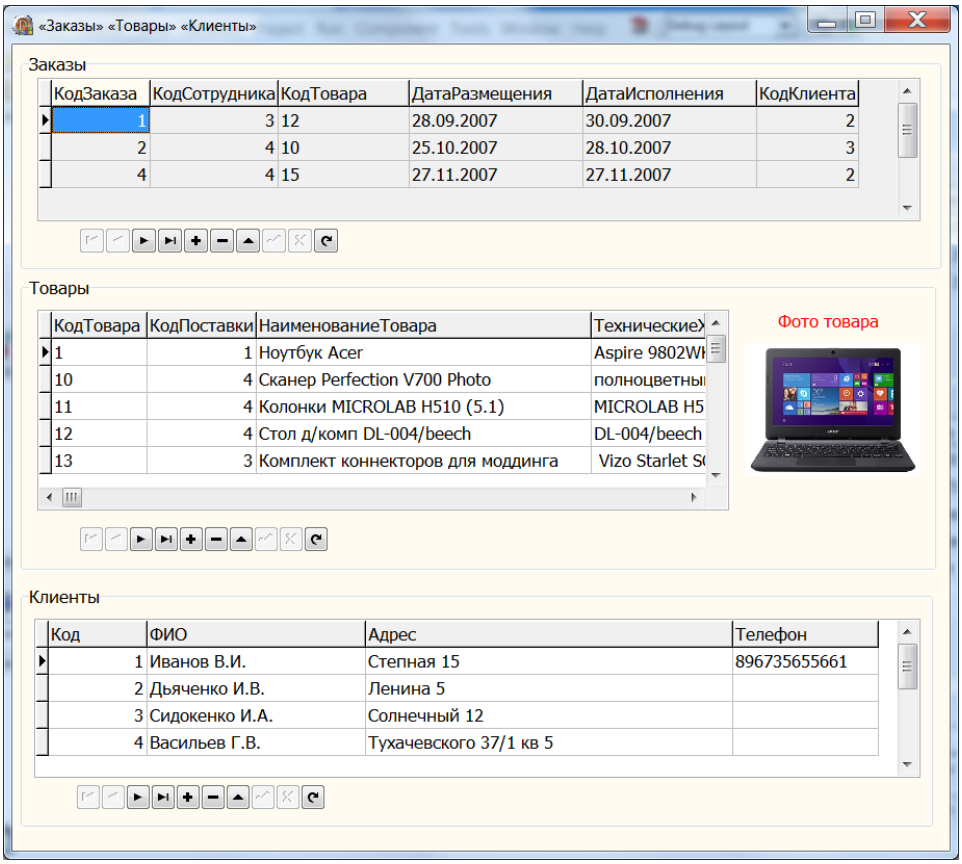


Рисунок 2 – Вид окна для пункта ЗаказыТоварыКлиенты

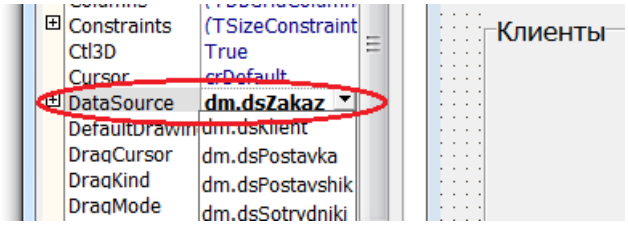
4 Переходим на Form1. Щелкаем дважды мышкой по компоненту **MainMenu1**, а затем дважды по пункту **ЗаказыТоварыКлиенты**. Пишем код.

```
procedure TForm1.N2Click(Sender: TObject);
begin
Form2.Show;    // Метод Show выводит на экран окно Form2
end;
```

5 **Оформляем Form2.** Окончательный вид окна показан на рисунке 2.

5.1 Знакомим **Form1** и **Form2** с модулем **DatMod** (см. 3.1 и 3.2).

5.1 Положим на **Form2** три компонента **GroupBox** (закладка Standard) и используя свойства **Caption** дайте им заголовки «Заказы» (для GroupBox1), «Товары» (для GroupBox2) и «Клиенты» (для GroupBox3).
5.2 Положите в **GroupBox1** сетку **DbGrid1** (закладка Data Controls). Для **DbGrid1** свяжите свойство **DataSource** с источником данных **dsZakaz**. В **Окне редактора полей** (двойной клик по DbGrid1) выберите команду **Add all fields**.



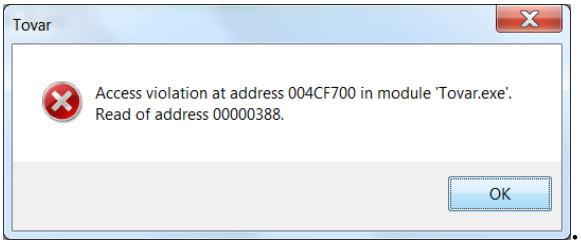
Проделайте ту же операцию для «Товары» и «Клиенты».
6 Сохраняем (File -> Save All) и компилируем (F9) проект.
7 Положите в **GroupBox1** под сетку **DbGrid1** компонент **DBNavigator** (закладка Data Controls). В инспекторе **DBNavigator** свойство **DataSource** = **dsZakaz**.
Проделайте ту же операцию для «Товары» и «Клиенты».
8 Вывод картинки. В бокс **GroupBox2** (Товары) положите компонент **Image** (вкладка Additional) для визуализации изображений товаров, а также компонент **Label** (вкладка Standart) для поясняющего текста.
Разрабатываемое приложение предполагает, что папка **foto** с изображениями товаров находятся в каталоге таблицы базы данных. Во время добавления информации в базу данных пользователь вводит в поле **Изображение** таблицы **Товары** имя файла (с расширением) картинки, а во время просмотра картинка автоматически появляется в поле **Image1**. Картинки могут иметь расширения bmp или jpeg.
9 Пишем код отображения фото. Переходим в модуль **dm (DatMod)**.
Заходим в Код (F12). Чтобы приложение понимало формат файлов *.jpg, добавьте в список USES (в блоке implementation) имя модуля **JPEG,** т.е. `implementation uses Unit2, jpeg;`. Для формата BMP – ничего делать не надо.
Возвращаемся к форме **dm** (F12). Выделяем компонент TabTovar, в Инспекторе переходим во вкладку **Events** и выбираем событие **AfterScroll** (двойной клик).

В окне кода для модуля dm (DatMod) пишем код

```
// Событие возникает при переходе к другой записи таблицы TabTovar (следующей,
// предыдущей, первой или последней), т.е. при скроллинге
procedure Tdm.TabTovarAfterScroll(DataSet: TDataSet);
var
  PicPath: string; // Для нахождения пути к фото
  PicName: string; // Для файла картинки
begin
 PicPath:=ExtractFilePath(ParamStr(0))+'foto\'; // Путь к папке foto
 if TabTovar.FieldValues['Изображение']<>'' then  // Поле 'Изображение' не пусто
  PicName:=TabTovar.FieldByName('Изображение').AsString  //Из поля 'Изображение'
  else  PicName:='fotonet.bmp';   // Если поле 'Изображение' пусто
 try  // Если не будет ошибки при загрузке фото (имя задано не правильно)
   Form2.Image1.Picture.LoadFromFile(PicPath+PicName); // Загружаем картинку
   Form2.Image1.Visible:=True;   // делаем Image1 видимой
 except   // Если ошибка при загрузке фото
   Form2.Image1.Picture.LoadFromFile(PicPath+'fotonet.bmp'); //Выводим Нет фото
   Form2.Image1.Visible:=True;
 end;  // Конец обработки исключительной ситуации try...except
end;   // Конец функции скроллирования таблицы
```

Краткое пояснение. Событие **AfterScrool** происходит всякий раз после перехода к другой записи таблицы. Процедура **Tdm.TabTovarAfterScroll** анализирует содержимое поля **'Изображение'** таблицы **TabTovar** и, если оно не пустое, выводит картинку из папки **foto**. В переменной **PicPath** сохранится полный путь к папке с приложением, а в переменной **PicName** – содержимое ячейки поля **'Изображение'** в записи с курсором. Функция **ExtractFilePath(ParamStr(0))** возвращает путь к папке, где запущено приложение, возвращает тип **string**. Методы **TabTovar.FieldValues['Изображение']** и **TabTovar.FieldByName('Изображение').AsString** – два способа извлечь содержимое из ячейки текущей записи поля 'Изображение'. Метод **LoadFromFile(PicPath+PicName)** загружает картинку из файла.

10 Сохраняем (File -> Save All) и компилируем (F9) проект. Если при этом может выскочить сообщение о том, что в приложении есть обращение на несуществующий объект:



Ничего страшного. Просто мы сначала создали модуль **dm** (DatMod), а потом **Form2** (Unit2), и это прописалось в модуле **Tovar.pas**. Поэтому при запуске приложения создается модуль **dm**, при этом процедура Tdm.TabTovarAfterScroll ссылается на форму Form2, которой еще нет. Чтобы обойти эту ситуацию откройте файл **Tovar.dpr** и поменяйте местами соответствующие строчки:

```
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(Tdm, dm);
  Application.CreateForm(TForm2, Form2);
  Application.Run;
end.
```

Замена

```
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(Tdm, dm);
  Application.Run;
end.
```

11 Сохраняем (File -> Save All) и компилируем (F9) проект.

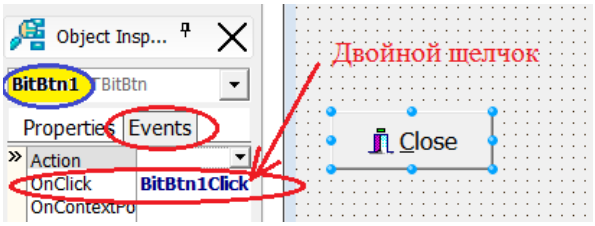
Задание. Создайте форму Form3 и проведите разработку окна для работы с пунктом меню «СотрудникиПоставщикПоставка». Не забудьте в выбранных таблицах (модуль **dm**) установить свойство **Active = True**.

Закрытие главной формы

Чтобы внесенные пользователем изменения в таблицы были сохранены в БД «Товар» в Access, таблицы в Delphi по окончании работы необходимо закрыть. Для этого используем метод **Close**. Для закрытия главной формы (и приложения в целом) используем метод **Free** – закрывает форму и освобождает память. Отметим, что если главную форму закрыть, то рабочие формы закрываются автоматически.

1 Переходим на **Form1**. Устанавливаем кнопку **BitBtn1** (вкладка Additional). В инспекторе **BitBtn1 у**станавливаем свойство **Kind = bkClose**.

2 В инспекторе **BitBtn1** заходим в Events и вызываем обработчик события **OnClick** (Двойным щелчком).



