

**Департамент образования Вологодской области  
бюджетное профессиональное образовательное учреждение  
Вологодской области  
«ВОЛОГОДСКИЙ СТРОИТЕЛЬНЫЙ КОЛЛЕДЖ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
к практическим работам  
ПО ДИСЦИПЛИНЕ  
ОП.05. УСТРОЙСТВО И ФУНКЦИОНИРОВАНИЕ  
ИНФОРМАЦИОННОЙ СИСТЕМЫ**

2017г.

Рассмотрено на заседании предметной цикловой комиссии общепрофессиональных, специальных дисциплин и дипломного проектирования по специальностям 08.02.01 Строительство и эксплуатация зданий и сооружений, 08.02.07 Монтаж и эксплуатация внутренних сантехнических устройств, кондиционирования воздуха и вентиляции, 43.02.08 Сервис домашнего и коммунального хозяйства.

Данные методические указания предназначены для студентов специальности 09.02.04 Информационные системы (по отраслям) БПОУ ВО «Вологодский строительный колледж» при выполнении практических работ по дисциплине ОП.05. Устройство и функционирование информационной системы.

Настоящие методические указания включают в себя краткий теоретический материал, практические задачи, указания к их выполнению.

Методические указания могут быть рекомендованы к использованию студентами специальности 09.02.04 Информационные системы (по отраслям).

***Авторы:***

*Попова И.В.* - преподаватель

## Содержание

Введение

Тематический план проведения практических работ

### **Раздел 1. Общие сведения об автоматизированных информационных системах**

#### **Тема 1.1. Автоматизация производства. Понятие организационной структуры**

Практическая работа №1. Оценка предметной области, описание бизнес-процессов организации

### **Раздел 2. Моделирование и проектирование АИС**

#### **Тема 2.1. Основные принципы моделирования ИС**

Практическая работа №2. Построение моделей систем. Построение структурной схемы сложной модели. Построение семантической сети

#### **Тема 2.3. Технология проектирования ИС**

Практическая работа №3. Изучение интерфейса инструментальной среды ERWin

Практическая работа №4. Разработка информационной модели для реализации каталога документов

Практическая работа №5. Разработка информационной модели

Практическая работа №6. Изучение интерфейса инструментальной среды BPWin.

Практическая работа №7. Разработки функциональной модели для реализации бизнес-плана банка по привлечению и размещению ресурсов

Практическая работа №8. Разработка функциональной модели.

### **Раздел 3. Реализация АИС**

#### **Тема 3.2. Технические средства построения ИС**

Практическая работа №9. Выбор вычислительной модели. Выбор конфигурации сервера.

#### **Тема 3.3. Организация труда при разработке ИС**

Практическая работа №10. Расчет показателей и критериев оценивания ИС.

## Введение

Практическое занятие – это одна из форм систематических учебных занятий, на которых обучающиеся приобретают необходимые умения и навыки по определенному разделу дисциплины, входящей в состав учебного плана.

Общие цели практического занятия сводятся к закреплению теоретических знаний, более глубокому освоению уже имеющихся у обучающихся умений и навыков и приобретению новых умений и навыков, необходимых им для осуществления своей профессиональной деятельности и составляющих квалификационные требования к специалисту исходя из специальности и занимаемой им должности.

Основными задачами практических занятий являются:

- ✓ углубление теоретической и практической подготовки обучающихся;
- ✓ отражение в учебном процессе требований научно-технического прогресса, современных достижений науки и техники;
- ✓ развитие инициативы и самостоятельности обучающихся во время выполнения ими практических занятий.

Дидактические цели практических занятий сводятся:

- ✓ к закреплению теоретических знаний;
- ✓ овладению умениями и навыками, предусмотренными целями конкретной темы или раздела программы;
- ✓ изучению на практике методов исследований, обучению приемам исследовательской работы в условиях аудитории;
- ✓ развитие инициативы и самостоятельности в работе, оценке научно-теоретических положений.

Практические занятия могут быть:

- ✓ работами по образцу, цель которых решение типовых задач в строгом соответствии с содержанием методических указаний и рекомендациями преподавателя;
- ✓ реконструктивными работами, при которых обучающийся получает методические указания лишь в схематическом виде, а для выполнения работы требуется элемент творчества на основе ранее приобретенных умений и навыков;
- ✓ вариантными работами, в ходе которых изыскиваются новые варианты выполнения работы, не предусмотренные указаниями преподавателя и методическими рекомендациями;
- ✓ творческими работами, цель которых поиск решения новой задачи.

Объем практических занятий в часах и по тематике определяется учебным планом.

## Тематический план проведения практических работ

### *Цели проведения практических занятий*

Цели и задачи выполнения практических заданий по дисциплине – требования к результатам освоения дисциплины.

В результате выполнения практических работ по дисциплине ОП. 05. Устройство и функционирование информационной системы обучающийся должен

#### **уметь:**

- выделять жизненные циклы проектирования информационной системы;
- использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;
- использовать и рассчитывать показатели и критерии оценивания информационной системы, осуществлять необходимые измерения;

#### **знать:**

- цели автоматизации производства;
- типы организационных структур;
- реинжиниринг бизнес-процессов;
- требования к проектируемой системе, классификацию информационных систем, структуру информационной системы, понятие жизненного цикла информационной системы;
- модели жизненного цикла информационной системы, методы проектирования информационной системы;
- технологии проектирования информационной системы, оценку и управление качеством информационной системы;
- организацию труда при разработке информационной системы;
- оценку необходимых ресурсов для реализации проекта

#### **обладать общими компетенциями:**

- ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
- ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
- ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность
- ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
- ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.
- ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
- ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.
- ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
- ОК 9. Ориентироваться в условиях частой смены технологий и профессиональной деятельности

#### **обладать профессиональными компетенциями:**

- ПК 1.1 Собирать данные для анализа использования и функционирования информационной системы, участвовать в составлении отчетной документации, принимать участие в разработке проектной документации на модификацию информационной системы.

- ПК 1.3 Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения.
- ПК 1.4 Участвовать в экспериментальном тестировании информационной системы на этапе опытной эксплуатации, фиксировать выявленные ошибки кодирования в разрабатываемых модулях информационной системы.
- ПК 1.5 Разрабатывать фрагменты документации по эксплуатации информационной системы.
- ПК 1.6 Участвовать в оценке качества и экономической эффективности информационной системы
- ПК 1.9 Выполнять регламенты по обновлению, техническому сопровождению и восстановлению данных информационной системы, работать с технической документацией.

### Тематический план практических занятий

№ п/п	Практические занятия	Объем часов
1	Оценка предметной области, описание бизнес-процессов организации	2
2	Построение моделей систем. Построение структурной схемы сложной модели. Построение семантической сети	2
3	Изучение интерфейса инструментальной среды ERWin	2
4	Разработка информационной модели для реализации каталога документов	2
5	Разработка информационной модели	2
6	Изучение интерфейса инструментальной среды BPWin.	2
7	Разработки функциональной модели для реализации бизнес-плана банка по привлечению и размещению ресурсов	2
8	Разработка функциональной модели.	2
9	Выбор вычислительной модели. Выбор конфигурации сервера	2
10	Расчет показателей и критериев оценивания ИС.	2
<b>ИТОГО :</b>		<b>20</b>

## **Практическая работа №1.**

### **Оценка предметной области, описание бизнес-процессов организации**

#### ***1 Этап. Импортное моделирование***

Прежде всего, Вам необходимо импортировать по три модели (функциональную, организационную и информационную), названия которых совпадают с номером вашего варианта. Для этого Вы переходите в режим Import/Export (на горизонтальной панели инструментов нажимае-



те на кнопку Import/Export), затем на панели инструментов нажимаете на ADL Import и сначала указываете файл, который хотите импортировать, а потом папку, в которую необходимо поместить данные модели.

#### ***2 Этап. Построение функциональных моделей***

Для построения функциональных моделей требуется перейти в режим моделирования –



Modeling и открыть функциональную модель, которую Вы только что импортировали (на панели горизонтального меню: Model → Open...).

Перед Вами появится набор элементов, из которых, с помощью приведённого в приложении 1 описания, Вы должны составить логически законченную функциональную модель. На этой диаграмме необходимо в правильной последовательности соединить между собой изолированные вершины. Таким образом, Вы должны добавить только соединительные элементы (все остальные элементы уже заданы).

Построение модели процессов должно соответствовать правилам: системным (обязательным) и рекомендуемым. Ниже приведены только те правила построения моделей, которые должны соблюдаться в данной лабораторной работе. В последующих лабораторных работах эти правила будут дополнены.

К первой группе требований относятся следующие:

- модель бизнес-процессов обязательно должна содержать элементы начала и конца;
- если процесс содержит параллельно выполняемые действия, то для визуализации этих участков на диаграммах в Adonis должны быть использованы элементы разветвления и объединения процес-

сов;

- в случае процесса с параллельными участками (операциями) при задании переменных они должны быть объявлены как глобальные (в notebook/variable score/global);
- каждый блок на диаграмме должен иметь уникальное имя, так как

все события отличаются местом, временем исполнения или исполнителями;









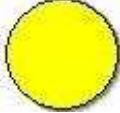


- при построении модели необходимо указывать операции, связанные с получением и сдачей заданий, документов и пр.

Существует также несколько рекомендуемых правил: диаграмма должна быть

- наглядной и легко читаемой; при разработке модели необходимо ее детализировать в той степени, какая требуется для описания поставленной задачи, но не более;
- при необходимости следует элементы модели сопровождать комментариями.

В таблице 1 представлены основные элементы функциональных моделей.

Таблица 1. Элементы моделирования бизнес-процессов Adonis

Основные элементы		Соединители	
	начало процесса (process start)		связь предыдущего элемента с последующим (subsequent)
	подпроцесс/ подпрограмма (subprocess)		связь от элемента к комментарию (has note)
	процесс, действие, операция (activity)		
	элемент условного перехода, ЕСЛИ (decision)		
	разветвление процессов (parallelity)		
	объединение процессов (merging)		
	конец процесса (end)		
	комментарий (note)		
	группа (aggregation)		



У каждого элемента функциональной модели (за исключением соединителя элемента с комментарием) имеется записная книжка Notebook, открыть которую можно по двойному щелчку или с помощью контекстного меню. Она содержит одну или несколько глав, каждая из которых может содержать одну или более страниц. Признаки, описанные в записной книжке, могут обновляться (наименование ссылки на модель автоматически изменится при переименовании оригинала).

Большинство полей сопровождаются описанием/подсказкой – значок «i» в верхнем правом углу над полем.

Структура «записной книжки» основных элементов модели бизнеспроцессов (notebook) Adonis представляет собой несколько закладок различного функционального назначения:

- Description – описание элемента;
- Input/output – входная и выходная документация (данные параметры позволяют для каждого функционального элемента назначать документы проекта; например, для разработки технологического процесса входными данными будет чертеж детали, а выходными комплект технологической документации);
- Time/cost – параметры времени и стоимости деятельности;
- Working environment – описание рабочей среды;

Other simulation data – дополнительные параметры имитационного моделирования (например, синхронный и асинхронный процесс);

- Simulation result – результаты имитационного моделирования (все параметры доступны только для просмотра).

В рамках данной лабораторной работы необходимо заполнить закладки Description и Input/output. В приложении 2 представлено описание атрибутов этих закладок.

### **3 Этап. Построение организационных моделей**

Теперь Вам необходимо открыть файл с организационной структурой. Перед Вами также появится набор элементов рабочей среды, которые необходимо правильно соединить между собой согласно описанию, приведённому в приложении 1.

В таблице 2 представлены основные элементы моделирования рабочей среды Adonis.

Модель рабочей среды может состоять из нескольких независимых частей, отображающих работу различных отделов компании. На диаграмме обязательно должны быть указаны департамент, служащие и роли (специализация сотрудников). В качестве «performer» (исполнителя) можно указать должность работника или конкретную личность.

Структура «записной книжки» элементов модели рабочей среды содержит следующие закладки:



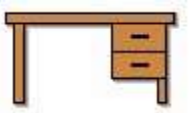







- Description – описание элемента;
-

Simulation data – параметры, необходимые для проведения имитационного моделирования процессов;

- Simulation results – результаты имитационного моделирования (все параметры доступны только для просмотра);
- Beschreibung – то же, что и Description;
- Data for analysis – данные, необходимые для проведения расчетов.

В данной лабораторной работе заполняется только поле Name закладки Description.

Таблица 2. Основные элементы моделирования рабочей среды Adonis

Основные элементы		Соединители	
	отдел, организационный модуль, департамент (organization unit)		иерархическая связь, связь между отделами (is subordinated)
	исполнитель, служащий (performer)		связь принадлежности, от служащего к отделу (belongs to)
	роль, специализация, должность (role)		связь принадлежности, от начальника/менеджера к отделу (позволяет выделить руководителей среди прочих служащих) (is man- ager)
	комментарий (note)		связь от элемента «performer» к «role», позволяет определить специализацию каждого исполните- ля (has role)
	группа (aggregation)		связь от элемента к комментарию (has note)

#### 4 Этап. Построение информационных моделей

Данный вид диаграмм представляет собой перечень документов (библиотеку процесса). Данная модель должна содержать все документы, входящие в проект.






Структура «записной книжки» элементов Document model содержит только описательные характеристики (закладка «Description»). Помимо имени документа (Name) и описания (Description), характеристиками документа являются:

- Executable - тип файла (Word, Excel, PowerPoint);
- Program argument – местоположение файла, механизм поиска созданного ранее файла; можно указать любой документ, к которому имеется доступ;
- Responsible role – дает возможность указать роль (должность) сотрудника, ответственного за данный документ (его создание, распространение, корректировку и пр.);
- Reference document model – ссылка на присоединенную модель документов, позволяет детально описать каждый документ проекта.

Для построения информационной модели Вам необходимо открыть файл с моделью документов, которую Вы импортировали. В данном случае Вы должны просто оформить информационную модель (любым образом) с помощью комментариев и соединительных стрелок. Данный вид диаграмм в Adonis не влияет на проведение различных расчётов, поэтому здесь нет жёстких ограничений по построению моделей.

Таблица 3 содержит описание элементов информационного моделирования.

Таблица 3. Элементы информационного моделирования Adonis

Основные элементы		Соединители	
	документ (document)		связь между документами (has subdocument)
	комментарий (note)		связь от элемента к комментарию (has note)
	группа (aggregation)		

### 5 Этап. Установление связей между функциональной, организационной и информационной моделями

Модели рабочей среды и документов служат окружением диаграмм бизнес-процессов. Связи с ними задаются в Notebook какой-либо конкретной функции (Activity). Для модели рабочей среды связь устанавливается на закладке Description в поле Responsible role (из каталога выбирается не только конкретная модель, но и роль исполнителя данной функции

(технолог, начальник участка и пр.)); для модели документов – на закладке Description в поле Referenced document и закладке Input/Output в полях Input (указывается входной документ) и Output (указывается выходной документ).