В открывшемся окне **Code** пишем команду на закрытие **Form1**.

```
// Нажатие на кнопку Close (BitBtn1)
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
 Form1.Free; // Закрыть Form1. Освободить память.
end;
```

Пояснение. При попытке закрытия приложения (команда **Form1.Free**) автоматически возникает событие **OnCloseQuery**. Обработчик события автоматически получает логическую переменную **CanClose**. По умолчанию эта переменная имеет значение **True**, и форму можно закрыть. Если параметр **CanClose** в значении False, то закрытие формы отменяется. Такую возможность используем для подтверждения закрытия главной формы **Form1** с помощью **MessageDlg()**. Здесь при закрытии Form1 выдается типичный запрос на подтверждение операции – Yes и No.

3 В инспекторе **Form1** заходим в **Events** и вызываем обработчик события **OnCloseQuery** (двойным щелчком).

В открывшемся окне **Code** пишем код обработчика события **OnCloseQuery** (запрос на закрытие).

```
// Обработчик события закрытия TForm1. Включается при закрытии формы
// любым из способов (кнопка Close, команда системного меню или Alt+F4 )
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
CanClose := MessageDlg('Закрыть приложение?', mtConfirmation,
        [mbYes, mbNo], 0) = mrYes;  // Подтверждение закрытия приложения
if CanClose=True then  begin
 dm.TabZakaz.Close; // Закрыть таблицу TabZakaz
 dm.TabKlient.Close;  // Закрыть таблицу TabKlient
 dm.TabTovar.Close;
 dm.TabSotrydniki.Close;   // Закрытие таблиц
 dm.TabPostavshik.Close;
 dm.TabPostavka.Close;
 Form2.Free;   // Закрыть Form2. Освободить память.
 Form3.Free;   // Закрыть Form3. Освободить память.
 end;
end;
```

По аналогии с построенным рамочным проектом «Информационная система Товар», созданной на основе БД Access и приложения Delphi, спроектируйте и реализуйте

информационную систему согласно Вашему варианту. В качестве базы данных используйте базу данных, разработанную ранее. Помните, любые новшества только приветствуются. Например, поле **Memo** можно вывести отдельно от записи в таблице.

Раздел 1. Проектирование ИС
Тема 1.6. Проектирование клиентской части АИС АИС (20 часов)

Практическая работа №12
«Запросы на добавление данных».

Задачи обучающегося:
1. Получить навыки формирования SQL запросов на добавление, изменение, извлечение и удаление данных

Опорные понятия: SQL запросы.
Планируемый результат:
Студент должен
Уметь создавать запросы на добавление данных
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

1. Создать базу данных используя мастер создания БД в SQL Server Management Studio согласно схеме, представленной на рисунке 1
2. Написать SQL запросы на добавление данных в таблицы. Данные представлены на рисунках 2 – 5.
3. Изучить 15 примеров простых и вложенных запросов на извлечение данных.
4. На свое усмотрения создать три триггера из примеров
5. Протестировать их и результаты теста привести в отчете.
6. Для своей схемы БД написать 25 запросов различной степени сложности (аналогично проделанным выше примерам).
7. Результаты выполнения представить в отчете.

Перед тем как приступить к выполнению лабораторной работы вам необходимо создать базу данных используя мастер создания БД согласно ниже представленной схеме

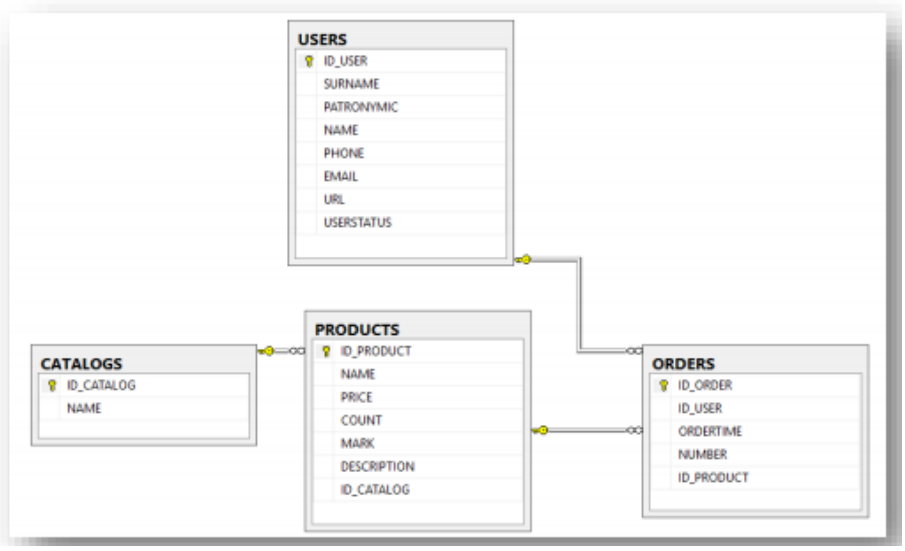


Рис. 1 - Схема базы данных

Следующим шагом будет заполнение БД данными. Ниже представлены данные, которыми необходимо наполнить базу данных используя SQL запросы

	ID_CATALOG	NAME
1	1	Процессоры
2	2	Материнские платы
3	3	Видеоадаптеры
4	4	Жесткие диски
5	5	Оперативная память

Рис. 2 - Данные в таблице CATALOGS

	ID_PRODUCT	NAME	PRICE	COUNT	MARK	DESCRIPTION	ID_CATALOG
1	1	Celeron 1.8	1595.00	10	3,6	Процессор Celeron® 1.8GHz, 128kb, 478-PGA, 400Mhz, OEM 0.18	1
2	2	Celeron 2.0GHz	1969.00	2	3,7	Процессор Celeron® 2.0GHz, 128KB, 478-PGA, 400MHz, OEM	1
3	3	Celeron 2.4GHz	2109.00	4	3,9	Процессор Celeron® 2.4GHz, 128kb, 478-PGA, 400Mhz, OEM	1
4	4	Celeron D 320 2.4GHz	1962.00	1	4,1	Процессор Celeron® D 320 2.4GHz, 256kb, 478-PGA, 533Mhz, OEM	1
5	5	Celeron D 325 2.53GHz	2747.00	6	4,1	Процессор Celeron® D 325 2.53GHz, 256kb, 478-PGA, 533Mhz, OEM	1
6	6	Celeron D 315 2.26GHz	1880.00	6	4,1	Процессор Celeron® D 315 2.26GHz, 256kb, 478-PGA, 533Mhz, OEM	1
7	7	Intel Pentium 4 3.2GHz	7259.00	5	4,5	Процессор Intel® Pentium®4 3.2GHz, 1Mb, 478-PGA, 800Mhz, Hyper-Thre...	1
8	8	Intel Pentium 4 3.0GHz	6147.00	1	4,6	Процессор Intel® Pentium®4 3.0GHz, 512Kb, 478-PGA, 800Mhz, Hyper-Thr...	1
9	9	Intel Pentium 4 3.0GHz	5673.00	12	4,5	Процессор Intel® Pentium®4 3.0GHz, 1Mb, 478-PGA, 800Mhz, Hyper-Trea...	1
10	10	Gigabyte GA-8I848P-RS	1896.00	4	3,9	Материнская плата SOCKET-478 Gigabyte GA-8I848P-RS i848, (800Mhz), ...	2
11	11	Gigabyte GA-8IG1000	2420.00	2	3,8	Материнская плата SOCKET-478 Gigabyte GA-8IG1000 i865g,FSB800/53...	2
12	12	Gigabyte GA-8IPE1000G	2289.00	6	3,7	Материнская плата Socket-478 Gigabyte GA-8IPE1000G i865PE(800/533...	2
13	13	Asustek P4C800-E Delux	5395.00	4	4,1	Материнская плата Socket-478 Asustek P4C800-E Delux i875P,FSB800/5...	2
14	14	Asustek P4P800-VM\\L i865G	2518.00	6	4	Материнская плата Socket-478 Asustek P4P800-VM\\L i865G FSB800/53...	2
15	15	Epox EP-4PDA3I	2289.00	5	4	Материнская плата Socket-478 Epox EP-4PDA3I i865PE(800Mhz), 2chDD...	2
16	16	ASUSTEK A9600XT/TD	5156.00	2	4,7	Видеоадаптер ASUSTEK A9600XT/TD 128Mb DDR SDRAM, 2x400MHz ...	3
17	17	ASUSTEK V9520X	1602.00	6	4	Видеоадаптер ASUSTEK V9520X 128Mb DDR SDRAM, 400MHz DAC, AG...	3
18	18	SAPPHIRE 256MB RADEON 9550	2730.00	3	3,8	ВИДЕОКАРТА SAPPHIRE 256MB RADEON 9550, TV-out, DVI, OEM	3
19	19	GIGABYTE AGP GV-N59X128D	5886.00	6	3,6	ВИДЕОКАРТА GIGABYTE AGP GV-N59X128D FX5900XT OEM	3
20	20	Maxtor 6Y120P0	2456.00	6	4,5	Винчестер 120 GB Maxtor 6Y120P0, UDMA-133, 7200rpm, 8MB	4
21	21	Maxtor 6B200P0	3589.00	4	4	Винчестер 200 GB Maxtor 6B200P0, UDMA-133, 7200rpm, 8Mb	4
22	22	Samsung SP0812C	2093.00	5	4	Винчестер 80 GB Samsung SP0812C, SATA, 7200rpm SpinPoint P80 Serial...	4
23	23	Seagate Barracuda ST3160023A	3139.00	3	4,1	Винчестер 160 GB Seagate Barracuda ST3160023A, UDMA-100, 7200rpm,...	4
24	24	Seagate ST3120026A	2468.00	8	4,2	Винчестер 120 GB Seagate ST3120026A, UDMA-100, 7200rpm, 8MB	4
25	25	DDR-400 256MB Kingston	1085.00	20	4,8	Оперативная память DDR-400 256MB Kingston	5
26	26	DDR-400 256MB Hynix Original	1179.00	15	4,6	Оперативная память DDR-400 256MB Hynix Original	5
27	27	DDR-400 256MB PQI	899.00	10	4,2	Оперативная память DDR-400 256MB PQI	5
28	28	DDR-400 512MB Kingston	1932.00	20	4,8	Оперативная память DDR-400 512MB Kingston	5
29	29	DDR-400 512MB PQI	1690.00	12	4,2	Оперативная память DDR-400 512MB PQI	5
30	30	DDR-400 512MB Hynix	1717.00	8	4,5	Оперативная память DDR-400 512MB Hynix	5

Рис. 3 – Данные в таблице PRODUCTS

ID_USER	SURNAME	PATRONYMIC	NAME	PHONE	EMAIL	URL	USERSTATUS
1	Иванов	Валерьевич	Александр	58-98-78	ivanov@email.ru	NULL	active
2	Прокопчук	Иванович	Сергей	9057777777	pro@email.ru	NULL	passive
3	Семенов	Вячеславович	Игорь	9056666100	simdyanov@softtime.ru	http://www.softtime.ru	active
4	Петров	Валерьевич	Максим	NULL	kuznetsov@softtime.ru	http://www.softtime.ru	active
5	Посев	Юрьевич	Анатолий	NULL	losev@email.ru	NULL	lock
6	Корнеев	Александрович	Александр	89-78-36	korneev@domen.ru	NULL	gold

Рис. 4 – Данные в таблице USERS

ID_ORDER	ID_USER	ORDERTIME	NUMBER	ID_PRODUCT
4	3	2005-04-01 10:39:38.000	1	8
5	6	2005-10-02 09:40:29.000	2	10
6	1	2005-02-18 13:41:05.000	4	20
7	3	2005-10-03 18:20:00.000	1	20
8	3	2005-03-17 19:15:36.000	1	20

Рис. 5 – Данные в таблице ORDERS

Ниже представлен перечень (24) простых запросов к БД (схема которой описана выше. Данные примеры покрывают большой спектр конструкций языка SQL, начиная от простых запросов, кончая запросов с использованием функций и сортировок. Для выполнения лабораторной работы, Вам необходимо проделать все 24 запроса и привести результат выполнения в виде скриншота. Все конструкции языка SQL подробно описаны в лекционном материале

Пример 1. Вывод данных таблицы CATALOGS

```
SELECT ID_CATALOG, NAME FROM CATALOGS –- Здесь выводятся два поля
ID_CATALOG и NAME из таблицы CATALOGS
SELECT * FROM CATALOGS –- Здесь выводятся все поля из таблицы CATALOGS
```

Пример 2. Вывод данных таблицы CATALOGS с присвоением псевдонима

```
SELECT
    ID_CATALOG AS 'Идентификатор категории', -- команда AS изменяет имя
столбца в результирующей таблице на имя, указанное после этой
команды в кавычках (ID_CATALOG изменяется на 'Идентификатор категории')
    NAME AS 'Имя категории'
FROM CATALOGS
```

Пример 3. Добавление данных с помощью SELECT в результирующую таблицу

```
SELECT NAME, ID_CATALOG, 5, 'COMMENTS' FROM CATALOGS
```

Пример 4. Извлечение из таблицы CATALOGS записи, чей первичный ключ ID_CATALOG больше 2

```
SELECT * FROM CATALOGS
WHERE ID_CATALOG > 2 –- здесь WHERE это условие. Т.е. мы указываем, что нам
```

нужно выбрать все данные из всех столбцов (*) таблицы CATALOGS и вывести те строки, где значение поля ID_CATALOG будет больше 2

Пример 5. Составное условие: извлечение из таблицы CATALOGS записи, чей первичный ключ ID_CATALOG больше 2, но меньше или равен 4

```
SELECT * FROM CATALOGS
WHERE ID_CATALOG > 2 AND ID_CATALOG <= 4 –- AND позволяет создавать
```

составные условия, т.е. после WHERE мы можем указать несколько условий для выборки, разделяя их AND

SELECT * FROM CATALOGS
WHERE ID_CATALOG BETWEEN 3 AND 4 -- Здесь записана упрощенная версия запроса выше. BETWEEN буквально означает «между», т.е. весь запрос можно воспринять так: Выбрать все столбцы из таблицы CATALOGS, и вывести те строки, где значение поля ID_CATALOG находится между 3 и 4

Пример 6. Вывод записей, удовлетворяющих не диапазону, а списку

SELECT * FROM CATALOGS
WHERE ID_CATALOG IN (1, 2, 5) –- IN указывает, что необходимо в результирующую таблицу поместить только те строки, значения поля ID_CATALOG которых равны значениям указанным в скобках после IN

Пример 7. Вывод записей, удовлетворяющих условию, заданному текстом: вывести все записи, содержащие слово процессор

SELECT * FROM CATALOGS
WHERE NAME = 'процессоры' –- выводятся все строки, значения поля NAME в которых равно «процессоры».

Пример 8. Вывод записей, удовлетворяющих условию, заданному текстом: вывести все записи, не содержащие слово процессор

SELECT * FROM CATALOGS
WHERE NOT NAME = 'процессоры' –- то же, что и в предыдущем примере, только выводится все, кроме «процессоры»

Пример 9. Вывод записей, удовлетворяющих условию, заданному частью текста

SELECT * FROM USERS
WHERE SURNAME LIKE 'И%' –- LIKE дает возможность указать какую-либо часть значения. В данном случае мы получаем все строки, значения поля SURNAME которых будет начинаться с буквы И

Пример 10. Работа с датой: извлечение из таблицы ORDERS записи, соответствующие сделкам, осуществленным за февраль 2005 г

SELECT * FROM orders WHERE ORDERTIME >= '2005-02-01' AND ORDERTIME < '2005-03-01'

Пример 11. Сортировка по значению одного из столбцов

SELECT * FROM CATALOGS
ORDER BY ID_CATALOG –- сортируем все строки по полю ID_CATALOG (по возрастанию)

SELECT * FROM CATALOGS
ORDER BY NAME –- сортируем все строки по полю NAME

Пример 12. Извлечение из таблицы PRODUCTS записи товаров, количество которых COUNT на складе от 4 до 8 с сортировкой по полю COUNT и полю MARK (для краткости выведем только столбцы COUNT и MARK)

SELECT COUNT, MARK FROM PRODUCTS
WHERE COUNT BETWEEN 4 AND 8 ORDER BY COUNT, MARK –- выбираем строки, значение поля COUNT которого находятся между 4 и 8, затем сортируем полученные строки сначала по COUNT затем по MARK

Пример 13. Изменение порядка сортировки (по умолчанию, сортировка производится в прямом порядке (ASC))

SELECT ORDERTIME FROM ORDERS
ORDER BY ORDERTIME DESC -- DESC означает сортировка по убыванию

Пример 14. Извлечение первых пяти записей с обратной сортировкой по полю COUNT

SELECT TOP 5 ID_PRODUCT, COUNT FROM PRODUCTS
ORDER BY COUNT DESC

Пример 15. Подсчет количества проданных ТОВАРОВ

SELECT SUM (NUMBER) AS 'Всего продано' -- SUM означает сумму значений поля NUMBER
FROM ORDERS

Раздел 1. Проектирование ИС
Тема 1.6. Проектирование клиентской части АИС АИС (20 часов)

Практическая работа №13
«Запросы на редактирование и удаление данных».

Задачи обучающегося:
1. Получить навыки редактирования данных в MS Acces
Опорные понятия: запросы Acces.
Планируемый результат:
Студент должен
Уметь редактировать данные в MS Acces
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

Теоретическая часть

Запросы являются мощным инструментом работы с данными, которые позволяют извлекать данные, автоматически обновлять или удалять записей из одной или нескольких таблиц, а также выполнять вычисления, используя значения, хранимые в таблице. С помощью запросов можно создать новую таблицу по результатам запроса или скопировать данные из одной таблицы в другую. Запросы позволяют реализовать сложные операции манипулирования данными, например, удалить сразу несколько записей, удовлетворяющих определенному условию, или изменить значение поля в нескольких записях.

Перечислим основные преимущества запросов:

запросы позволяют собирать воедино информацию из нескольких таблиц, учитывая связи, установленные между таблицами в БД;

при разработке запроса можно выбрать, какие поля исходных таблиц и в какой последовательности будут включены в таблицу результатов;

запросы позволяют выполнять вычисления, основываясь на значениях полей таблицы.

СУБД Access позволяет создавать запросы различных типов:

запрос на выборку данных,
запрос на обновление данных,
запрос на удаление данных,
запрос на добавление данных,
запрос на создание новой таблицы,
перекрестный запрос.

Запрос на выборку является наиболее часто используемым типом запросов. Он позволяет осуществить выборку данных, соответствующих заданным условиям отбора, из одной или нескольких таблиц. В результате его выполнения будет сформирован набор записей, удовлетворяющих заданному условию. Условие отбора может быть достаточно сложным. Для его задания используются арифметические операции, логические операции, операции отношения и специальные функции, предусмотренные в СУБД. Кроме того, применяя при использовании запроса на выборку групповые операции, можно группировать данные или, например, вычислять суммы, средние значения или количество записей, удовлетворяющих критерию отбора.

Запрос на обновление данных позволяет изменить данные для группы записей одной или нескольких таблиц.

Запрос на добавление данных позволяет добавить данные из одной таблицы в другую.

Запрос на удаление данных позволяет удалить из одной или нескольких таблиц записи, удовлетворяющие определенному критерию.

Запрос на создание таблицы позволяет сформировать новую таблицу. При этом записи результирующего набора становятся основой для новой таблицы, структуру которой определяет структура самого запроса. Как правило, такие запросы используются для создания таблицы при экспорте данных в другие БД Microsoft Access или для резервного копирования существующих таблиц.

Результатом перекрестного запроса является набор записей, представленных в специальном формате, напоминающем электронную таблицу. Данные в такой таблице группируются по двум наборам данных: первый выводится в столбце слева

СУБД Access предоставляет пользователю возможность разрабатывать запросы с параметрами. Они позволяют задавать один или несколько параметров для условия отбора. Прежде чем запрос будет выполнен, на экран выводится диалоговое окно с приглашением ввести один или несколько параметров. Таким образом, запрос обеспечивает возможность уточнения параметров отбора при каждом своем запуске.

Создать запрос можно с помощью Мастера запросов или Конструктора запросов. Мастер запросов позволяет реализовать только самые простые запросы на выборку, в которых нельзя определить условия отбора записей. Главное преимущество Мастера запросов заключается в том, что с его помощью можно выбрать только те поля, которые вам необходимы. Самый мощный инструментарий, позволяющий сформулировать свой вопрос к БД − это Конструктор запросов. Он позволяет создавать запросы любой сложности, изменить запросы, построенные с помощью Мастера запросов.

Задание:

Исходя из выбранной темы, сделайте запросы на редактирование и удаления данных из БД

Раздел 1. Проектирование ИС
Тема 1.6. Проектирование клиентской части АИС АИС (20 часов)
Практическая работа №14

«Сортировка, поиск, фильтрация данных: в базе данных и выборках».

Задачи обучающегося:

1. Получить навыки сортировки, поиска, фильтрации данных в MS Acces

Опорные понятия: запросы Acces.

Планируемый результат:

Студент должен

Уметь сортировать данные в MS Acces

Пользоваться инструментальными средствами программного обеспечения

Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

Теоретическая часть:

Сортировка − упорядочение данных по какому-либо признаку. Для сортировки и поиска (фильтрации) информации в Access 2007 предусмотрен целый блок команд *Сортировка и фильтр* на карточке *Главная* (см. рис. 1).