Для визуализации на диаграмме бизнес-процесса прикрепленной модели рабочей среды и отображения ссылки существует два способа:

- На панели меню выбрать View → Views → Visualize responsible role → на появившейся панели указать модель → Ok → Yes → Ok
- В Notebook каждой функции установить галочку напротив Display responsible role

## Пример выполнения

### 1 Этап. Исходные данные

Таблица 4. Описание этапов процесса «Проектирование КД оснастки»

№ п./п.	Этап	Входной документ	Выходной документ	Исполнитель	Кол-во исполнителей	Должность исполнителя	Подразделение
1	Проработка задания	Заявка на проектирование КД оснастки	Техническое задание на проектирование КД оснастки	Инженерконструктор	3	Инженер	КБ
2	Проработка детали (на технологичность, конструкторские особенности)	Техническое задание на проектирование КД оснастки	Техническое задание на проектирование КД оснастки	Инженерконструктор	3	Инженер	КБ
3	Конструирование	Техническое задание на проектирование КД оснастки	Комплект КД	Инженерконструктор	3	Инженер	КБ
4	Согласование с цехом-заказчиком → <u>Условный переход 1:</u> <u>Требуется перепроектирование КД оснаст-</u>	Комплект КД	Комплект КД	Инженерконструктор	3	Инженер	КБ

<i>№ п./п.</i>	<i>Этап</i>	<i>Входной документ</i>	<i>Выходной документ</i>	<i>Исполнитель</i>	<i>Кол-во исполнителей</i>	<i>Долж- ность ис- полнителя</i>	<i>Подразделение</i>
	<u>ки?</u>						
5	<i>Если перепроектирования КД оснастки не требуется → Оконча- ние процесса</i>						
6	<i>Если перепроектирование КД оснастки требуется → Перепроектирование КД оснастки → Возврат на четвёртый этап</i>	Комплект КД	Комплект КД	Инженерконструктор	3	Инженер	КБ

В данном случае нам необходимо построить модель проектирования КД оснастки. При этом не нужно строить модели с нуля. Как уже было сказано, требуется импортировать заготовки функциональной, организационной и информационной моделей.

## 2 Этап. Построение функциональной модели

1. После того, как мы импортировали модели, открываем функциональную модель.  
Перед нами должен появиться набор элементов бизнеспроцесса.

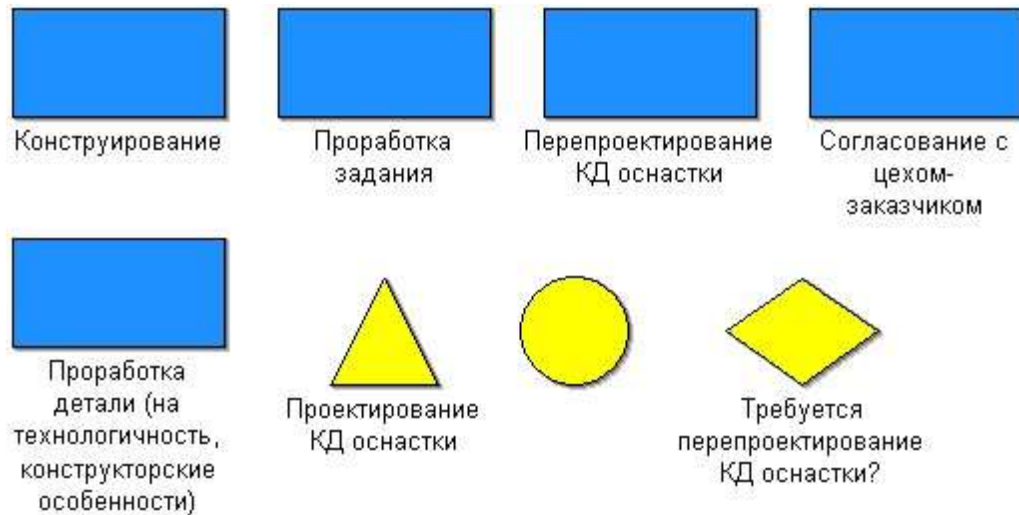


Рис. 1. Набор элементов функциональной модели

2. В таблице 4 приведено подробное описание данного процесса. Таким образом, необходимо просто упорядочить набор элементов и добавить соединители. В результате у нас должна получиться следующая модель:

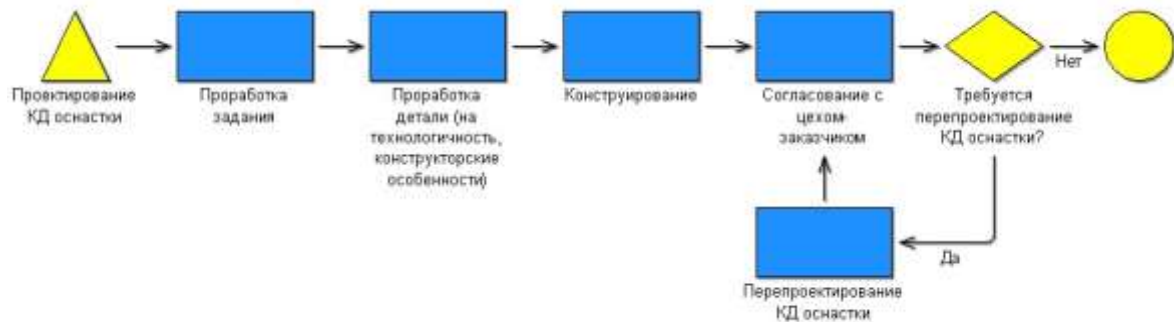


Рис. 2. Модель бизнес-процесса

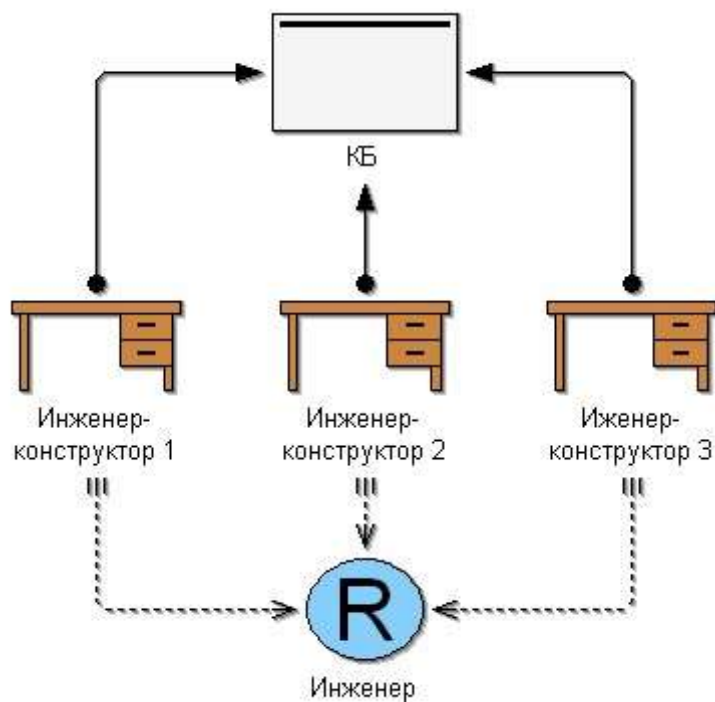
## 3 Этап. Построение организационной модели 1. Открываем

заготовку организационной модели.



*Рис. 3. Набор элементов организационной модели*

2. Так же как и при построении функциональной модели, необходимо построить модель рабочей среды с помощью описания, представленного в таблице 4.



*Рис. 4. Организационная модель*

**4 Этап. Построение информационных моделей**

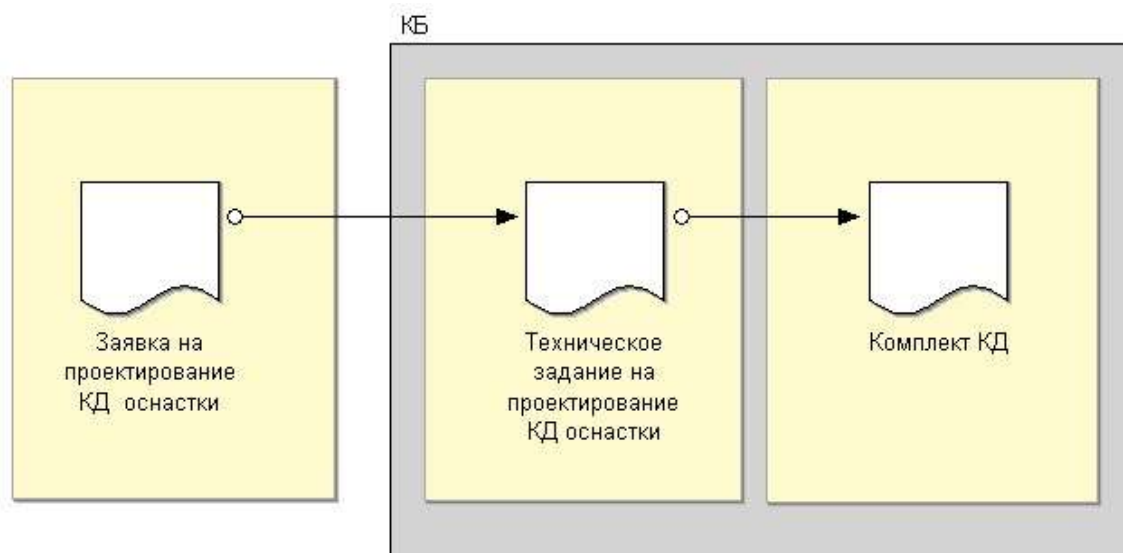
1. Открываем заготовку модели документов:





**Рис. 5. Набор элементов информационной модели**

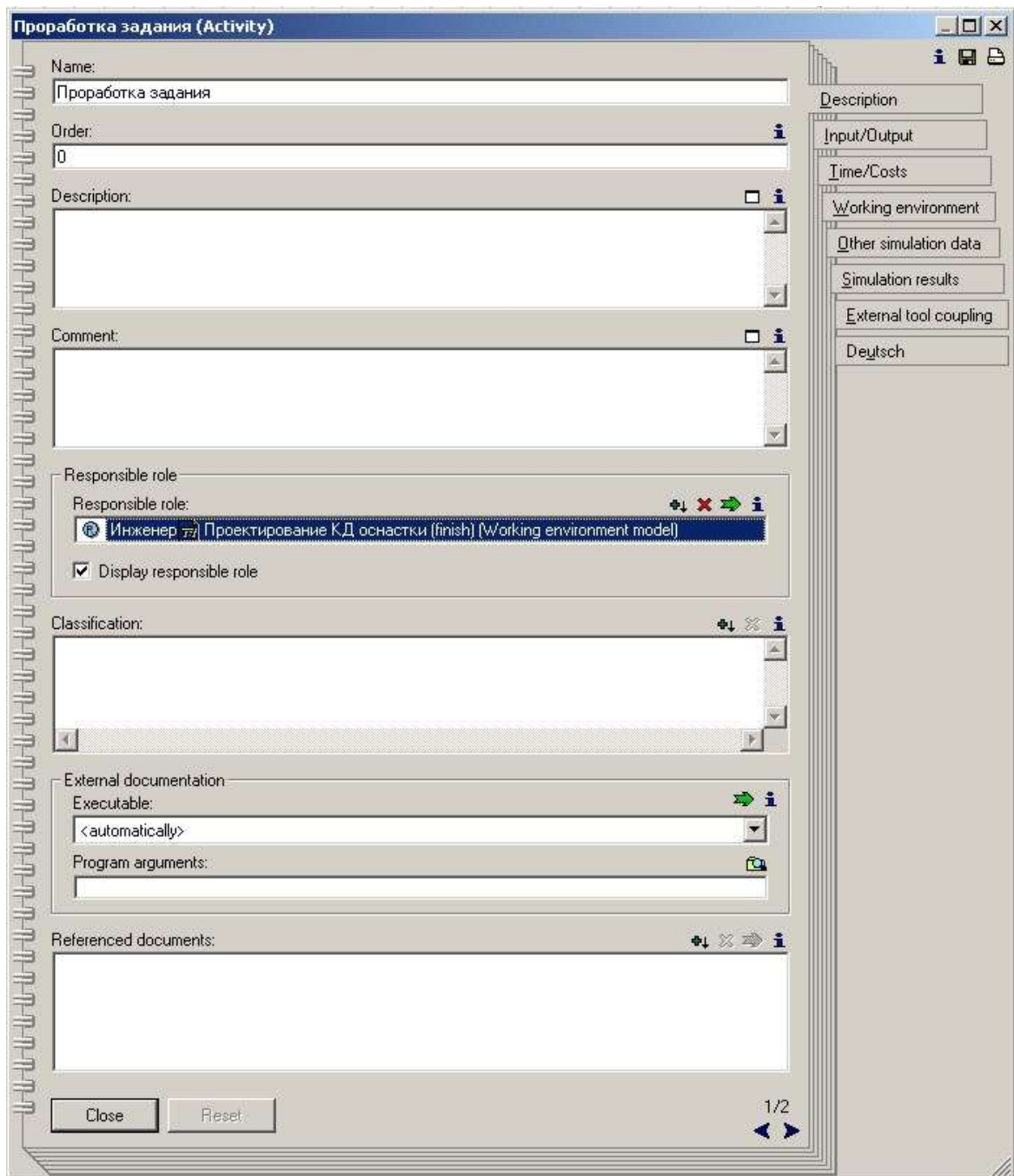
2. В данном случае необходимо просто визуальное оформление диаграммы с помощью соединителей и комментариев.



**Рис. 6. Информационная модель**

**5 Этап. Установление связей между функциональной, организационной и информационной моделями**

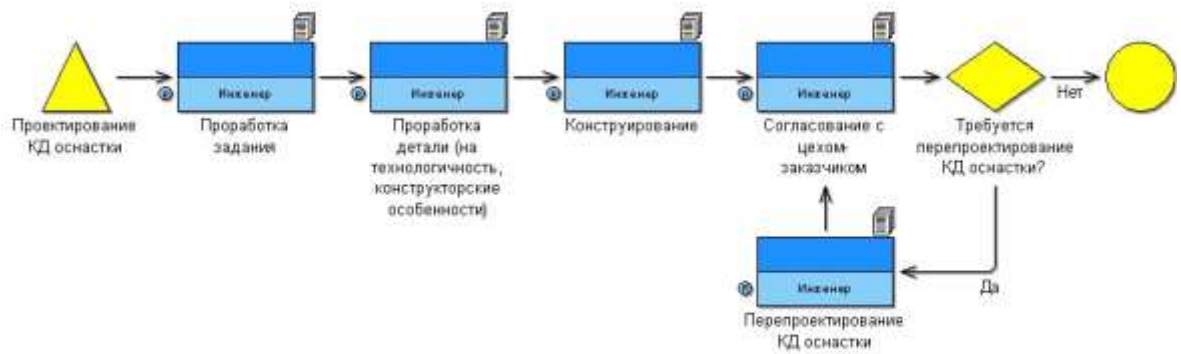
1. Открываем построенную функциональную модель (рис. 2)
2. Открываем записную книжку первой функции (Activity). На закладке Description в поле Responsible role задаём должность исполнителя.



**Рис. 7. Notebook с атрибутами элемента функциональной модели**

3. На той же закладке в поле Referenced documents указываем документы, которые необходимы при выполнении данной функции.
4. На закладке Input/Output в поле Input также необходимо указать входной документ, в Output – выходной документ.
5. С помощью функции Visualize responsible role отображаем ссылки на прикрепленную модель рабочей среды.

В результате функциональная модель должна выглядеть следующим образом:



**Рис. 8. Функциональная модель с установленными связями**

Таким образом, построив функциональную, организационную и информационную модели, а также установив взаимосвязи между ними, мы создали концептуальную модель производства.

## Практическая работа №2.

### Построение моделей систем. Построение структурной схемы сложной модели.

#### Построение семантической сети

**Задание 1.** Используя текстовый редактор, нарисовать схему, отражающую состав и структуру танкового батальона армии ФРГ, приведенную на рис. 1.