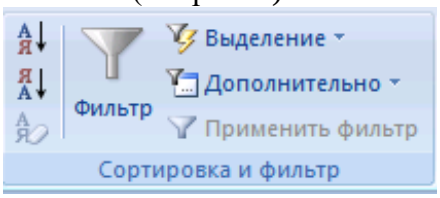


Рис. 1

При сортировке все строки таблицы перестраиваются в указанном порядке. Сортировка позволяет упорядочить данные любого типа: числа (в порядке возрастания), текст (по алфавиту), даты (в порядке возрастания года в дате, при одинаковых годах в порядке возрастания месяца).

Задание 1. Выполните следующие виды сортировок:

- Сортировка списка сотрудников по фамилиям в алфавитном порядке.
- Сортировка списка сотрудников по датам рождения в порядке убывания возраста.
- Сортировка списка сотрудников по ключевому полю в порядке возрастания.

Последовательность работы:

1. Откройте таблицу Сотрудники.
2. Выделите поле сортировки (Фамилия) щелчком на названии поля: при этом выделяется весь столбец с заголовком.
3. Щелкните на кнопке *Сортировка* по возрастанию. Просмотрите результаты сортировки: все фамилии расположены в алфавитном порядке.
4. Проведите другие виды сортировки, указанные в задании.

Задание 2. Поиск с использованием фильтра «*Выделение*».

Поиск (фильтрация) − выбор данных, удовлетворяющих некоторому условию. Выбор из базы данных тех записей, которые удовлетворяют требованиям пользователя, осуществляется с помощью фильтров - условий, по которым производится поиск и отбор записей.

Одним из самых простых способов отбора записей является использование фильтра «*Выделение*».

Порядок работы:

1. Откройте таблицу с данными о сотрудниках.
2. В какой-нибудь записи выделите значение одного из полей или его часть, например, первую букву фамилии Белкина.
3. После применения фильтра в таблице останутся только записи, удовлетворяющие выбранному условию. К уже отобранным записям можно вновь применить другой фильтр. Тогда останутся только записи, удовлетворяющие двум последовательно примененным критериям отбора.
4. Чтобы просмотреть все записи, надо нажать на кнопку *Применить фильтр*,

которая включает и отключает фильтрацию. Среда баз данных помнит последний установленный фильтр.

Фильтр можно задать также в форме или запросе. Технология работы аналогична приведенной выше. Проведите в таблице Сотрудники отбор записей, удовлетворяющих следующим условиям:

- Фамилия сотрудника начинается на букву «Б».
- День рождения сотрудника в декабре.
- Сотрудники, работающие в подразделении Дирекция.
- Сотрудники, имеющие должность «менеджер».
- Менеджеры, работающие в отделе снабжения.

Использование простого фильтра − другая возможность отбора данных. Простой фильтр позволяет задать сразу несколько критериев отбора по разным полям.

Последовательность работы:

1. Откройте таблицу с данными о сотрудниках**.**
2. Выберите команду *Фильтр*. В зависимости от положения курсора (типа поля, текстовое или числовое) появятся различные варианты, см. рис.2.

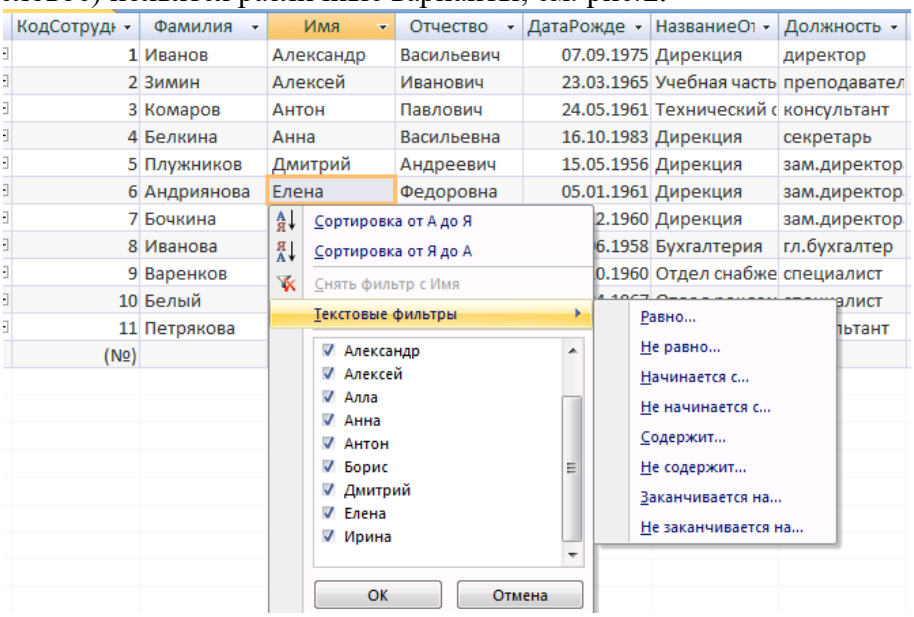


Рис.2

3. Выберите критерии отбора.
4. Дальнейшие действия аналогичны применению фильтра «*По выделенному*»**.**
5. Выполните фильтрацию, используя простой фильтр, по критериям задания 2.

Раздел 1. Проектирование ИС
Тема 1.6. Проектирование клиентской части АИС АИС (20 часов)

Практическая работа №15
«Работа с отчетами».

Задачи обучающегося:
1. Получить навыки работы с отчетами в MS Acces
Опорные понятия: запросы Acces.
Планируемый результат:
Студент должен
Уметь формировать отчеты в MS Acces
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

1. В главном меню базы данных на карточке *Создание* выберите блок команд *Отчеты→Мастер отчетов.*

2. На первом шаге появившегося диалога выберите из таблицы Структура предприятия поле Название отдела, из запроса Вычисляемые поля − поле ФИО, а из таблицы Сотрудники − поле Должность.

3. На втором шаге выберите группировку данных по названиям отделов. Выделите название отдела и нажмите стрелку повышения уровня (см. рис.1). Группировка позволяет вывести названия отделов в виде оформленных подзаголовков.

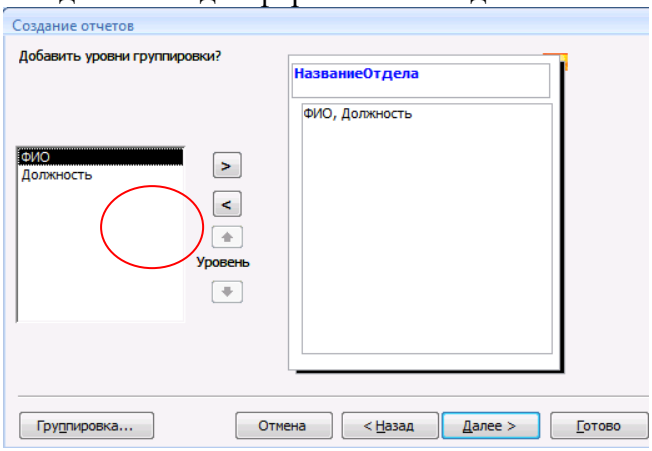


Рис.1

4. На третьем шаге задайте сортировку по полю ФИО для того, чтобы фамилии в отчете были расположены в алфавитном порядке.

5. На четвертом шаге выберите вид макета для отчета, например, ступенчатый.

6. На пятом шаге выберите стиль оформления.

7. На шестом шаге введите заголовок отчета: Список сотрудников, и завершите работу мастера, щелкнув на поле *Готово*. Отчет имеет приблизительно такой вид (см. рис. 2):

Рис. 2

8. После завершения работы мастера включается Режим *предварительного просмотра отчета*.

9. При просмотре некоторых отчетов можно заметить некоторые недочеты, которые можно исправить следующим образом: вставить пробелы, записать полностью Фамилия, Имя, Отчество, убрать рамки у названий отделов, выделить некоторые надписи жирным шрифтом.

10. Для исправления отчета перейдите в режим конструктора. В меню *Вид* выберите *Панель элементов*, выделите кнопку *Надпись* и нарисуйте небольшую рамку для надписи в области заголовка. В рамке напишите «По состоянию на …». Перенесите из

области нижнего колонтитула в область заголовка объект с функцией *Now()*, который выводит в отчет текущую дату (выделите его и следите за тем, чтобы двунаправленная стрелка находилась на нижней границе, а не на темном уголке, иначе он не перейдет в другую область) (см. рис.3).

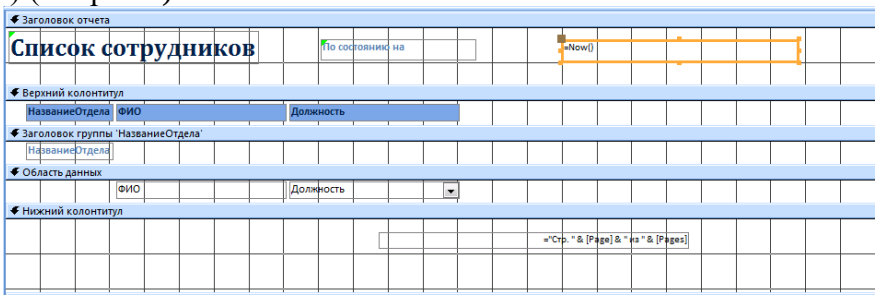
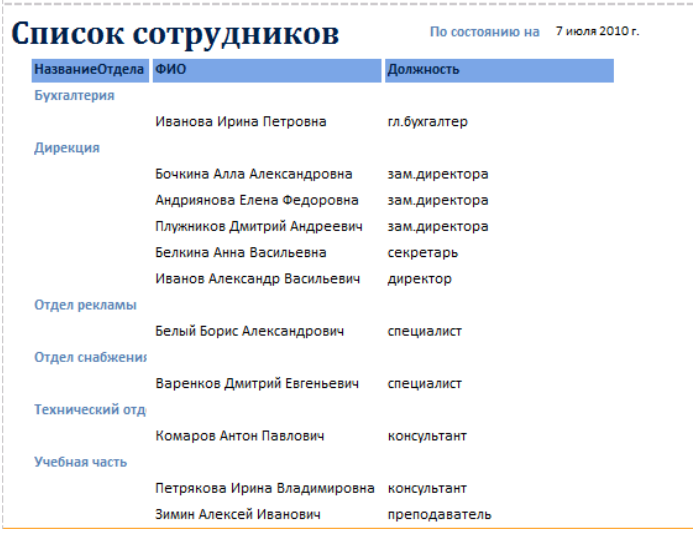


Рис.3. Отчет в режиме *Конструктора*



Список сотрудников По состоянию на 7 июля 2010 г.

НазваниеОтдела	ФИО	Должность
Бухгалтерия		
	Иванова Ирина Петровна	гл.бухгалтер
Дирекция		
	Бочкина Алла Александровна	зам.директора
	Андриянова Елена Федоровна	зам.директора
	Плужников Дмитрий Андреевич	зам.директора
	Белкина Анна Васильевна	секретарь
	Иванов Александр Васильевич	директор
Отдел рекламы		
	Белый Борис Александрович	специалист
Отдел снабжени		
	Варенков Дмитрий Евгеньевич	специалист
Технический отд		
	Комаров Антон Павлович	консультант
Учебная часть		
	Петрякова Ирина Владимировна	консультант
	Зимин Алексей Иванович	преподаватель

Рис.4. Готовый отчет

Задание 2. Отчет «Дни Рождения»

В этом отчете сформируем список сотрудников и их дни рождения, расположенные в порядке следования в календарном году.

Порядок работы:

1. Добавьте в запрос «Вычисляемые поля» поле Дата Рождения, изменения сохраните и закройте запрос.
2. Запустите *Мастер отчетов* и включите в отчет поля из запроса «Вычисляемые поля»: ФИО, Возраст, Месяц, День, Дата Рождения.
3. Удалите из получившегося в результате работы *Мастера* макета надписи и поля Месяц и День (см. рис.5).

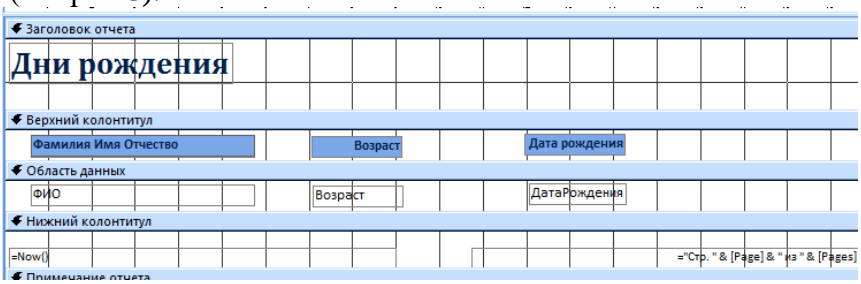


Рис.5

4. Исправьте надписи в верхнем колонтитуле: ФИО на Фамилия Имя Отчество, Дата Рождения на Дата рождения. Изменяйте только надписи, а не сами поля. Примерный вид отчета – на рисунке 6.

Дни рождения

Фамилия Имя Отчество	Возраст	Дата рождения
Иванов Александр Васильевич	35	07.09.1975
Зимин Алексей Иванович	45	23.03.1965
Комаров Антон Павлович	49	24.05.1961
Белкина Анна Васильевна	27	16.10.1983
Плужников Дмитрий Андреевич	54	15.05.1956
Андриянова Елена Федоровна	49	05.01.1961
Бочкина Алла Александровна	50	23.12.1960
Иванова Ирина Петровна	52	30.06.1958
Варенков Дмитрий Евгеньевич	50	09.10.1960
Белый Борис Александрович	43	26.04.1967
Петрякова Ирина Владимировна	35	05.12.1975

Рис. 6. Отчет «Дни рождения»

Задание 3. Отчет «Представительский»

В этом отчете создается единая форма нагрудной представительской карточки для сотрудников фирмы. Отчет содержит эмблему предприятия, фамилию, имя, отчество сотрудника, название отдела и должность.

Последовательность работы:

1. Запустите *Мастер отчетов* и включите в него поля Имя, Отчество, Фамилия, Должность из таблицы Сотрудники и поле Название Отдела из таблицы Структура предприятия.

3. Выберите вид макета - «*в столбец*», стиль оформления - *Обычный*.

4. После создания отчета при помощи мастера перейдите в режим *Конструктора* для корректировки макета отчета.

5. Удалите объекты из области заголовка отчета и нижнего колонтитула. Для этого щелчком выделите объект и нажмите *Del.* Следите за тем, чтобы не удалить и поля вместе с надписями. Чтобы этого не случилось, выберите меню *Упорядочить*, выделите прямоугольники-надписи и удалите макет оформления (рис. 62), нажав кнопку *Удалить*. Проделайте все это для каждой надписи. Все это для того, чтобы каждый прямоугольник можно было изменять и удалять по отдельности.

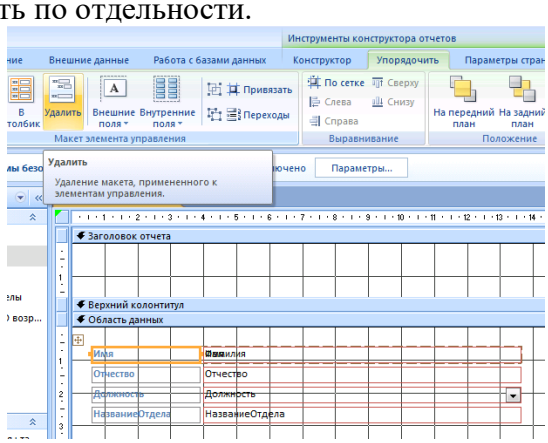


Рис.7

6. Удалите надписи полей из области данных.

7. Уменьшите до нуля высоту всех областей отчета, кроме области данных.

8. Измените размеры области данных до размеров нагрудной карточки 9х5,5 см. Для этого наведите курсор на правую (или нижнюю) границу области до появления двусторонней стрелки, захватите и переместите границу.

9. Добавьте рисунок эмблемы (если есть).

10. Обведите визитку рамкой, выбрав инструмент *Прямоугольник* на панели элементов. Измените формат рамки: вид, толщину границы, цвет границы и внутреннего заполнения. Просмотрите результат (данные могут не совпадать, см. рис.8).

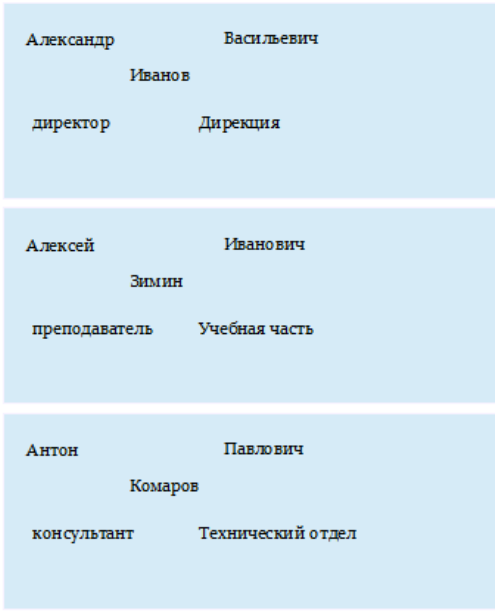


Рис. 8. Отчет «Представительский»

Раздел 1. Проектирование ИС
Тема 1.6. Проектирование клиентской части АИС АИС (20 часов)

Практическая работа №16
«Копирование клиентской части».

Задачи обучающегося:
1. Ознакомиться с основными конструкциями SQL, технологиями среды MS SQL Server Management, объектами SMO (среды MS Visual Studio) для резервного копирования и восстановления БД

Опорные понятия: запросы Acces.

Планируемый результат:
Студент должен
Уметь создавать резервные копии
Пользоваться инструментальными средствами программного обеспечения

Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

Задание №1. необходимо создать резервные копии базы данных «MMM» с использованием полного резервного копирования, разностного резервного копирования и резервного копирования журнала транзакций.

Ход работы:

1. Запустите SQL Server Management Studio (SSMS), подключитесь к своему экземпляру SQL Server, используя технологию 1.
2. Создайте папку с именем c:\Student\ВашаПапка\test.
3. Откройте окно нового запроса. Измените контекст на базу данных master, используя технологию 6. Наберите и исполните следующую команду, чтобы создать полную резервную копию базы данных:

BACKUP DATABASE MMM TO DISK = 'C:\.....TEST\AW.BAK’

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

4. Внесите изменение в таблицу «Модель» базы данных MMM. Добавьте одну запись (придумайте сами)/
5. Откройте окно нового запроса наберите и исполните следующую команду, чтобы создать резервную копию журнала транзакций и сохранить только что внесенное изменение:

BACKUP LOG MMM TO DISK = ‘C:\.....TEST\AW1.TRN'

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

6. Внесите еще одно изменение в таблицу«Модель».
7. Откройте окно нового запроса наберите и исполните следующую команду, чтобы создать разностную резервную копию базы данных:

BACKUP DATABASE MMM TO DISK = ‘C:\.....\TEST\AWDIFF1.BAK' WITH DIFFERENTIAL

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

8. Внесите еще одно изменение в таблицу«Модель».
9. Откройте окно нового запроса наберите и исполните следующую команду, чтобы создать полную резервную копию базы данных в указанном месте на диске:

BACKUP LOG MMM TO DISK = ‘C:\....TEST\AW2.TRN'

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

Задание №2. необходимо провести восстановление базы данных «MMM» из сделанных в задании №1 резервных копий.

Ход работы:

1. Если необходимо, запустите SSMS, подключитесь к своему экземпляру SQL Server, используя технологию 1.
2. Выполните восстановление БД из первой полной резервной копии (C:\...TEST\AW.BAK) средствами оболочки SSMS. Для этого выполните:
   - В обозревателе объектов вызовите контекстное меню на вашей БД и выберите задачу восстановления базы данных (см. рисунок 1).

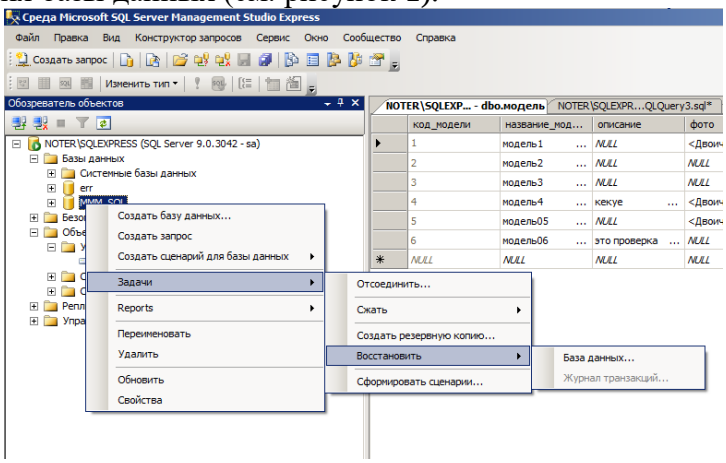


Рисунок 1 – Восстановление БД

   - В открывшемся окне необходимо задать следующие параметры восстановления

На закладке «Общие» необходимо выбрать:

a. Базу данных для восстановления (вашу MMM)
b. Выбрать источник набора данных для восстановления с устройства → файл C:\...TEST\AW.BAK
c. После определения файла-источника данных необходимо флажком выбрать базу данных для восстановления (рисунок 7).