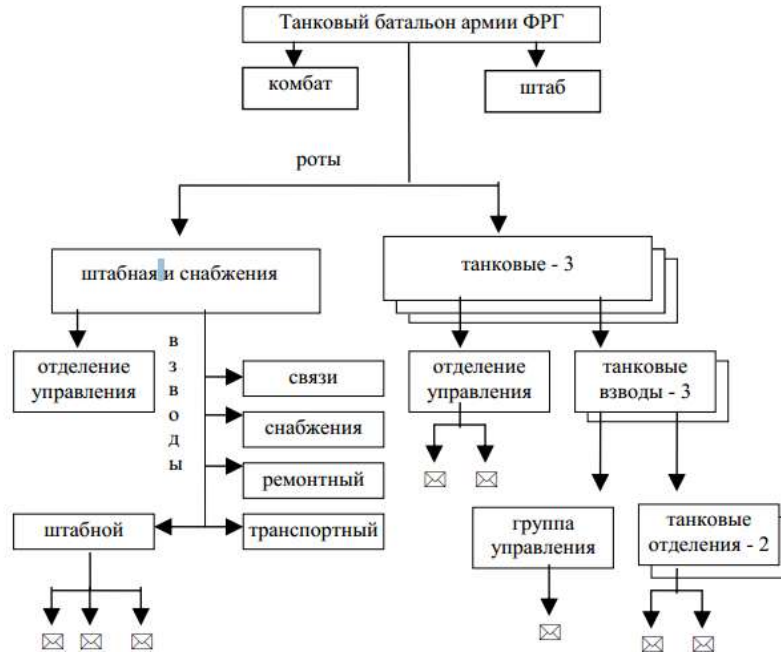


Рис.1.

Подсчитать количество танков в роте и общее количество танков в батальоне

**Задание №2.** Построить граф, отображающий состав и структуру мотопехотного батальона армии ФРГ (образца 1970 года) по следующему описанию.

Батальон на БМП (боевые машины пехоты) имел численность 764 человека. Во главе батальона стоял командир, которому подчинялись штаб и 5 рот: штабная и снабжения, минометная и 3 мотопехотные. Рота штабная и снабжения состояла из управления и 3 взводов: штабного, связи и снабжения. В штабном взводе было 4 отделения: штабное, мотоциклистов, транспортное и разведывательное. Во взводе связи было 3 отделения радиосвязи и 2 отделения проводной связи. Во взводе снабжения было 4 отделения: материально-технического обеспечения, продовольственного снабжения, транспортное и санитарное. Минометная рота состояла из управления, двух отделений передового наблюдения, отделения обеспечения и 6 расчетов 120-мм минометов. Каждая мотопехотная рота (163 чел.) имела в своем составе 3 мотопехотных взвода по 48 чел. Каждый взвод состоял из группы управления и 4 мотопехотных отделений по 10 чел. Отделение делилось на две группы: первая - командир машины, наводчик-оператор (МБП

"Мардер" оснащался 20-мм пушкой и двумя пулеметами) и механик-водитель; вторая - командир отделения, пулеметчик, гранатометчик, огнеметчики 3 стрелка.

**Задание №3.** Используя графические средства, нарисовать схему, отражающую классификацию геометрических объектов, приведенную на рис.2.

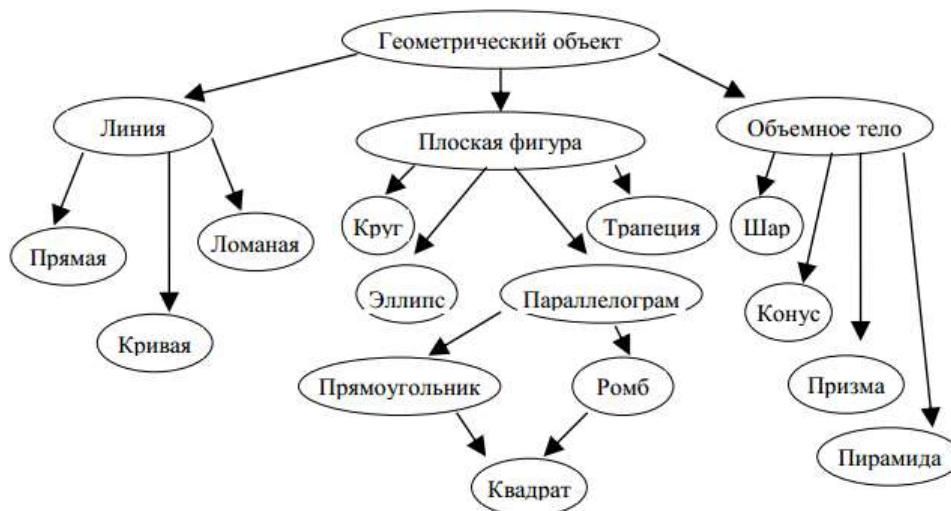


Рис.2.

**Задание №4.** Построить граф классификации для описанной системы «Биологическая классификация»

.Согласно биологической классификации выделяют 3 империи (надцарства): археобактерии, эукариоты и прокариоты. К империи эукариотов относятся царства грибов, растений и животных. К царству животных относятся типы членистоногих, моллюсков, иглокожих, кишечнополостных, хордовых и др. К типу хордовых относятся классы рыб, амфибий, рептилий, млекопитающих, птиц. К классу млекопитающих относятся отряды китов, ластоногих, хищных, грызунов, копытных и др.. К отряду хищных относятся семейства медвежьих, енотовых, псовых, виверровых, кошачьих и др.. К семейству псовых относятся роды лисиц, енотовидных собак, собак, фенек, песцов и др. К роду собак относятся виды собак домашних, волков, шакалов, койотов. К виду собак домашних относятся овчарки, спаниели, водолазы, сенбернары, доги, болонки и др.

**Задание №5.** Построить граф классификации для описанной системы «Классификация в русском языке»

Предложения в русском языке классифицируются по составу, по интонации и по цели высказывания.

По составу предложения делятся на нераспространенные и распространенные. Нераспространенные предложения состоят только из двух членов: подлежащего и сказуемого. Пример нераспространенного предложения: "Птицы прилетели". Распространенные предложения состоят из подлежащего, сказуемого и второстепенных членов предложения. Пример распространенного предложения: "Ранней весной прилетели первые птицы".

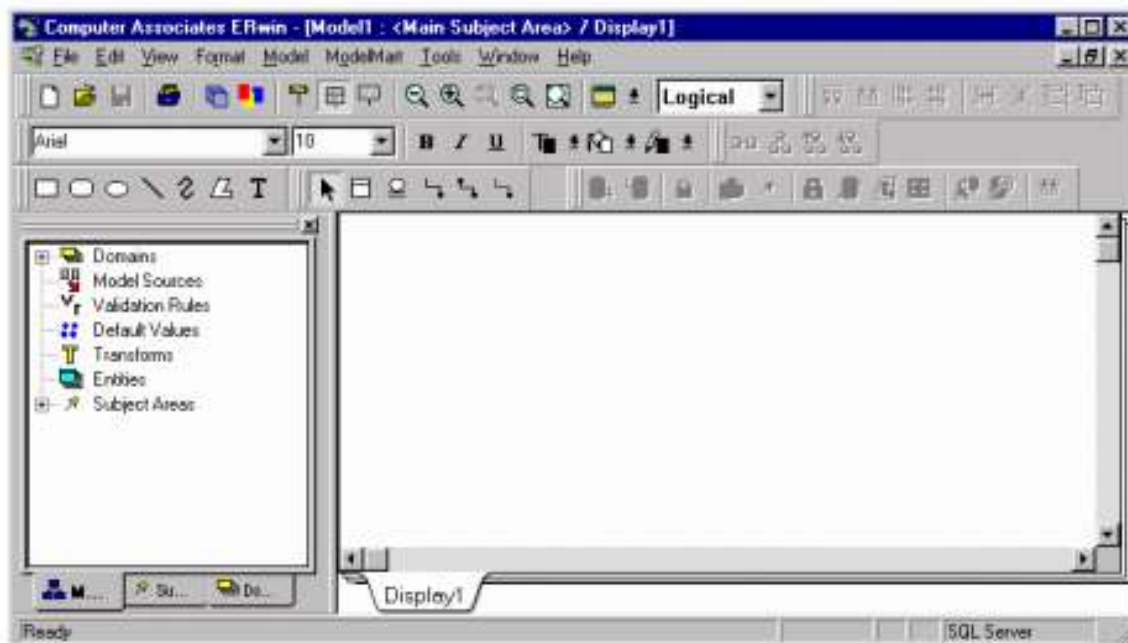
По интонации предложения делятся на восклицательные ("Пришла весна!") и невосклицательные ("Пришла весна.").

По цели высказывания предложения делятся на повествовательные, вопросительные и побудительные. Повествовательное предложение: "Мы собрали много грибов и ягод". Вопросительное предложение: "Вы собрали много грибов и ягод?". Побудительное предложение: "Собирайте грибы и ягоды!".

### Практическая работа №3. Изучение интерфейса инструментальной среды ERWin

#### Описание интерфейса программы ERWin

Основное окно программы имеет следующий вид:



После запуска программы ERWin на экране появится окно программы с диалогом, предлагающий открыть существующую модель или создать новую.

Если пользователь выбрал открытие существующей модели, ему предложат стандартный диалог Windows для выбора файл модели. Либо пользователь может сразу выбрать имя файла из списка недавно использованных.

При выборе пункта «*Create a new model*» пользователю предлагается выбрать тип

создаваемой модели: логическая или физическая, а также воспользоваться одним из ранее созданных шаблонов. Если среди опций была выбрана модель типа «*Physical*» или «*Logical/Physical*», будет предложено выбрать также тип целевой базы данных, куда будет

Далее на экране появляется основное окно программы (Рис. 1).

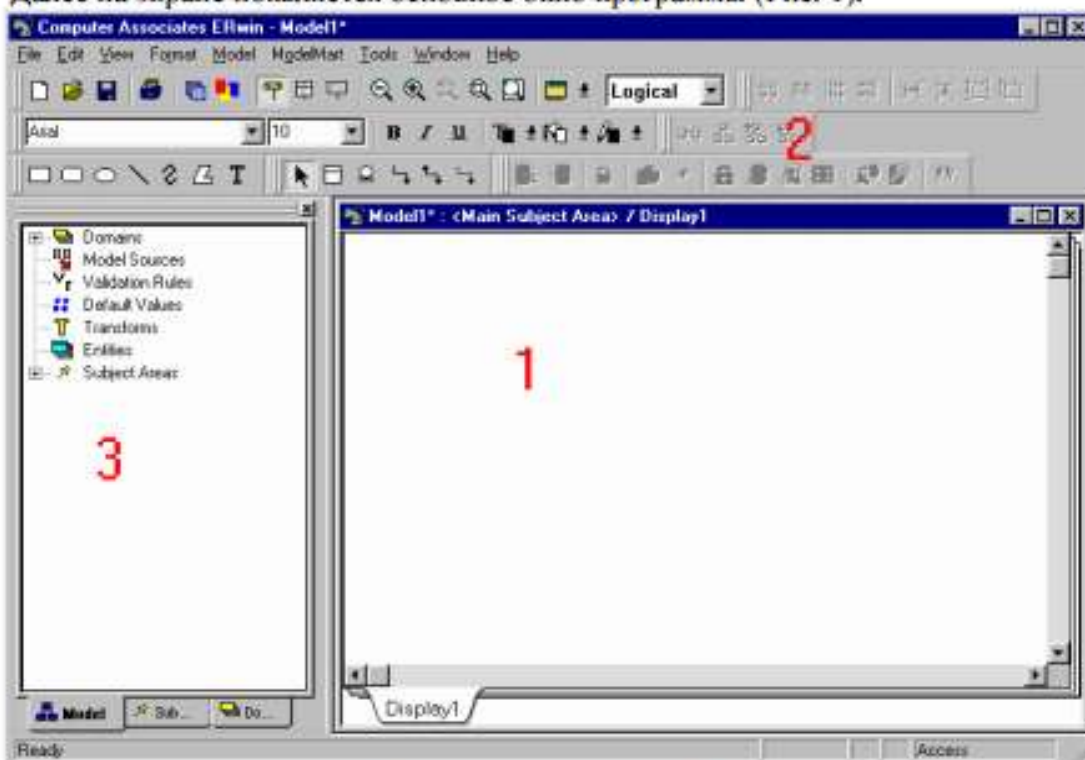


Рис. 1. Основное окно ERWin

Основное окно программы содержит следующие части:

- 1) Область модели данных или рабочая область
- 2) Панели инструментов
- 3) Браузер элементов модели

Рассмотрим подробнее содержимое каждой из частей программы:

Область модели данных – содержит собственно разрабатываемую модель. Если создана новая модель – это окно пустое, если открыта существующая – содержит сущности и связи между ними.

Панели инструментов: эти панели содержат практически все используемые при работе элементы. По умолчанию все панели отображаются на экране. При необходимости пользователь может отключить или, наоборот, включить требуемые модели, используя меню «**View – Toolbars**». Имеются следующие панели инструментов:

- **Standard** – содержит кнопки для управления файлами (новый, открыть, сохранить), кнопки отображения сущностей (только сущности, сущности и атрибуты, сущности и их определение), кнопки управления масштабом изображения, переключатель уровней модели (логический/физический).
- **Toolbox** – инструментальные кнопки создания элементов модели, сущностей и связей. Набор кнопок незначительно различается для логической и физической модели. Содержит кнопки: стрелка – выбор объекта, создание независимой сущности, создание связки-подкатегории, создание сущности-представления, создание идентифицирующей и неидентифицирующей связи «один-к-одному», создание связи «многие-ко-многим», создание связи для сущности-представления.



- **Font & Color** – управление шрифтом, начертанием и цветом различных элементов модели. Позволяет выделить цветом и шрифтом какие-либо смысловые части модели для облегчения восприятия.
- **ModelMart** – панель кнопок специального инструментального средства, предназначенного для связывания пакета BPWin и пакета ERWin.
- **Transforms** – преобразования сущностей (объединение, разделение и т.д.)
- **Drawing** – дополнительные рисуемые элементы (линии, прямоугольники, круги и т.д.)
- **Alignment** – выравнивание элементов модели

Браузер элементов модели помогает ориентироваться среди множества элементов модели, выделяя и группируя их. Имеет три вкладки: «**Model**», «**Subject Areas**» и «**Domains**».

Вкладка «**Model**» объединяет все элементы, в том числе элементы вкладок «**Subject Areas**» и «**Domains**». Здесь можно просмотреть список всех сущностей, связей, узнать, какая сущность с какой связана, и каким отношением (родительская/дочерняя), просмотреть созданные правила проверки значений, типы данных и т.д.

Вкладка «**Subject Areas**» содержит список предметных областей. Предметная область – это выборка из полной модели отдельных сущностей и их областей только тех, которые относятся к заданной области. Например, можно из общей информационной модели вуза выбрать только те сущности и связи, которые относятся, например, к деятельности деканата.

Вкладка «**Domains**» позволяет просматривать используемые в модели типы данных для атрибутов.

При создании новой модели пользуются панелью инструментов «**Toolbox**» (Рис. 2).



Рис. 2. Панели Toolbox для логического (а) и физического (б) уровней модели

Рассмотрим элементы этой панели (слева направо по порядку).