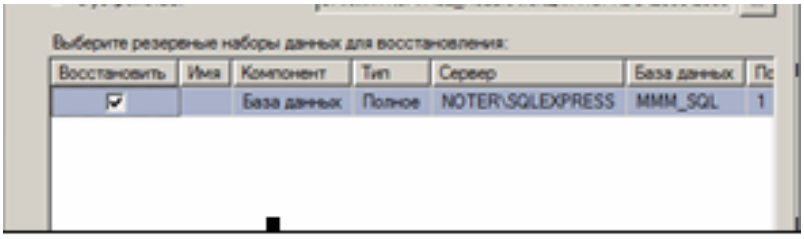


Рисунок 2- Выбор БД для восстановления

На закладке «Параметры»

a. необходимо включить опцию «Перезаписать БД» и «оставить БД готовой к использованию», (рисунок 3).

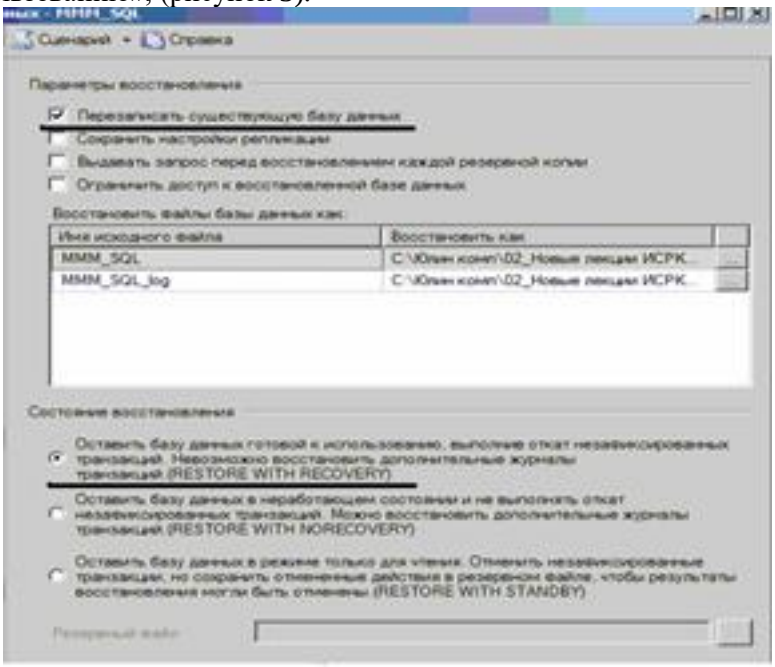


Рисунок 3 – Задание параметров восстановления

3. Нажмите ОК
4. После восстановления БД, откройте таблицу «Модель» и убедитесь, что она не содержит всех добавлений, вносимых вами в процессе выполнения упражнения, так как восстановление происходило из первой резервной копии (без изменений).

Задание №3. необходимо организовывать со стороны клиентского приложения, созданного в Visual Studio удаленное администрирование БД (резервное копирование).

Ход работы:

В Visual Studio

1. Создайте новый проект Windows Application и сохраните его в своей папке под именем Лабы_МММ_2 семестр.
2. В главную форму добавьте меню, изображенное на рисунке 9:

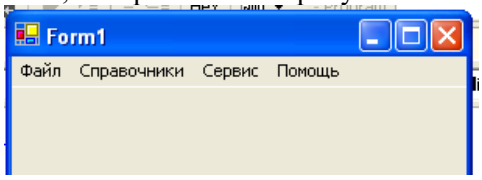


Рисунок 9 – Главное меню проекта

Файл (Открыть, Закрыть, Выход)
Справочники (Модель, Магазин, Дерево моделей)
Заказы (Работа с заказами)
Отчеты (Прайс-лист, Бланк заказов)
Администрирование БД (Резервное копирование, Безопасность)
Сервис (Калькулятор)
Помощь (Справка, О программе)

3. Добавьте новую форму в проект
4. Добавьте на только что созданную форму компоненты в соответствии с рисунком 4.

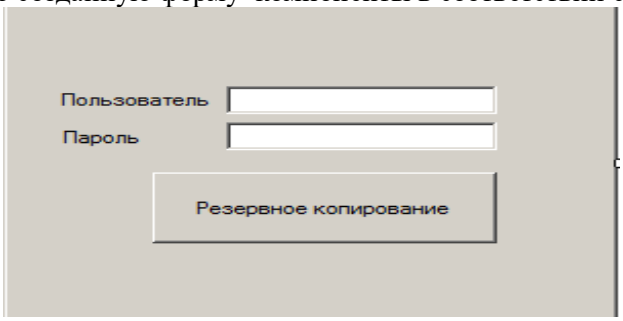


Рисунок 4 – Форма для подключения к серверу

5. Обеспечьте функциональную работу формы (напишите обработчик кнопки «Резервное копирование» с использованием объектов SMO. Описание объектов SMO, их свойств и методов см. в лекционном материале.)
6. Добавьте возможность открытия данной формы при выборе в главной форме пункта меню Администрироване БД → Резервное копирование
7. Запустите проект, проверьте работу формы.
8. Закройте проект
9. Убедитесь в появлении файла резервной копии на диске (файл, который указан в тексте программы).
10. Откройте SSMS. Добавьте в таблицу «Модель» новую строку данных (самостоятельно).
11. Средствами оболочки SSMS, выполните восстановление БД из резервной копии, созданной вашей программой
12. Убедитесь, что после восстановления добавленных строк в таблице «Модель» нет.

Задание №4. Ответьте на вопросы теста и представьте результаты преподавателю.

1. Вы выполняете разностное резервное копирование базы длимых AdveniurtWorks каждые четыре часа, начиная с 04:00. полная резервная копия создается в полночь. Какие данные будут содержаться в разностной резервной копии сделанной в полдень?
    a. A Страницы данных, измененные после полуночи.
    b. B. Экстенты, измененные после полуночи.
    c. C. Страницы данных, измененные после 08:00
    d. D. Экстенты. измененные после 08:00.
2. Вы выполняете полное резервное копирование базы данных Adventure Wortks,, которое завершается в полночь. Разностное резервное копирование выполняется по расписанию каждые четыре часа, начиная с 04:00. Резервное копирование журнала транзакций происходит по расписанию каждые пять минут. Какую информацию будет содержать резервная копия журнала транзакций, созданная в 09:15?
    a. A. Все транзакции, начатые после 09:10.
    b. B. Транзакции, завершенные после 09:10.
    c. C. Страницы, измененные после 09:10.
    d. D. Экстенты, измененные после 09:10.

Раздел 1. Проектирование ИС
Тема 1.7. Тестирование приложений АИС (4 часа)

Практическая работа №17
«Использование критериев оценки качества и надежности функционирования информационной системы».

Задачи обучающегося:

1. Научиться проводить оценку качества программного средства по различным показателям

Опорные понятия: оценка АИС.

Планируемый результат:

Студент должен
Уметь проводить оценку качества программного средства по различным показателям
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

Основные теоретические сведения

Все программы по характеру использования и категориям пользователей можно разделить на два класса - *утилитарные программы* и *программные продукты-(изделия)*.

Утилитарные программы («программы для себя») предназначены для удовлетворения нужд их разработчиков. Чаще всего утилитарные программы выполняют роль сервиса в технологии обработки данных либо являются программами решения функциональных задач, не предназначенных для широкого распространения.

Программные продукты (изделия) предназначены для удовлетворения потребностей пользователей, широкого распространения и продажи.

Существуют и другие варианты легального распространения программных продуктов:

- freeware - бесплатные программы, свободно распространяемые, поддерживаются самим пользователем, который правомочен вносить в них необходимые изменения;
- shareware - некоммерческие (условно-бесплатные) программы, которые могут использоваться, как правило, бесплатно.

Ряд производителей использует OEM - программы (Original Equipment Manufacturer), т.е. встроенные программы, устанавливаемые на компьютеры или поставляемые вместе с вычислительной техникой.

Программные продукты (НН) могут создаваться как:

- индивидуальная разработка под заказ;
- разработка для массового распространения среди пользователей.

Основными характеристиками программ являются:

- алгоритмическая сложность (логика алгоритмов обработки информации);
- состав и глубина проработки реализованных функций обработки;
- полнота и системность функций обработки;
- объём файлов программ;
- требования к операционной системе и техническим средствам обработки со стороны программного средства;
- объём дисковой памяти;
- размер оперативной памяти для запуска программ;
- тип процессора;
- версия операционной системы;
- наличие вычислительной сети и др.

Программные продукты имеют многообразие показателей качества, которые отражают различные аспекты.

Основная характеристика программного продукта - это его общая полезность, которая включает в себя мобильность, исходную полезность и удобство эксплуатации.

Мобильность ПП означает их независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п. Мобильный (многоплатформный) программный продукт может быть установлен на различных моделях компьютеров и операционных систем, без ограничений на его эксплуатацию в условиях вычислительной сети. Функции обработки такого программного продукта для массового использования без каких-либо изменений.

Исходная полезность характеризуется следующими показателями:

- надежность;
- эффективность;
- учет человеческого фактора;

Надежность работы ПП определяется бессбойностью и устойчивостью в работе программ, точностью выполнения предписанных функций обработки, возможностью диагностики возникающих в процессе работы программ ошибок.

Эффективность ПП оценивается как с позиций прямого его назначения - требований пользователя, так и сточки зрения расхода вычислительных ресурсов, необходимых для его эксплуатации. Расход вычислительных ресурсов оценивается через объем внешней памяти для размещения программ и объём оперативной памяти для запуска программ.

Учёт человеческого фактора означает обеспечение дружественного интерфейса для работы конечного пользователя, наличие котекстно- зависимой подсказки или обучающей системы в составе программного средства, хорошей документации для освоения и использования, заложенных в программном средстве функциональных возможностей, анализ и диагностику возникших ошибок и др.

Удобство эксплуатации включает следующие показатели качества:
- модифицируемость;
- коммуникативность.

Модифицируемость ПП означает способность к внесению изменений, например расширение функций обработки, переход на другую техническую базу обработки и т.п.

Коммуникативность ПП основана на максимально возможной их интеграции с другими программами, обеспечении обмене данными в общих форматах представления (экспорт/импорт баз данных, внедрение или связывание объектов обработки и др.).

Естественно, что в условиях существования рынка программных продуктов важными характеристиками являются: стоимость; количество продаж; длительность продаж (время нахождения на рынке); известность фирмы-разработчика и программы; наличие программных продуктов аналогического назначения.

Для оценки качества программного средства (ПС) используются различные способы получения информации о нём:
- измерительный - основан на получении информации о свойствах и характеристиках ПС с использованием инструментальных средств (например, объём ПС);
- регистрационный - получение информации во время испытаний или функционирования ПС, когда регистрируется и подсчитываются определённые события (число сбоев и отказов и др.);
- органолептический - анализ восприятия органов чувств, служащий для определения таких показателей как удобство применения ПС, его эффективность и др.;
- расчётный - на ранних этапах разработки применятся теоретические и эмпирические зависимости, статистические данные, накапливаемые при испытаниях, эксплуатации и сопровождении ПС;
- экспертный - рекомендован к применению при определении показателей наглядности, полноты и доступности программной документации, легкости освоения, структурности;
- социологические - основаны на обработке специальных анкет-вопросников.

Качество ПС определяется путём сравнения полученных расчётных значений показателей с соответствующими базовыми значениями показателей существующего аналога или расчётного ПС, принимаемого за эталонный образец.

Значения базовых показателей ПС должны соответствовать значениям показателей, отражающих современный уровень качества и прогнозируемый мирровый уровень. В качестве аналогов выбираются реально существующие ПС того же функционального значения, что и сравниваемое, с такими же основными параметрами, подобной структуры и

применяемые в тех же условиях эксплуатации.

Задание на лабораторную работу:

1. Скачать калькулятор любого производителя или взять разработанный студентами.
2. Сравнить два программных продукта: калькулятор фирмы Microsoft и калькулятор, написанный студентами (скачанный). Сравнение проводить по следующим оценочным элементам: надежность ПС, сопровождаемость, корректность. Критерии оценки (1 или 0)
3. Все сравнение занести в следующую таблицу

Код элемента	Наименование	Метод оценки	Оценка калькулятора фирмы Microsoft	Оценка калькулятора
Оценочные элементы фактора «Надежность ПС»				
		Всего		

Выводы:

Контрольные вопросы

1. Для чего предназначены программные продукты?
2. Какие варианты легального распространения программных продуктов существуют?
3. Чем определяется надежность ПН?
4. Как оценивается эффективность НН?
5. Что обозначает модифицируемость ПН?
6. На чем основана коммуникативность ПН?

Раздел 1. Проектирование ИС
Тема 1.7. Тестирование приложений АИС (4 часа)

Практическая работа №18
«Тестирование информационной системы».

Задачи обучающегося:
1. Научиться проводить тестирование информационной системы по различным показателям
Опорные понятия: оценка АИС.
Планируемый результат:
Студент должен
Уметь проводить оценку качества программного средства по различным показателям
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

Изучите теоретический материал:

Испытание информационной системы и тестирование программного продукта на первый взгляд одно и то же, но на практике это не совсем так. Если учесть, что информационная система – это не только используемые в ее составе программные компоненты, но и аппаратное и организационное обеспечение, то и в результатах ее испытаний должны быть отражены показатели выбранных серверов, рабочих станций,

сетевого оборудования (их надежность и производительность), а также эффективность разработанного регламента эксплуатации системы. Все виды испытаний информационной системы можно разделить на функциональные и нефункциональные тесты.

Функциональное тестирование призвано показать (доказать), что автоматизированные рабочие места информационной системы предоставляют пользователям ровно ту функциональность, которую они от нее ожидают. Система выполняет свои функции корректно.

Нефункциональное тестирование подтверждает или опровергает соответствие таких свойств информационной системы, как производительность, надежность, эргономичность и т.д. заданным на этапе ее проектирования параметрам. Система выполняет свои функции в срок, в должном объеме и с приемлемым качеством, и пользоваться ею удобно.

Виды функционального тестирования

Компонентное тестирование – испытание отдельных программных компонентов информационной системы, в ходе которых подтверждается корректность проводимых этими компонентами вычислений.

Интеграционное тестирование – испытания, направленные на выявление проблем взаимодействия отдельных компонентов системы. Если программная архитектура информационной системы довольно сложная, то в ней выделяются подсистемы, для каждой из которых проводят последовательно компонентное и интеграционное тестирование. В завершении проводят интеграционное тестирование всех выделенных подсистем, как компонентов единой системы.

Тестирование прототипа – испытания информационной системы на первых этапах ее разработки, когда готовы не все ее функциональные блоки. Отсутствующие компоненты заменяются функциональными заглушками, имитирующими их будущую работу. Информационная система на данном этапе представляет собой прототип целевого программного продукта.

Виды нефункционального тестирования

Нагрузочное тестирование (load testing) – испытание информационной системы в условиях прогнозируемой нормальной нагрузки. Под величиной нагрузки понимается количество запросов к системе, которое она должна успевать обрабатывать, не превышая определенное исходными требованиями время отклика.

Стрессовое тестирование (stress testing) – испытание информационной системы в условиях минимальных аппаратных ресурсах и максимально допустимой нагрузки. Цель стрессового тестирования, как понятно из названия, - проверить работоспособность системы в стрессовых ситуациях.

Объемное тестирование (volume testing) – испытания информационной системы в условиях максимальных (предельно допустимых) объемов информации в базе данных. Основным объектом тестирования в данном случае является зависимость времени отклика и прочих аспектов производительности системы от объемов контролируемых данных.

Тестирование стабильности (stability testing) – проверка, может ли испытываемая информационная система длительное время нормально функционировать в условиях, близких к нормальным условиям (средняя нагрузка, средние объемы данных, рекомендуемые аппаратное ресурсы и т.д.).

Тестирование надежности (reliability testing) – гибрид всех перечисленных ранее видов тестирования, направленный на то, чтобы проверить способность системы возвращаться к нормальному режиму работы после коротких периодов максимальной нагрузки, стрессов, предельных объемов данных и т.д.

Тестирование эргономики решений – испытания пользовательского интерфейса на предмет удобства и безопасности эксплуатации информационной системы.

Испытание информационной системы на этапах подготовки к эксплуатации

После завершения этапа реализации информационной системы Разработчик, совместно с Заказчиком, может проводить следующие виды испытаний.

Тестирование процесса установки (installation testing) – проверка корректности развертывания программных компонентов системы в различных ее конфигурациях, предусмотренных исходными требованиями.

Тестирование на различных конфигурациях (configuration testing) - проверка работоспособности системы при развертывании отдельных ее компонентов (серверной части, клиентских рабочих мест) в условиях всех возможных (предусмотренных исходными требованиями) вариантах операционных систем и конфигурациях аппаратных и программных ресурсов.

Приемочное тестирование (acceptance testing) – комплексное испытание информационной системы, выполняемое представителями Заказчика по специально разработанной Исполнителем программе и методике испытаний (ПМИ). Цель приемочного испытания – показать, что разработанная и развернутая на территории Заказчика информационная система делает ровно то, что от нее требуется и делает это с заданными параметрами производительности. В программу приемочных испытаний, помимо функциональных тестов, могут входить и тестирование процесса установки системы и тестирование ее работы на различных конфигурациях, а также все виды нефункционального тестирования.

Особенность приемочных испытаний, в сравнении с прочими этапами функционального и нефункционального тестирования как раз в том, что тестируемое решение развернуто на целевых аппаратных и системных программных ресурсах Заказчика (или арендованных Заказчиком), проводится представителями Заказчика (будущими пользователями) по программе, согласованной с Заказчиком. Решение об успешности приемочного тестирования также принимает Заказчик, переводя систему в эксплуатацию или отправляя ее на доработку.

Испытание информационной системы на этапах ее сопровождения

Регрессионное тестирование (regression testing) – тестирование, проводимое по результатам исправления обнаруженных дефектов и ошибок в работе системы и направленное на то, чтобы показать - исправленный дефект или ошибка в настоящий момент не проявляются, а целевая функциональность системы не нарушена.

Предварительное или дымовое тестирование (smoke testing) - вид испытаний, проводимый после выхода новой версии программных компонентов, входящих в состав информационной системы, целью которого является быстро показать общую работоспособность или неработоспособность системы. Если после установки новых версий программных продуктов от системы “не пошел дым”, то это означает, что на первый взгляд все работает, и можно приступать к более детальным видам тестирования. Дымовое тестирование позволяет экономить время, поскольку длится намного меньше, чем весь остальной комплекс испытаний, а его отрицательный результат говорит о том, что дальше можно и не продолжать, поскольку дефекты сборки уже обнаружены.

Задание:

Во время выполнения лабораторной работы необходимо составить набор тестов к разработанной ранее программе и провести ее отладку.

Составленный набор тестов необходимо представить в отчете.

Раздел 2. Сбор и анализ информации для определения потребностей клиента при проектировании ИС

Тема 2.1. Технология сбора информации (1 час)

Практическая работа №19

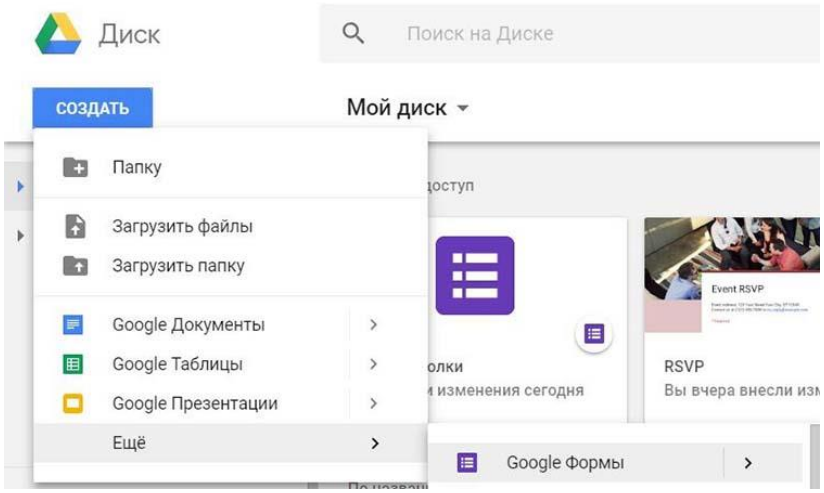
«Создание анкеты для выявления потребностей клиента и проведение анкетирования».