Для логического уровня:

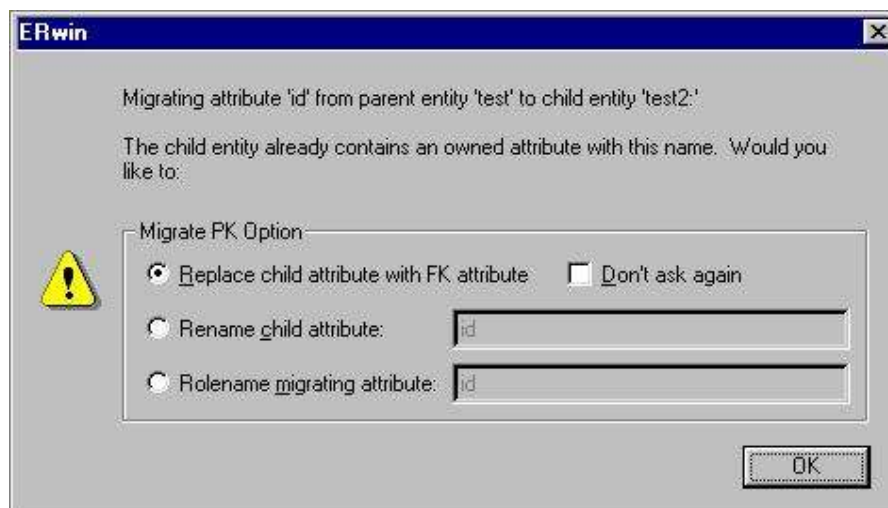
- **Select** – выбор объектов в модели
- **Create entity** – создание новой сущности
- **Create complete sub-category** – создание связки-разветвления
- **Identifying relationship** – создание идентифицирующей связи «один-ко-многим».
- **Many-to-many relationship** – создание связи «многие-ко-многим»
- **Non-identifying relationship** – создание неидентифицирующей связи «один-ко-многим».

Для создания сущности надо выбрать соответствующую кнопку , после чего щелкнуть мышью на свободном месте. Создастся новая сущность , при этом активизируется ввод ее имени. После ввода имени и нажатия Enter выделение переходит в часть ввода ключевых атрибутов. Имена атрибутов также набираются текстом. После ввода имени ключевого атрибута можно либо клавишей Enter добавить еще один ключевой атрибут, либо клавишей Tab перейти к вводу неключевых атрибутов.

Для создания связи между сущностями надо выбрать требуемый тип связи и щелкнуть последовательно на родительной и дочерней сущности. Для связей «многие-ко-многим» порядок щелканья безразличен. При создании связи «один-ко-многим» ключевые атрибуты родительской таблицы переходят в качестве внешних ключей в дочернюю таблицу. Если в дочерней таблице уже есть атрибуты с таким именем, пользователю предложат сделать выбор из следующих вариантов:

- Заменить существующий атрибут. В дочерней таблице существовавший атрибут становится внешним ключом для первичного ключа родительской таблице.
- Переименовать атрибут в дочерней таблице. При этом внешний ключ, приходящий из дочерней таблице, остается неизменным, меняется имя дочернего атрибута, совпадающее с внешним ключом.
- Переименовать атрибут, переходящий из родительской таблицы. При этом в родительской таблице атрибут свое имя не меняет, в дочерней таблице появляется новый атрибут с указанным именем.

При разработке БД чаще всего применяется именно третий вариант – переименование мигрирующего атрибута.



Для физического уровня панель инструментов содержит следующие кнопки:

- Select** – выбор объектов в модели
- Create entity** – создание новой сущности
- View table** – создание сущности-представления
- Identifying relationship** – создание идентифицирующей связи «один-ко-многим».

- ***View relationship*** – создание связи для сущности-представления.
- ***Non-identifying relationship*** – создание неидентифицирующей связи «один-ко-многим».

Сущность-представление – это виртуальная сущность, получаемая выборкой атрибутов из существующих сущностей. В терминах баз данных соответствует запросу.

Использование этих инструментов производится также, как в логической модели.

## **Практическая работа №4.**

### **Разработка информационной модели для реализации каталога документов**

Рассмотрим задачу разработки информационной модели для реализации каталога документов.

#### **Формулировка задачи**

Необходимо создать информационную модель библиотечного каталога. Обеспечить хранение сведений об авторах, книгах и издательствах. Обеспечить получение информации из базы данных в следующих видах:

- 1) перечень книг, изданных в указанном издательстве;
- 2) список книг указанного автора;
- 3) перечень издательств с указанием количества книг данного издательства, имеющихся в библиотеке.

Примеры представления информации приведены ниже:

#### **Порядок выполнения работы**

Общий порядок разработки информационной модели можно представить следующим образом:

- 1) Выделение сущностей (таблиц);
- 2) Выделение связей между таблицами;
- 3) Формализация модели и устранение избыточности.

Обратите внимание на последний этап. Этот этап во многом повторяет первый и второй этапы и может показаться лишним. На данном этапе в модели выделяются избыточные элементы, и модель модифицируется таким образом, чтобы минимизировать избыточность.

Первичный анализ постановки задачи позволяет выделить следующие сущности: книга, издательство, автор. Эти сущности предварительно имеют следующие атрибуты:

Сущность «Книга»:

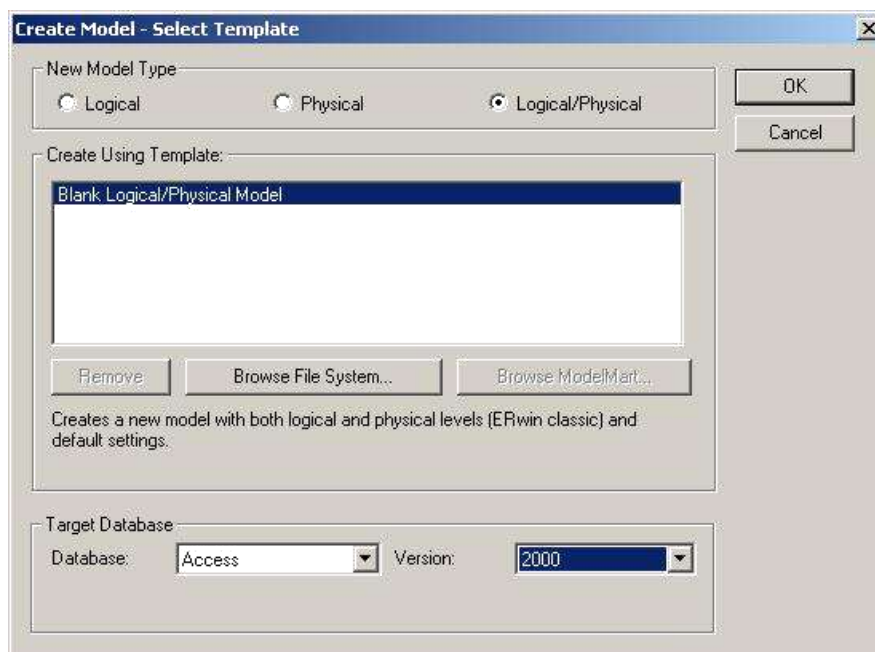
- авторы;
- название;
- издательство;
- год издания;
- жанр книги;
- прочие атрибуты.

Сущность «Автор»:

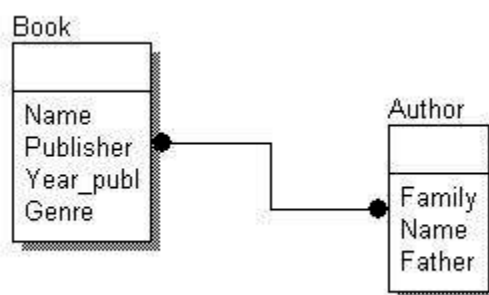
- ФИО автора;

Так как должна быть взаимосвязь между книгами и авторами, то между этими сущностями появляется связь. Она имеет следующие особенности: книга обязательно имеет одного или нескольких авторов, один и тот же автор может написать несколько книг. То есть напрашивается связь типа «многие-ко-многим».

Сформируем в среде пакета ERWin модель в соответствии с полученными результатами. Для этого запускаем ERWin, и создаем новую модель. В качестве типа модели указываем «*Logical/Physical*», сервер – Access 2000.



Далее на логическом уровне создаем нашу модель. Для создания модели воспользуемся панелью инструментов «*Toolbox*». В готовом виде должна получиться примерно такая картинка (используем названия на английском языке):



Проанализируем полученный результат. Можно выявить следующие недостатки:

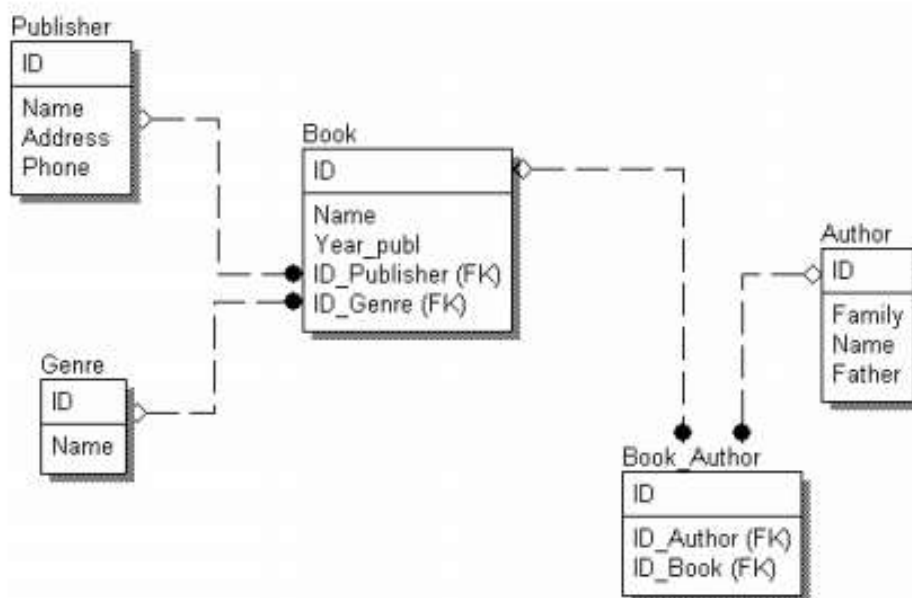
- 1) Если атрибут «жанр» рассматривать как текст, то для многих книг это поле будет иметь одно и то же значение. Например, все учебники или детективные романы. В то же время самих жанров книг относительно немного – их количество измеряется десятками, в то время как количество самих книг может исчисляться десятками и сотнями тысяч;
  - 2) То же самое можно сказать про атрибут «Издательство» – самих издательств гораздо меньше, чем публикуемых ими книг;
  - 3) Присутствует связь «многие-ко-многим». Большинство современных СУБД не поддерживают в чистом виде реализацию связи данного типа.
- Следовательно, полученная модель требует оптимизации.

Оптимизация осуществляется выделением атрибута «Жанр» в отдельную сущность. При этом возникает связь со следующими требованиями: одна книга может относиться только к одному жанру, в одном жанре могут быть написаны множество книг. Этим требованиям отвечает связь типа «один-ко-многим». Аналогично выполняется выделение сущности «Издательство».

Далее, связь типа «многие-ко-многим» формализуется введением дополнительной сущности «автор книги» и заменой связи «многие-ко-многим» двумя связями «один-ко-многим».

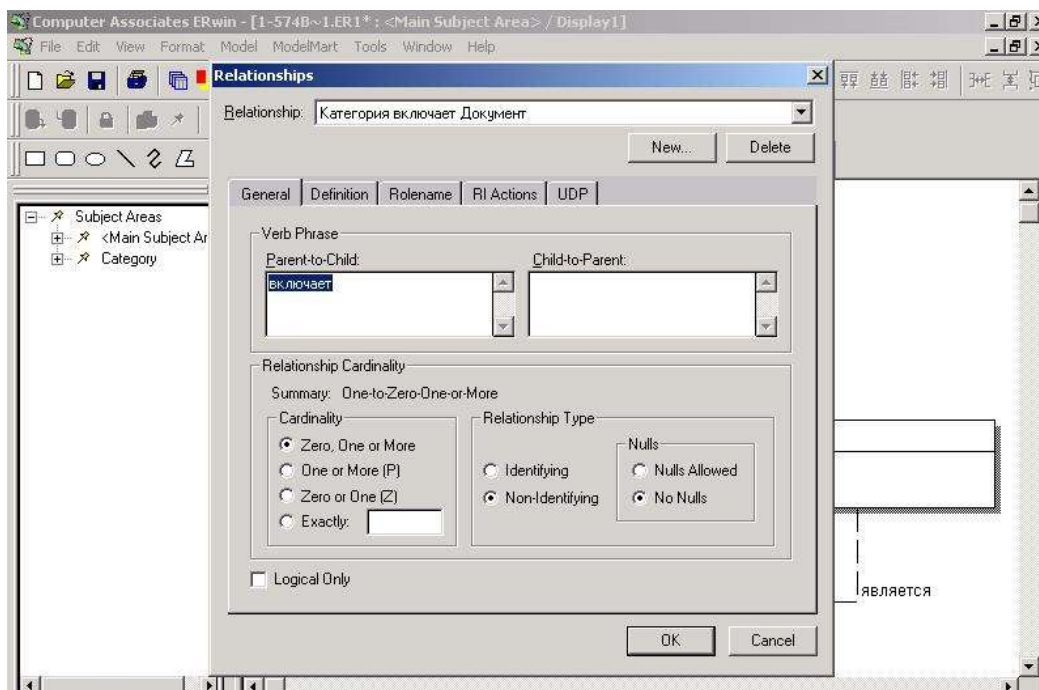
При связывании таблиц удобно задавать специальные поля-идентификаторы (первичные ключи), которые будут использоваться для перекрестных ссылок между таблицами. Такие связи рекомендуется делать неидентифицирующими.

В результате получаем такую модель:



Полученная модель уже близка к корректной модели, которую можно после доработки реализовывать с использованием какой-либо СУБД.

Далее, после формирования сущностей и связей между ними следует настроить все свойства связей. Именно свойствами связей, как правило, задаются ограничения целостности данных для готовой БД. Пользователь в любой момент может изменить тип связи, используемой в модели. Это достигается двойным щелчком на связи, и в появившемся диалоговом окне на вкладке «**General**» выбирается тип связи (область «**Relationship Cardinality**»).



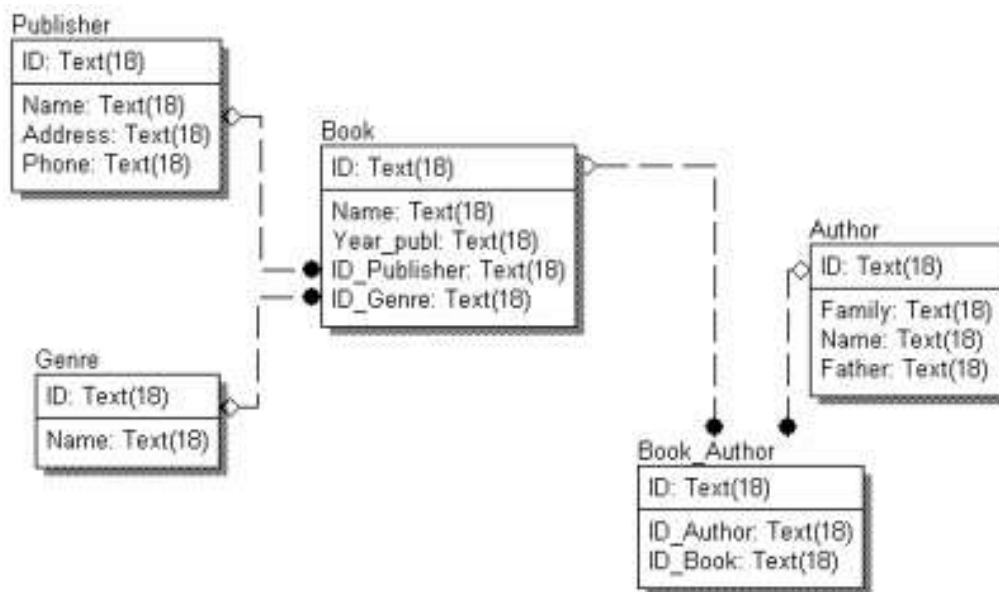
Здесь есть следующие варианты:

- Zero, One or More** – связь «Ноль, один или много»
- One or More** – связь «Один или много»
- Zero or One** – связь «Ноль или один»
- Exactly** – связь «Точно указанное количество»

Далее в секции «**Relationship Type**» задается характер связи – идентифицирующая или неидентифицирующая. В первом случае первичный ключ родительской таблицы входит составной частью в первичный ключ дочерней таблицы, во втором случае – не входит. Если связь задана неидентифицирующая, можно дополнительно указать допустимость пустых значений (NULL values). Если пустые значения не допускаются, при всех операциях с базой данных в этом поле обязательно должно быть какое-то значение. Значение «NULL» не является нулем, пустой строкой и т.д. Оно говорит о том, что в данном поле вообще отсутствует какое-либо значение.

После того, как общий вид связей определен, следует переходить к редактированию физической модели. Для этого в панели инструментов выбирается из выпадающего списка вид модели «**Physical**».

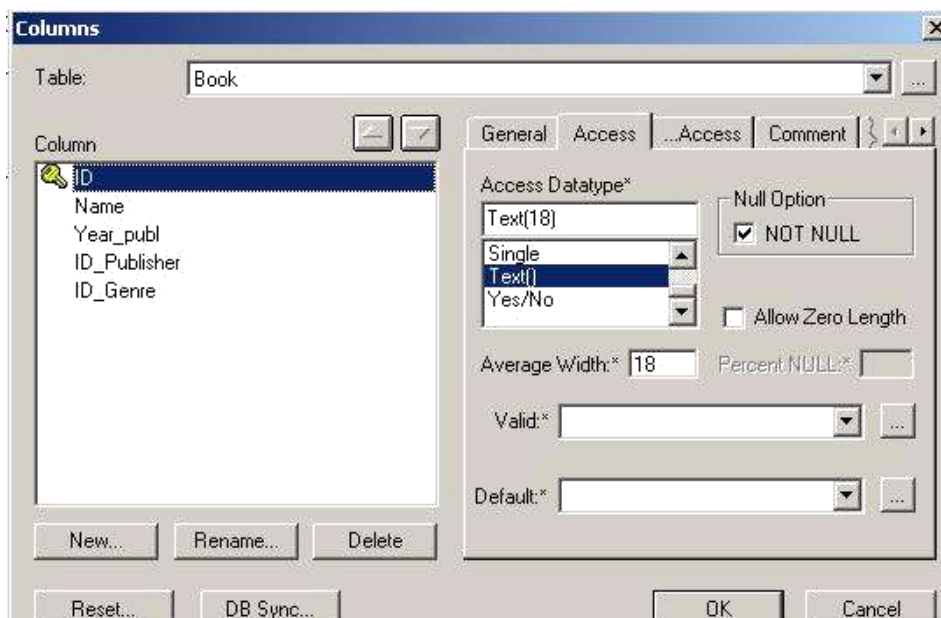
Модель приобретает вид, как на рисунке:



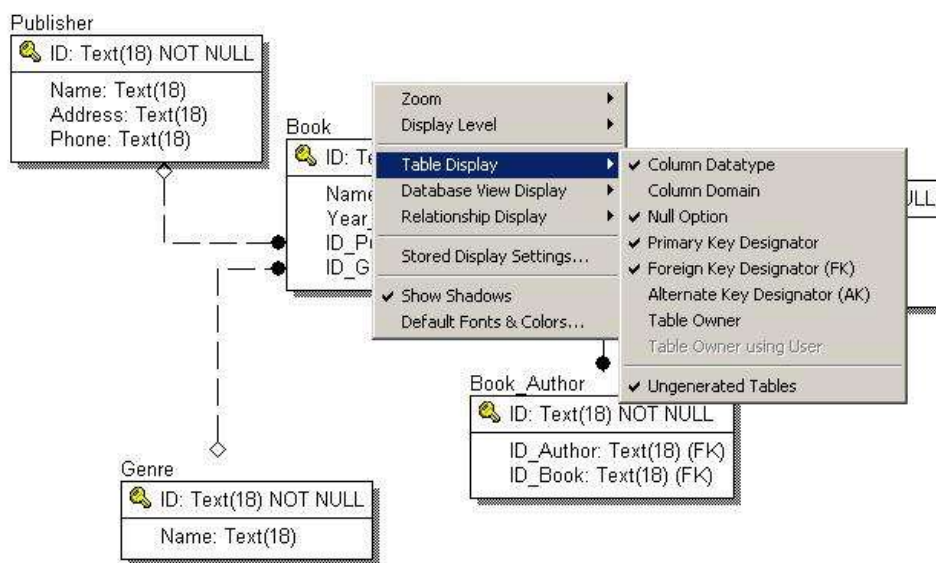
На физическом уровне модель выглядит так, как она должна быть реализована средствами СУБД. При модификации физической модели рекомендуется придерживаться английских наименований атрибутов и сущностей. Хотя СУБД MS Access поддерживает русские наименования атрибутов и сущностей, желательно использовать английские. Это связано с тем, что далеко не все приложения, работающие с разработанной БД, смогут корректно обработать русские наименования полей и таблиц.

Для переименования атрибутов необходимо сделать двойной клик на сущности, и выбирая атрибут из списка, кнопкой «Rename» вызывать диалог переименования атрибута. Кроме того, после переименования следует проверить значения типа данных для каждого атрибута. Это делается на вкладке «Access» в правой части окна. Следует обратить

внимание, что предлагаемые типы данных зависят от типа данных, установленного для атрибута на вкладке «*General*». Если при создании модели или в меню «*Database – Choose Database*» был указан другой тип сервера, то вкладка с типами данных атрибутов (обычно она идет сразу после вкладки *General*) будет иметь имя выбранного типа базы данных. Например, при выборе типа базы SQL Server, эта вкладка будет называться «*SQL Server*».



Следующий шаг – проверка типов атрибутов. Следует внимательно проверить все типы данных, используемых для атрибутов, а также допустимость пустых значений. Для облегчения восприятия рекомендуется включить отображение дополнительной информации об атрибутах. Для этого надо щелкнуть правой кнопкой мыши на свободном месте модели (где нет атрибутов и связей), в появившемся меню выбрать пункт «*Table Display*», и далее проставить галочки напротив пунктов «*Column Datatype*», «*Null Option*», «*Primary Key Designator*», «*Foreign Key Designator*».



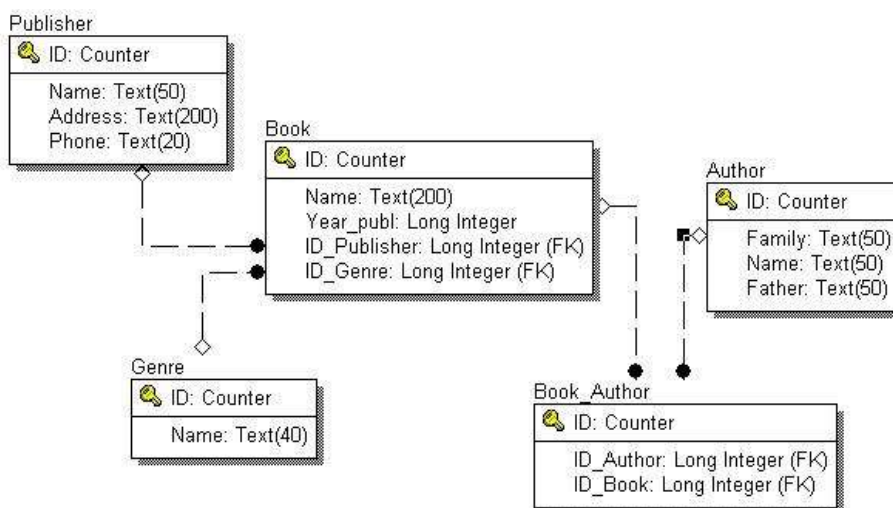
Для всех первичных ключей следует установить тип данных «*Counter*» (в русской версии Access этот тип называется «Счетчик»). Это означает, что для данного поля будут



автоматически генерироваться новые уникальные значения при каждом добавлении новой записи.

Кроме того, следует внимательно проверить допустимость пустых значений для каждого атрибута. Там, где такие значения являются недопустимыми, следует явно указать признак «**NOT NULL**». Пустое значение – это не ноль, не пустая строка или любое другое значение! Значение NULL указывает, что для данного атрибута вообще не задано какое-либо значение.

После этого модель на экране выглядит следующим образом:



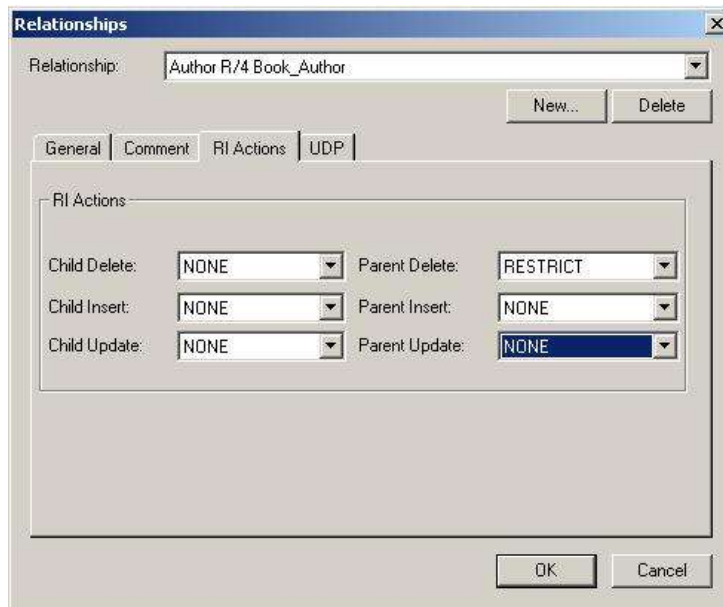
Перед завершением работы над моделью следует проверить действия сервера при различных действиях с БД: добавлении, удалении, изменении записей. Для этого следует открыть окно свойств связи (двойным щелчком на линии связи) и на вкладке «**RI Actions**» установить требуемые ограничения.

Здесь можно установить следующие действия:

- NONE** – ничего не делать;
- RESTRICT** – при наличии связанных записей действие запрещено;
- CASCADE** – действие распространяется на связанные записи;
- NO ACTION** – аналогично NONE;
- SET NULL** – установить значение внешнего ключа в NULL.

Например, для связи между таблицами «**Book**» и «**Book\_Author**» можно установить в качестве действия на «**Parent Delete**» значение «**CASCADE**». В переводе на человеческий язык это означает, что при удалении книги автоматически будут удалены все связи между этой книгой и ее авторами.

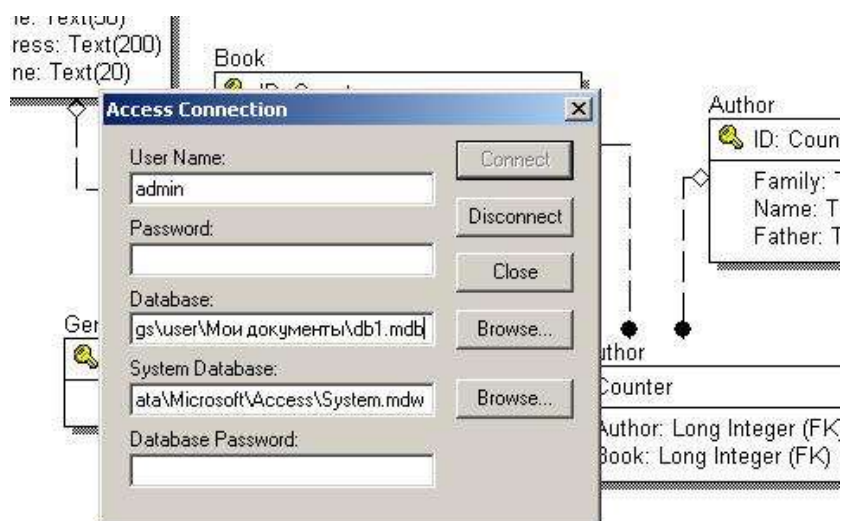
С другой стороны, для аналогичной связки между «**Author**» и «**Book\_Author**» для того же действия (**Parent Delete**) следует указать «**RESTRICT**», то есть нельзя удалить автора, пока в библиотеке имеется хотя бы одна книга этого автора.



Подготовленную таким образом модель уже можно транслировать в СУБД Access. Для этого необходимо выполнить следующие шаги: выбрать базу данных, в которой будет создаваться структура данных, настроить опции генерации таблиц.

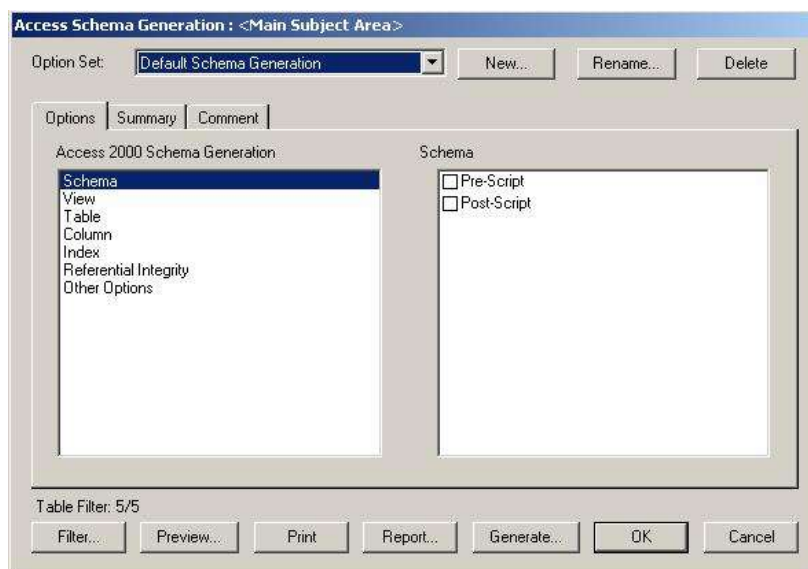
Для выбора базы данных следует в меню «**Database**» выбрать пункт «**Database Connection**». В появившемся диалоге надо указать имя пользователя и пароль для доступа к БД, имя БД пользователя, имя системной БД, и также пароль для доступа к системной БД.

Для работы с Access следует выставить те параметры, которые приведены на рисунке. Расположение системной БД (System Database) следует уточнить у преподавателя.



Предполагается, что тип сервера задан изначально при создании модели. Если это не так, следует выбрать в меню «**Database**» пункт «**Choose Database**». В появившемся диалоге настраивается тип сервера данных.

После настройки соединения с базой данных подготовленную модель данных можно транслировать на сервер БД. Для этого в меню «**Tools**» выбирается пункт «**Forward Engineering/Schema Generation...**».



В этом окне указываются опции генерации структуры данных. Если таблицы уже существуют, в разделе «*Table*» ставится галочка «*Drop Table*». Это означает, что все существующие таблицы с совпадающими именами будут удалены (естественно, со всеми данными). Поэтому этой опцией следует пользоваться с осторожностью.

После завершения всех настроек после нажатия кнопки «*Generate*» на сервере БД будет создана структура данных по модели. При необходимости предварительного контроля нажатие кнопки «*Preview*» позволяет увидеть SQL-код, который создает все требуемые таблицы.

## **Практическая работа №5. Разработка информационной модели**

### **Задание**

Пользуясь пакетом ERWin, по варианту задания сформировать модель данных для реализации БД с указанными требованиями. Структура связей между таблицами определяется правилами: один-ко-многим или многие-ко-многим.

При выполнении работы обеспечить соблюдение следующих требований:

- корректность модели данных;
- полнота информации, представляемой в модели данных. При необходимости студент самостоятельно дополняет модель требуемыми атрибутами и сущностями для адекватного представления рассматриваемой задачи;
- безизбыточность информации, представляемой в модели данных;

После завершения моделирования разработанная структура данных транслируется средствами пакета ERWin в базу данных MS Access.

### **Варианты заданий**

#### **1. Налоговая инспекция Таблицы:**

- налогоплательщик: ФИО, адрес, категория, ...
- налоги: наименование, процентная ставка, дата и размер платежа, ...
- льготы: вид льготы, сумма льготы, процентная ставка, ...

Правила:

- каждый налогоплательщик платит несколько разновидностей налогов – каждый налогоплательщик может иметь несколько льгот

#### **2. Земельный реестр**

Таблицы:

- землевладелец: наименование, адрес, ...
- участок земли: расположение, площадь, категория земельного участка, ...
- строения на участке земли: наименование, площадь, стоимость, ... Правила:
  - один землевладелец может иметь несколько участков, одним участком может распоряжаться несколько землевладельцев;
  - на каждом участке может быть несколько строений;

#### **3. Бухгалтерия**

Таблицы:

- контрагенты: плательщики и получатели – наименование, адрес, банк, ...
- платежи: дата, сумма, контрагент, ...

Правила:

- каждый контрагент может выполнять несколько платежей
- в одном банке может быть зарегистрировано несколько контрагентов

#### **4. Банк**

Таблицы:

- счета: ФИО или наименование клиента, адрес, другие сведения, ...
  - операции по счету: зачисление или снятие средств, начисление процентов, ...
  - вклады: наименование, срок вклада, порядок начисления процентов, ...
- Правила:
- каждый клиент может иметь несколько вкладов
  - с каждым счетом выполняется множество операций

#### **5. Инвестиционный фонд**

Таблицы:

- объект инвестиций: наименование, срок инвестиций, срок возврата, процент, ...
- фирма-получатель инвестиций: наименование, адрес, ...

Правила:

- фонд одновременно может инвестировать несколько проектов – фонд получает платежи в несколько приемов – фонд может осуществлять инвестирование в несколько этапов
- одна фирма может получать инвестиции на несколько различных объектов.

#### **6. Лизинговая компания**

Таблицы:

- договор лизинга: дата, номер, контрагент, наименование оборудования, стоимость оборудования, размер выплат, ...
- контрагент, взявший оборудование в лизинг;
- выплата по договору лизинга: дата, сумма, договор, контрагент, ...

Правила:

- один агент может заключить несколько договоров лизинга
- по каждому договору выплаты осуществляются в несколько приемов

#### **7. Отдел кредитования магазина «Электроника»**

Таблицы:

- покупатель: ФИО, адрес, место работы, паспортные данные, ...
- кредит: вид кредита, сумма, срок полной выплаты, ...

Правила:

- по кредиту выплаты производятся в несколько приемов
- один покупатель может сделать несколько покупок в кредит

#### **8. Отдел начисления зарплаты**

Таблицы:

- сотрудники: ФИО, отдел, должность, размер зарплаты, ...

- выплаты: дата, сумма, за какой месяц, дополнительные выплаты, ... Правила:
- каждому сотруднику зарплата платится ежемесячно – каждая зарплата складывается из нескольких выплат

## **9. Отдел валютных операций банка**

Таблицы:

- текущий курс валюты – кросс-курсы валют – операции с валютой

Правила:

- каждая валюта имеет свой курс относительно рубля
- пересчет между разными валютами осуществляется с использованием соответствующего кросс-курса

## **10. Рекламное агентство**

Таблицы:

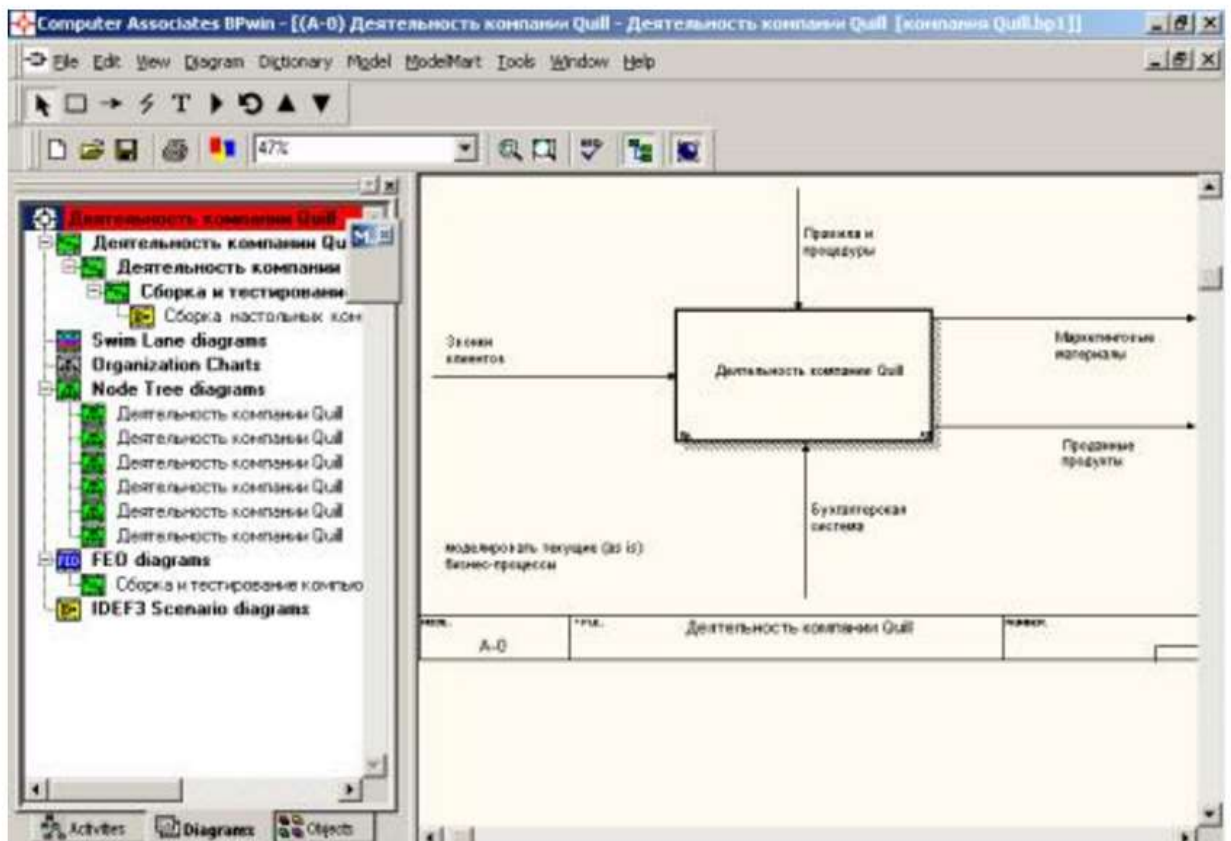
- рекламодатель: наименование, адрес, какие виды рекламы заказывает, ...
- договор рекламы: дата, номер, сумма, правила размещения рекламы, ...
- размещение рекламной информации: вид, дата, параметры, ... Правила:

- несколько рекламных объявлений по одному договору
- различные виды рекламы (радио, ТВ, баннеры, газетная реклама)

## Практическая работа №6. Изучение интерфейса инструментальной среды BPWin.

BPwin имеет достаточно простой и интуитивно понятный интерфейс пользователя. При запуске BPwin по умолчанию появляется основная панель инструментов, палитра инструментов (вид которой зависит от выбранной нотации) и, в левой части, навигатор модели — Model Explorer

BPwin поддерживает три методологии — IDEF0, IDEF3 и DFD, каждая из которых решает свои специфические задачи. В BPwin возможно построение смешанных моделей, т. е. модель может содержать одновременно диаграммы как IDEF0, так и IDEF3 и DFD. Состав палитры инструментов изменяется автоматически, когда происходит переключение с одной нотации на другую.



При создании новой модели возникает диалог, в котором следует указать, будет ли модель создаваться заново, или она будет открыта из файла либо из репозитория Model Mart. Также необходимо внести имя модели и выбрать методологию, в которой будет построена модель (рис.3).

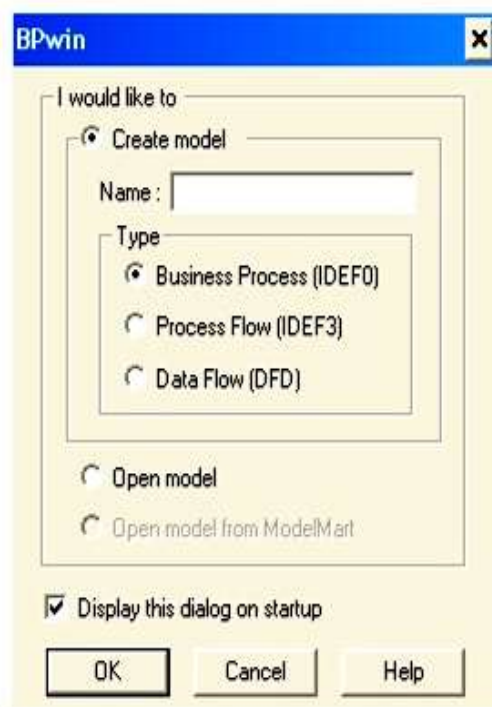


Рис.3 Диалог создания модели.

BPwin поддерживает три методологии моделирования:

- функциональное моделирование (IDEFO);
- описание бизнес-процессов (IDEF3);
- диаграммы потоков данных (DFD).











В зависимости от выбранной методологии программой автоматически подбирается нужная панель инструментов BPwin Toolbox. В BPwin существует три разных панели инструментов — по числу поддерживаемых программой методологий. На рис.4 представлена палитра для IDEF0.



При желании можно скрывать панель инструментов, используя функцию «View» на панели инструментов.

Описание элементов управления основной панели инструментов:



Элемент управления	Описание	Соответствующий пункт меню
	Создать новую модель	File/New
	Открыть модель	File/Open
	Сохранить модель	File/Save
	Напечатать модель	File/Print
	Вызвать генератор отчетов	Tools/Report Builder
	Выбор масштаба	View/Zoom
	Масштабирование	View/Zoom
	Проверка правописания	Tools/Spelling
	Включение и выключение навигатора модели Model Explorer	View/Model Explorer
	Включение и выключение дополнительной панели инструментов работы с ModelMart	ModelMart

Пункты контекстного меню Font и Color вызывают диалог Arrow Properties или Activity Properties для установки шрифта (в том числе его размера и стиля) и цвета объекта. В нижней части вкладки Font диалогов Arrow Properties и Activity Properties находятся группа опций Apply setting to, позволяющих изменить шрифт для всех работ или стрелок на текущей диаграмме, в модели, и группа Global, позволяющая изменить шрифт одновременно для всех объектов модели.

VRwin позволяет установить шрифт по умолчанию для объектов определенного типа на диаграммах и в отчетах. Для этого следует выбрать меню Model/Default Fonts, после чего появляется каскадное меню, каждый пункт которого служит для установки шрифтов для определенного типа объектов:

Context Activity - работа на контекстной диаграмме;

Context Arrow - стрелки на контекстной диаграмме;

Decomposition Activity - работы на диаграмме декомпозиции;

Decomposition Arrow - стрелки на диаграмме декомпозиции;

Node Tree Text - текст на диаграмме дерева узлов;

Frame User Text - текст, вносимый пользователем в каркасе диаграмм;

Frame System Text — системный текст в каркасе диаграмм;

Text Blocks - текстовые блоки;

Parent Diagram Text - текст родительской диаграммы;

Parent Diagram Title Text - текст заголовка родительской диаграммы;

Report Text - текст отчетов.

## **Практическая работа №7.**

### **Разработки функциональной модели для реализации бизнес-плана банка по привлечению и размещению ресурсов**

#### **Теоретические сведения**

Для описания логики взаимодействия информационных потоков более подходит IDEF3, называемая также WorkFlow Diagramming – методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. Диаграммы WorkFlow могут быть использованы в моделировании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например, последовательность обработки заказа события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

IDEF3 – это метод, имеющий основной целью дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе.

Техника описания набора данных IDEF3 является частью структурного анализа. В отличие от некоторых методик описаний процессов, IDEF3 не ограничивает аналитика чрезмерно жесткими рамками синтаксиса, что может привести к созданию неполных или противоречивых моделей.

IDEF3 может быть также использован как метод создания процессов. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа.

Каждая работа в IDEF3 описывает какой-либо сценарий бизнес процесса и может являться составляющей другой работы. Поскольку сценарий описывает цель и рамки модели, важно, чтобы работы именовались отглагольным существительным, обозначающим процесс действия, или фразой, содержащей такое существительное.

Точка зрения на модель должна быть задокументирована. Обычно это точка зрения человека, ответственного за работу в целом. Также необходимо задокументировать цель модели – те вопросы, на которые призвана ответить модель.

Единицы работы – Unit of Work (UOW). UOW, также называемые работами (activity), являются центральными компонентами модели. В IDEF3 работы изображаются прямоугольниками с прямыми углами и имеют имя, выраженное отглагольным существительным, обозначающим процесс действия, одиночным или в составе фразы, и номер (идентификатор); другое имя существительное в составе той же фразы обычно отображает основной выход (результат) работы (например, «Изготовление изделия»). Часто имя существительное в имени работы меняется в процессе моделирования, поскольку модель может уточняться и редактироваться. Идентификатор работы присваивается при создании и не меняется никогда. Даже если работа будет удалена, ее идентификатор не будет вновь использоваться для других работ. Обычно номер работы состоит из номера родительской работы и порядкового номера на текущей диаграмме.

Работа в IDEF3 требует более подробного описания, чем работа в IDEF0. Каждая UOW должна иметь ассоциированный документ, который включает текстовое описание компонентов работы: объектов (Objects) и фактов (Facts), связанных с работой, ограничений (Constraints), накладываемых на работу, и дополнительное описание работы (Description). Эта информация заносится во вкладку UOW диалога Activity Properties.

**Связи.** Связи показывают взаимоотношения работ. Все связи в IDEF3 однонаправленны и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо. В IDEF3 различают три типа стрелок,

изображающих связи, стиль которых устанавливается во вкладке Style диалога Arrow Properties (пункт контекстного меню Style).


Старшая (Precedence) стрелка – сплошная линия, связывающая единицы работ (UOW). Рисуеться слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется.

Стрелка отношения (Relational Link) – пунктирная линия, используемая для изображения связей между единицами работ (UOW), а также между единицами работ и объектами ссылок.

Потоки объектов (Object Flow) – стрелка с двумя наконечниками, применяется для описания того факта, что объект используется в двух или более единицах работы, например, когда объект порождается в одной работе и используется в другой.

**Старшая связь и поток объектов.** Старшая связь показывает, что работа-источник заканчивается ранее, чем начинается работа-цель. Часто результатом работы-источника становится объект, необходимый для запуска работы-цели. В этом случае стрелку, обозначающую объект, изображают с двойным наконечником. Имя стрелки должно ясно идентифицировать отображаемый объект. Поток объектов имеет ту же семантику, что и старшая стрелка.

Отношение показывает, что стрелка является альтернативой старшей стрелке или потоку объектов в смысле задания последовательности выполнения работ - работа-источник не обязательно должна закончиться прежде, чем работа-цель начнется. Более того, работа-цель может закончиться прежде, чем закончится работа-источник.

**Перекрестки (Junction).** Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны завершены перед началом следующей работы. Различают перекрестки слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для ветвления. Для внесения перекрестка служит кнопка  (добавить на диаграмму перекресток – Junction) в палитре инструментов. В диалоге Junction Type Editor необходимо указать тип перекрестка. Смысл каждого типа приведен в табл. 1.

Все перекрестки на диаграмме нумеруются, каждый номер имеет префикс J. Можно редактировать свойства перекрестка при помощи диалога Junction Properties (вызывается из контекстного меню). В отличие от IDEF0 и DFD в IDEF3 стрелки могут сливаться и разветвляться только через перекрестки.







**Объект ссылки.** Объект ссылки в IDEF3 выражает некую идею, концепцию или данные, которые нельзя связать со стрелкой, перекрестком или работой. Для внесения объекта ссылки служит кнопка  (добавить в диаграмму объект ссылки – Referent) в палитре инструментов.

Таблица 1 - Типы перекрестков

Обозначение	Наименование	Смысл в случае слияния стрелок	Смысл в случае разветвления стрелок
	Асинхронное «И»	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Синхронное «И»	Все предшествующие процессы	Все следующие процессы

		завершены одновременно	запускаются одновременно
	Асинхронное «ИЛИ»	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Синхронное «ИЛИ»	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно
	Исключающее «ИЛИ»	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Объект ссылки изображается в виде прямоугольника, похожего на прямоугольник работы. Имя объекта ссылки задается в диалоге Referent Properties (пункт контекстного меню Name), в качестве имени можно использовать имя какой-либо стрелки с других диаграмм или имя сущности из модели данных. Объекты ссылки должны быть связаны с единицами работ или перекрестками пунктирными линиями. Официальная спецификация IDEF3 различает три стиля объектов ссылок – безусловные (unconditional), синхронные (synchronous) и асинхронные (asynchronous). BPwin поддерживает только безусловные объекты ссылок. Синхронные и асинхронные объекты ссылок, используемые в диаграммах переходов состояний объектов, не поддерживаются.

При внесении объектов ссылок помимо имени следует указывать тип объекта ссылки. Типы объектов ссылок приведены в табл. 2.

Таблица 2 - Типы объектов ссылок

Тип объекта ссылки	Цель описания
ОБЪЕКТ	Описывает участие важного объекта
GOTO	Инструмент циклического перехода (в повторяющейся последовательности работ), возможно на текущей диаграмме, но не обязательно. Если все работы цикла присутствуют на текущей диаграмме, цикл может также изображаться и стрелкой, возвращающейся на стартовую работу. GOTO может ссылаться на перекресток
UOB (Unit of Behavior)	Применяется, когда необходимо подчеркнуть множественное использование какой-либо работы, но без цикла. Например, работа «Контроль качества» может быть использована в процессе «Изготовление изделия» несколько раз, после каждой единичной операции. Обычно этот тип ссылки не используется для моделирования автоматически запускающихся работ
NOTE	Используется для документирования важной информации, относящейся к каким-либо графическим объектам на диаграмме. Является

	альтернативой внесению текстового объекта на диаграмму
ELAB (Elaboration)	Используется для усовершенствования графиков или их более детального описания. Обычно употребляется для детального описания разветвления и слияния стрелок на перекрестках

**Задание.** Необходимо создать сценарий процесса привлечения и размещения ресурсов банка. Собрать информацию по отделениям банка, проверить и при необходимости скорректировать показатели полученного сводного плана ресурсов, составить планы привлечения и размещения ресурсов по банку в целом и по его отделениям.

### Ход работы

Общий порядок разработки сценария:

1. Выделение действий или подпроцессов моделируемой системы.
2. Определение последовательности выполнения выделенных действий.

Действия моделируемой системы были определены в практической работе 1. Аналогично проведем построение модели. На рисунках 1-3-8 показаны основные этапы.

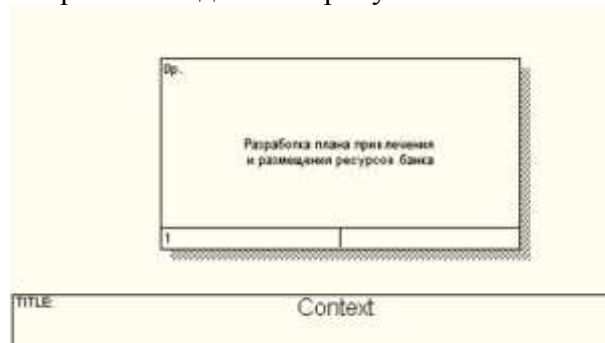


Рисунок 1 - Контекстная диаграмма сценария

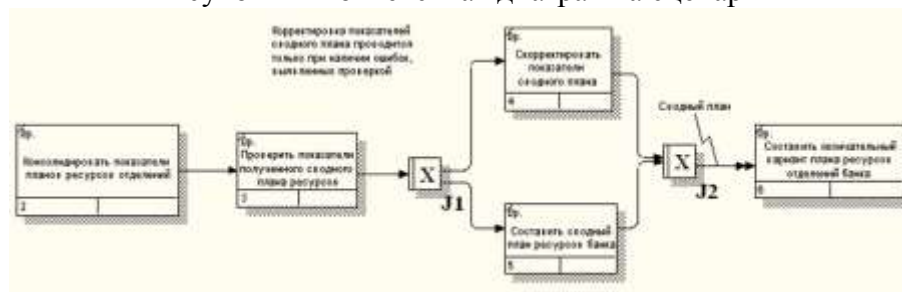


Рисунок 2 - Первый уровень декомпозиции

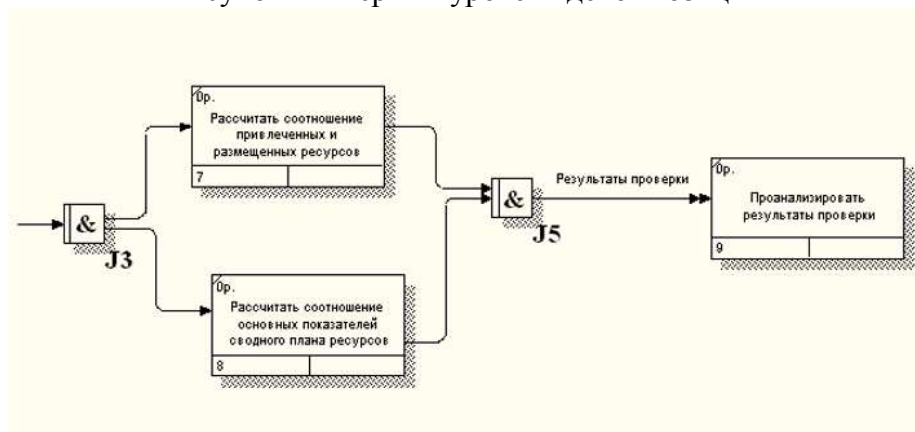


Рисунок 3 - Второй уровень декомпозиции

### Контрольные вопросы

1. Для чего строится диаграмма IDEF3?

2. Чем диаграмма IDEF3 отличается от диаграммы IDEF0?
3. Как графически обозначается работа в диаграмме IDEF3?
4. С какой целью между работами устанавливают перекресток?
5. Какие типы перекрестков вам знакомы?

## **Практическая работа №8. Разработка функциональной модели.**

### **Задание:**

- построить контекстную диаграмму;
- провести функциональную декомпозицию;
- после каждого сеанса декомпозиции провести сеанс экспертизы.

CASE-средство VPwin предназначено для проведения анализа и реорганизации бизнес-процессов. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. Такая технология построения модели позволяет построить модель, адекватную предметной области на всех уровнях абстрагирования. Под технологией проектирования (создания) информационных систем (ИС) понимают упорядоченный в логической последовательности набор методических приемов, технических средств и проектировочных методов, нацеленных на реализацию общей концепции создания или доработки проекта системы и ее компонентов.

Любая модель информационной системы не может быть построена без четко сформулированной цели. Цель должна отвечать на следующие вопросы:

- Почему этот процесс должен быть замоделирован?
- Что должна показывать модель?
- Что может получить заказчик?

Формулировка цели позволяет команде аналитиков сфокусировать усилия в нужном направлении.

Хотя при построении модели учитываются мнения различных людей, модель должна строиться с единой точки зрения. Точку зрения можно представить как взгляд человека, который видит систему в нужном для моделирования аспекте. Точка зрения должна соответствовать цели моделирования. Очевидно, что описание работы предприятия с точки зрения финансиста и технолога будет выглядеть совершенно по-разному, поэтому в течение моделирования важно оставаться на выбранной точке зрения. Как правило, выбирается точка зрения человека, ответственного за моделируемую работу в целом.

В ходе выполнения данной курсовой работы были созданы две модели информационной системы фирмы ООО "Мир Компьютеров": IDEF0, DFD.

На начальных этапах создания ИС необходимо понять, как работает организация. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы



каждого рядового сотрудника. Рядовой сотрудник хорошо знает, что творится на его рабочем месте, но плохо знает, как работают коллеги. Поэтому для описания работы предприятия необходимо построить модель. Такая модель должна быть адекватна предметной области, следовательно, она должна содержать в себе знания всех участников бизнес-процессов организации. Наиболее удобным языком моделирования бизнес-процессов является IDEF0. IDEF0-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения.

В методологии IDEF0 система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной, т.е. функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации. Под моделью в методологии IDEF0 понимают описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее определенные вопросы. Описание системы с помощью методологии IDEF0 называется функциональной моделью. Методология IDEF0 предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма) (рис. 4).

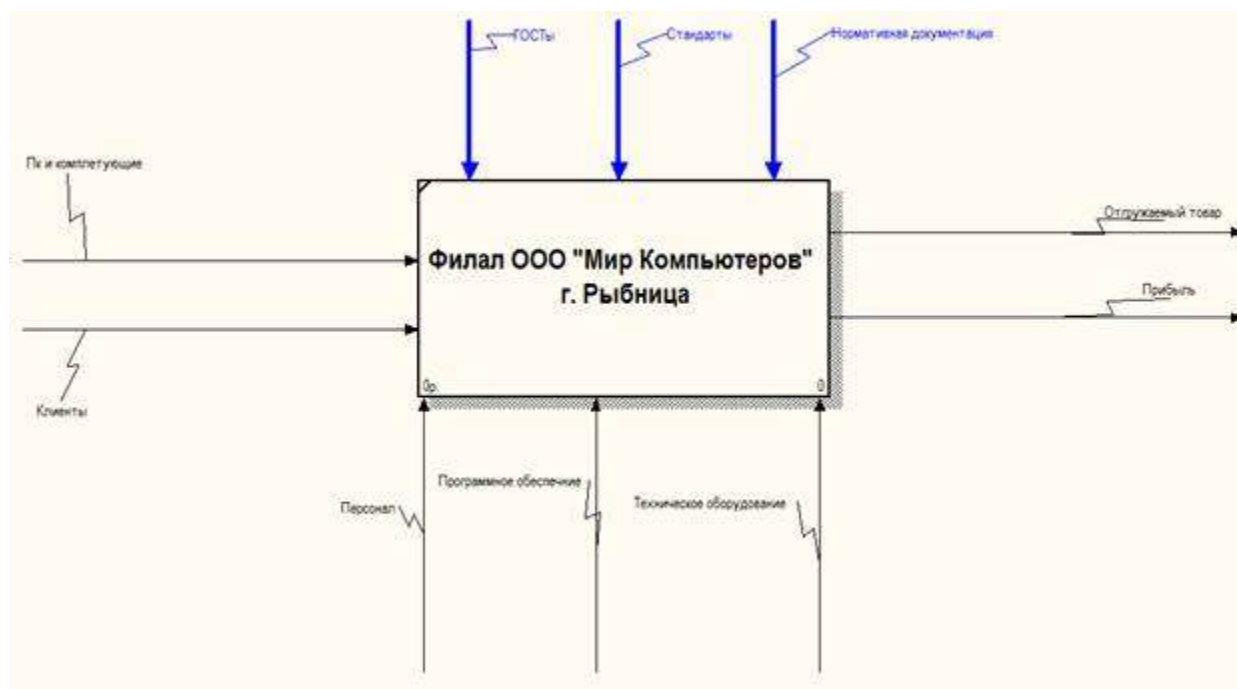


Рис. 4. Контекстная диаграмма фирмы ООО "Мир Компьютеров"

На входе данной информационной системы будут находиться Пк и комплектующие необходимые для продажи и хранения которые непосредственно предназначены для клиентов. На выходе в свою очередь будет находиться реализованная продукция и прибыль

Механизмами ООО "Мир Компьютеров", будут являться персонал фирмы и программно-технические ресурсы. Управление – правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Для данной фирмы управлением будут являться нормативная документация и ГОСТы.

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции.

Была проведена декомпозиция первого уровня информационной системы ООО " Мир Компьютеров " (рис. 5).

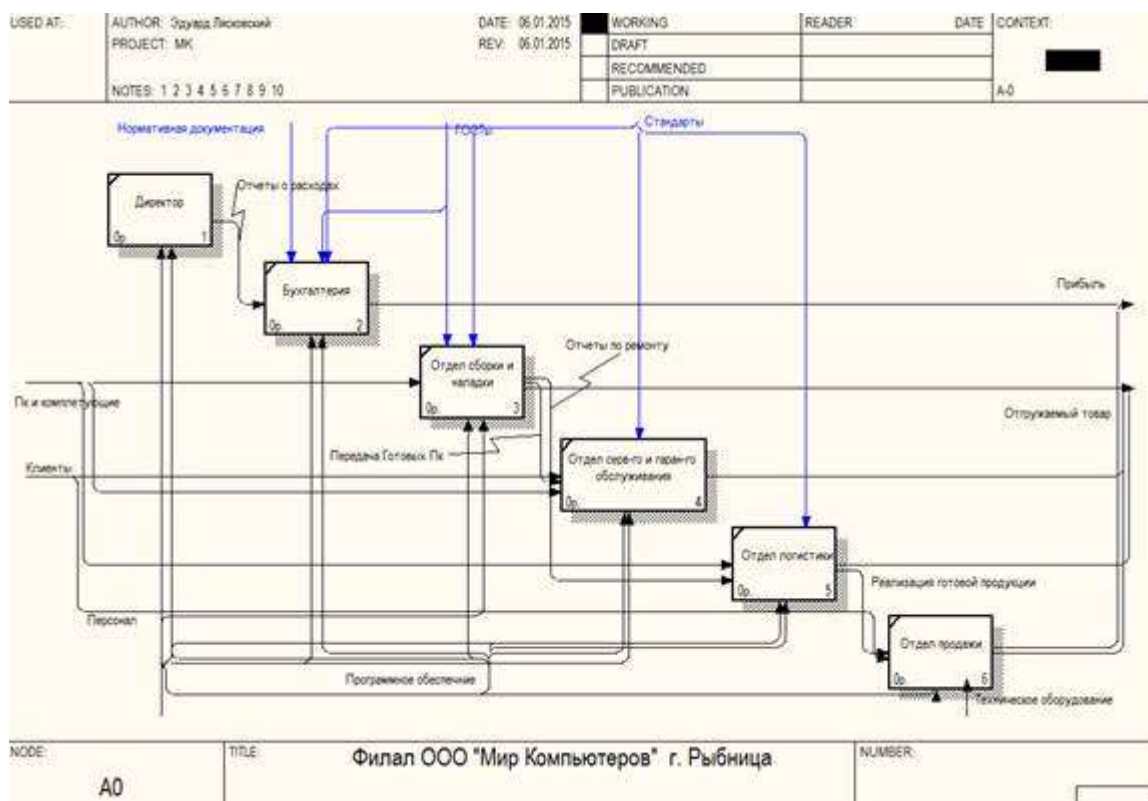


Рис. 5. Диаграмма декомпозиции первого уровня ООО "Мир Компьютеров"

В процессе декомпозиции первого уровня были выделены следующие работы: директор, бухгалтерия, отдел сборки и наладки, отдел сервисного и гарантийного обслуживания, отдел логистики и отдел продаж. Механизмами в данной диаграмме являются программно-технические ресурсы и персонал, управлением – ГОСТы и нормативная документация. На входе данной диаграммы будут находиться ПК и комплектующие, клиенты, на выходе – прибыль, продукция.

Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами (рис.6).

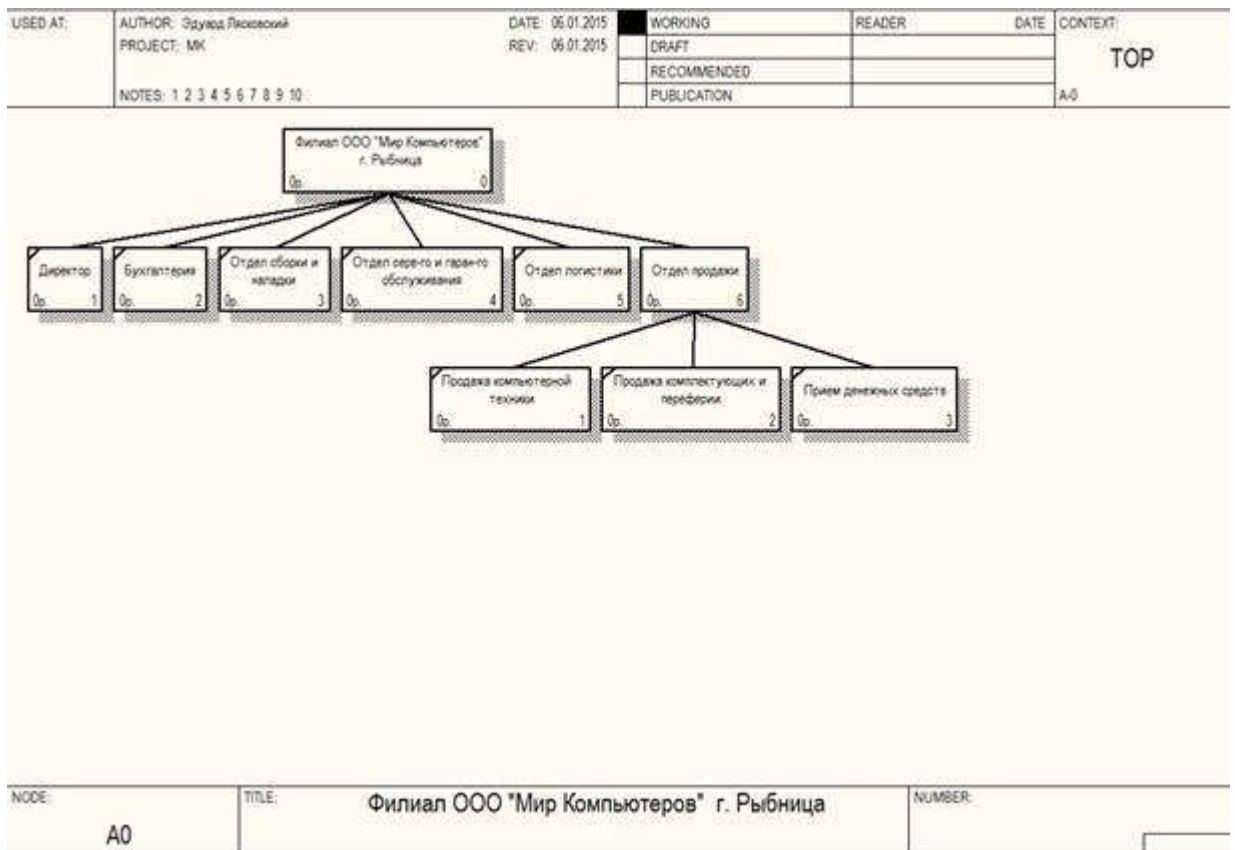


Рис. 6. Диаграмма дерева узлов ООО "Мир Компьютеров"

Диаграмм деревьев узлов может быть в модели сколь угодно много, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня. На данной диаграмме указывается на сколько уровней мы декомпозировали нашу информационную систему, и какие работы включает в себя непосредственной каждый уровень декомпозиции.

Диаграммы потоков данных (Data Flow Diagramming) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота.

Диаграмма DFD описывает:

- функции обработки информации;
- документы, объекты, сотрудников или отделы, которые участвуют в обработке информации;
- внешние ссылки (external references), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- таблицы для хранения документов (хранилище данных, data store).

В данной курсовой работе была создана диаграмма потока данных для фирмы ООО "Мир Компьютеров" (рис. 7).

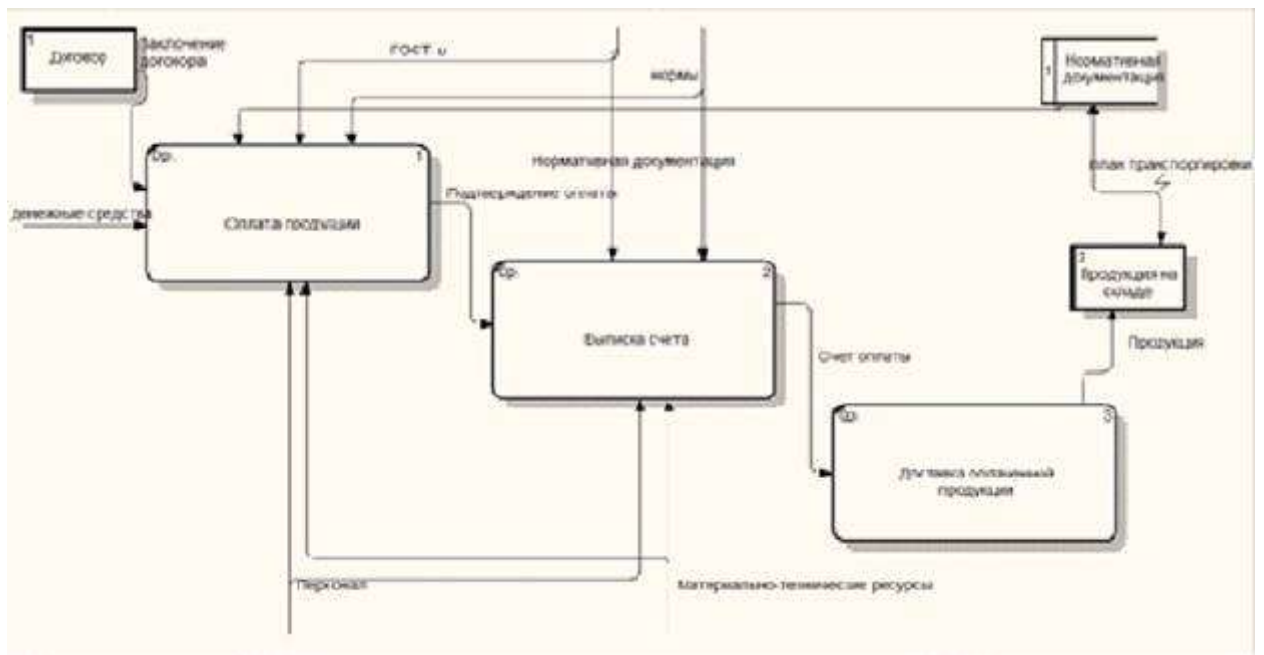


Рис. 7. Диаграмма потоков данных ООО "Мир Компьютеров"

На данной диаграмме описан процесс оплаты товара. Были выделены следующие работы: оплата продукции, выписка счета и доставка оплаченной продукции. На диаграмме присутствуют две внешние ссылки: договор и продукция на складе, которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы. Так же на диаграмме находится хранилище данных, представляющее из себя нормативную документацию фирмы.

В результате использования инструмента VRwin была создана модель информационной системы ООО "Мир Компьютеров", была создана контекстная диаграмма, проведена ее декомпозиция первого уровня, создано дерево узлов и диаграмма потоков данных.

## Практическая работа №9. Выбор вычислительной модели. Выбор конфигурации сервера

В большинстве прикладных систем СУБД можно выделить три логических части: пользовательский интерфейс (средства представления), обеспечивающий функции ввода и отображения данных; некоторую прикладную обработку, характерную для данной предметной области; и собственно сервисы СУБД. Пользовательский интерфейс и прикладная обработка обычно объединены в одном двоичном коде, хотя некоторые из наиболее продвинутых приложений теперь обеспечивают многопоточковую фронтальную обработку, которая отделена от средств представления. Как правило, по мере роста базы данных сервер СУБД реализуется на выделенной системе, чтобы гарантировать минимизацию помех для его работы.

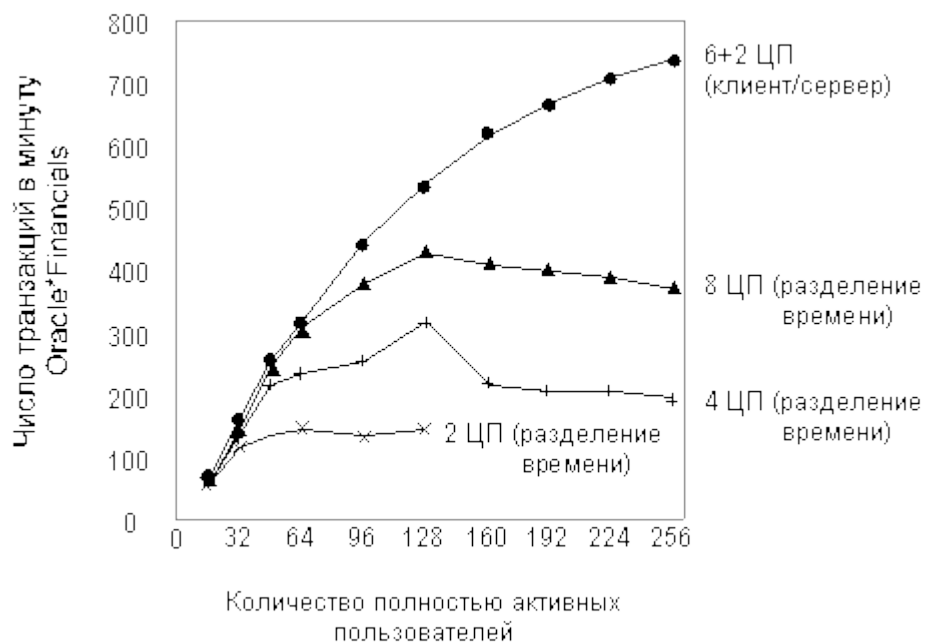


Рис. 2.3. Сравнение модели клиент/сервер с моделью разделения времени при работе прикладной системы Oracle\*Financials.

Если в системе можно реализовать модель вычислений клиент/сервер, отделяющую фронтальную (прикладную) обработку и средства представления от поставщика услуг СУБД, ее использование обычно обеспечивает существенное улучшение общей производительности системы. Такая организация системы позволяет наиболее важному ресурсу (серверу СУБД) работать без помех на своей собственной хост-машине. Это в первую очередь справедливо для систем, в которых работа средств представления связана с управлением сотнями или тысячами терминалов в режиме sbreak. Большинство программ, базирующиеся на curses- (screen-), сегодня используют sbreak-поддержку терминалов.

Режим разделения времени, в отличие от режима клиент/сервер, обычно обеспечивает большую производительность только тогда, когда требования к компоненту представления оказываются очень легкими, или когда одновременная пользовательская нагрузка невелика. Существенно что приложения, в основе компонента представления которых лежат формы, никогда не бывают легковесными. Даже приложения, работающие в диалоговом режиме printf/get обычно значительно более тяжелые, что позволяет оправдать использование конфигураций клиент/сервер. Тест TPC-A определяет возможно наиболее легкое требование приложения/представления (он включает ровно по одному вызову scsf(n) и

printf(n)). Следует отметить, что даже тест TPC-A работает только на 5% быстрее в режиме разделения времени. Некоторые сравнительно недавние исследования компании Sun с использованием Oracle\*Financials и Oracle 7 показали, что 6-процессорный сервер СУБД на базе SPARCserver 1000 с фронтальной системой на базе SPARCstation 10 Model 512 может поддерживать почти на 40% пользователей больше, чем 8-процессорный SPARCserver 1000, работающий в режиме разделения времени (рис. 2.3).

Рекомендации:

- Всегда, если это возможно, следует применять конфигурацию клиент/сервер, если только нагрузка по прикладной обработке и обработке представления являются необычно легкими.
- Где это возможно, собственно сервер СУБД должен работать на выделенной системе.

Если СУБД работает в режиме клиент/сервер, то сеть или сети, соединяющие клиентов с сервером должны быть адекватно оборудованы. К счастью, большинство клиентов работают с вполне определенными фрагментами данных: индивидуальными счетами, единицами хранящихся на складе изделий, историей индивидуальных счетов и т.п. При этих обстоятельствах скорость сети, соединяющей клиентов и серверы редко оказывается проблемой. Отдельной сети Ethernet или Token Ring обычно достаточно для обслуживания 100-200 клиентов. На одном из тестов Sun конфигурация клиент/сервер, поддерживающая более 250 пользователей Oracle\*Financial, генерировала трафик примерно 200 Кбайт/с между фронтальной системой и сервером базы данных. По очевидным причинам стоит более плотно наблюдать за загрузкой сети, особенно когда число клиентов превышает примерно 20 на один сегмент Ethernet. Сети Token Ring могут поддерживать несколько большую нагрузку, чем сети Ethernet, благодаря своим превосходным характеристикам по устойчивости к деградации под нагрузкой.

Даже если с пропускной способностью сети не возникает никаких вопросов, проблемы задержки часто приводят к тому, что более удобной и полезной оказывается установка отдельной, выделенной сети между фронтальной системой и сервером СУБД. В существующих в настоящее время конфигурациях серверов почти всегда используется большое количество главных адаптеров шины SCSI; поскольку как правило эти адаптеры спарены с портами Ethernet, обеспечение выделенного канала Ethernet между клиентом и сервером выполняется простым их подключением к дешевому хабу с помощью кабельных витых пар. Стоимость 4-портового хаба не превышает \$250, поэтому причины, по которым не стоило бы обеспечить такое выделенное соединение, практически отсутствуют.

### **Большие объекты данных**

Однако в ряде случаев возможностей сетей Ethernet или Token Ring может оказаться недостаточно. Чаще всего это случается, когда данные хранятся в базе в виде очень больших массивов. Например, в медицинских базах данных часто хранятся образы рентгеновских снимков, поскольку они могут быть легко объединены с другими данными истории болезни пациента; эти образы рентгеновских снимков часто бывают размером в 3-5 Мбайт. Другими примерами приложений с интенсивным использованием данных являются системы хранения/выборки документов, САПР в области механики и системы мультимедиа. В этих случаях наиболее подходящей сетевой средой является FDDI, хотя в сравнительно ближайшем будущем она может быть заменена на ATM или 100 Мбит Ethernet.

Для систем, требующих большей пропускной способности сети, чем обеспечивают Ethernet или Token Ring, по существу до конца 1994 года FDDI была единственным выбором. Хотя концентраторы FDDI имеют значительно большую стоимость по сравнению с хабами Ethernet, установка выделенной сети между фронтальной системой и сервером СУБД оказывается достаточно простой и недорогой. Как определено в стандарте FDDI, минимальная сеть FDDI состоит только из двух систем, соединенных между собой с помощью единственного