Задачи обучающегося:
1. Научиться проводить анкетирование для выявления потребностей
Опорные понятия: оценка АИС.
Планируемый результат:
Студент должен
Уметь проводить анкетирование
Пользоваться инструментальными средствами программного обеспечения
Необходимое оборудование: учебная литература, ПК

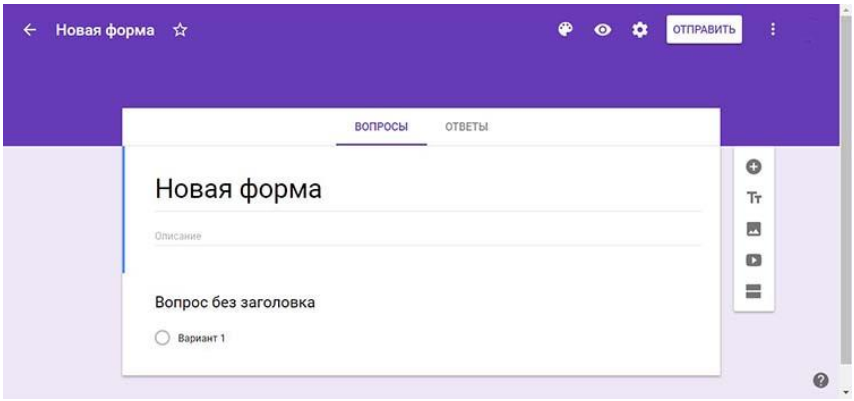
Порядок выполнения работы:

1.Создайте бриф на разработку ИС для заказчика по аналогии ниже

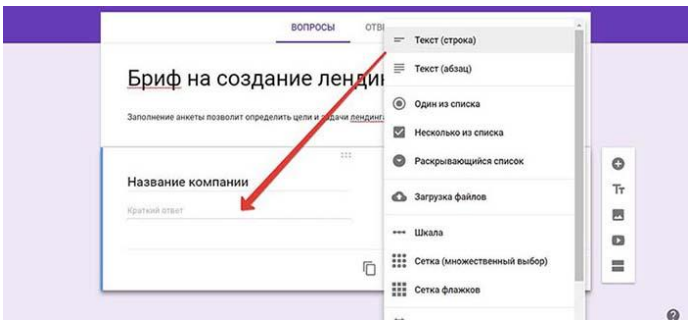
Пользоваться **Google Forms** можно только через аккаунт. Идете на **Google** Диск, нажимаете на кнопочку "Создать" и выбираете **Google формы**.



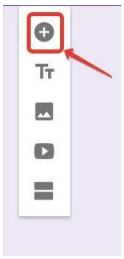
Шаг 1. Выбираете "Новая форма" и заменяете заголовок, описание на свои.



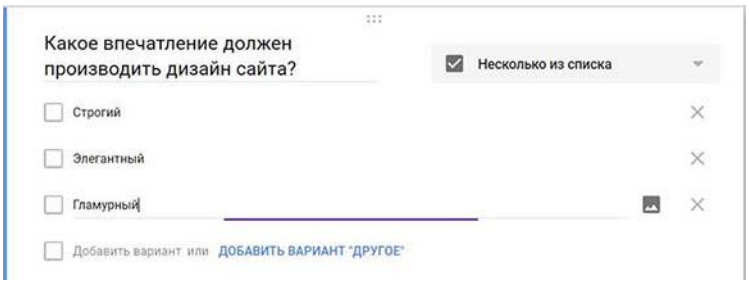
Шаг 2. В строке "Вопрос без заголовка" пишем свой вопрос и выбираем в раскрывающемся списке, тип ответа. Наш вопрос "Название компании" предполагает короткий ответ, значит мы выбираем тип ответа "Текст (строка)".



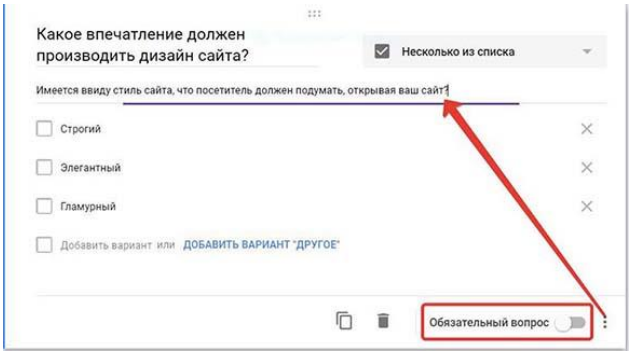
Шаг 3. Для добавления следующего вопроса, нажимаете на плюсик "Добавить вопрос" справа от брифа.



Теперь зададим вопрос, на который можно выбрать несколько вариант ответа. Справа выбираем "Несколько из списка".



Можно все вопросы сделать "обязательными", чтобы заказчик гарантированно заполнил все поля брифа. Также к сложному вопросу добавить, подсказку, раскрывающую смысл вопроса.



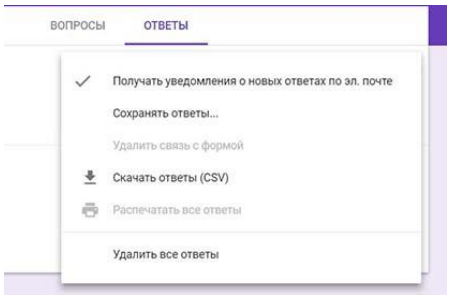
И так постепенно добавляете все вопросы в бриф, при желании сопровождая их картинками, видео и группируя их в разделы.

Шаг 4. Здесь вы выбираете способ доставки брифа на лендинг заказчику. У вас на выбор есть 4 способа:

Вставить бриф прямо в тело письма

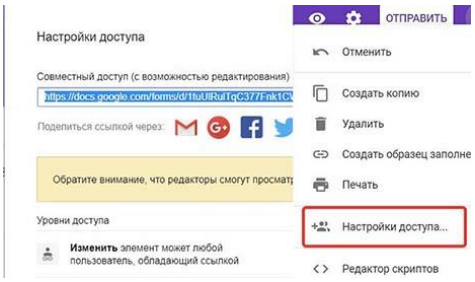
1. Отправить ссылку на веб-страницу с брифом
2. Скопировать **HTML-код** и вставить на ваш сайт <iframe>
3. Поделиться ссылкой в соцсетях

Шаг 5 Настроить уведомление на почту, как только клиент заполнит бриф. Перейти во вкладку "Ответы" и поставить галочку напротив "Получать уведомления о новых ответах по эл. почте". По умолчанию, там галочка не стоит, что немного странно. Как вы поняли, ответы вы увидите во вкладке "Ответы".

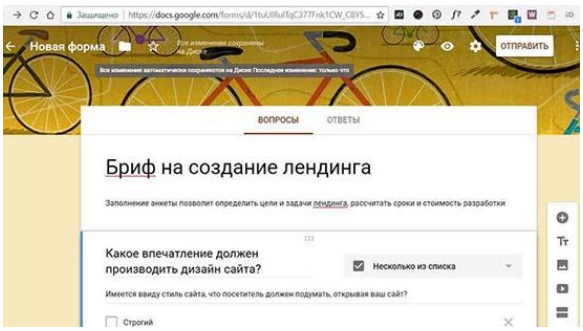


Совместный доступ

Если над проектом трудятся несколько специалистов – настройте совместный доступ к ответам брифа.



Нажмите иконку стиль, затем на значок "изображение" и выберите из готовых тем или загрузите свою фирменную.



Раздел 2. Сбор и анализ информации для определения потребностей клиента при проектировании ИС

Тема 2.2. Типовой состав документов на программный продукт (2 часа)

Практическая работа №19
«Разработка технического задания на программный продукт согласно ГОСТ19.102-77».

Задачи обучающегося:

1. Научиться разработки технического задания на создание программного продукта (ПП) с применением ГОСТ 19.102-77 «Стадии разработки программ и программной документации» и ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы».

Опорные понятия: оценка АИС.

Планируемый результат:

Студент должен

Уметь составлять техническое задание согласно ГОСТ

Пользоваться инструментальными средствами программного обеспечения

Необходимое оборудование: учебная литература, ПК

Порядок выполнения работы:

Основные теоретические сведения

Техническое задание — это документ, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления.

Техническое задание представляет собой документ, в котором

сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемо-сдаточных испытаний.

В разработке технического задания участвуют как представители заказчика, так и представители исполнителя.

В основе этого документа лежат исходные требования заказчика, анализ передовых достижений техники, результаты выполнения научно-исследова-тельских работ, предпроектных исследований, научного прогнозирования и т.п.

При разработке технического задания (ТЗ) необходимо решить следующие задачи:

- установить общую цель создания АИС;
- установить общие требования к проектируемой системе;
- разработать и обосновать требования, предъявляемые к информационному, математическому, программному, техническому и технологическому обеспечению;
- определить состав подсистем и функциональных задач;
- разработать и обосновать требования, предъявляемые к подсистемам;
- определить этапы создания системы и сроки их выполнения;
- провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности ее внедрения;
- определить состав исполнителей.

В Российской Федерации действует ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы», также на техническое задание существует стандарт ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению».

ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам устанавливает общие требования к оформлению программных документов. Программный документ должен состоять из следующих частей:

- Титульной;
- Информационной;
- Основной.

Титульная часть оформляется согласно ГОСТ 19.104-78 ЕСПД. Основные надписи.

Информационная часть должна состоять из аннотации и содержания. В аннотации приводят сведения о назначении документа и краткое изложение основной части.

Содержание включает перечень записей о структурных элементах основной части документа.

Состав и структура основной части программного документа устанавливается стандартами ЕСПД на соответствующие документы.

Основная часть технического задания должна содержать следующие разделы:

ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению :

- введение;
- основания для разработки;
- назначение разработки;
- требования к программному продукту;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приѐмки;

В зависимости от программного продукта допускается уточнять содержание разделов, объединять отдельные из них, вводить новые разделы. В техническое задание допускается включать приложения.

Задания для выполнения

1. Разработать техническое задание на программный продукт (см. варианты заданий) в соответствии с ГОСТ 19.201-78 и ГОСТ 34.602—89
2. Оформить работу в соответствии с ГОСТ 19.106—78. При оформлении использовать MS Office.
3. Сдать и защитить работу.

Варианты заданий

1. Разработать программный модуль «Учет успеваемости студентов».

Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

2. Разработать программный модуль «Личные дела студентов».

Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.

3. Разработать приложение Windows «Органайзер».

Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.

4. Разработать приложение Windows «Калькулятор».

Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и желательно (но не обязательно) несколько математических функций.

5. Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.).

Модуль предназначен для использования сотрудниками отдела кадров и деканата.

7. Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень).

Модуль предназначен для использования сотрудниками профкома и отдела кадров.

8. Разработать программный модуль «Автосервис».

При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.

9. Разработать программный модуль «Учет нарушений правил дорожного движения».

Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.

10. Разработать программный модуль «Картотека агентства недвижимости», предназначенный для использования работниками агентства.

В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки на обмен (куплю, продажу) производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.

11.Разработать программный модуль «Картотека абонентов АТС».

Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.

12. Разработать программный модуль «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты.

В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиа билетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.

13. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена).

Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.

14. Разработать программный модуль «Автостоянка».

В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.

15. Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме.

Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

Примечание. При разработке программы не ограничиваться функциями, приведенными в варианте, добавить несколько своих функций.

Содержание и оформление отчета по лабораторной работе

Отчет по лабораторной работе должен состоять из:

1. Постановки задачи.

2. Технического задания на программный продукт.

Отчѐт должен содержать титульный лист, аннотацию, содержание и основную часть, оформленную в соответствии с ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» или ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению».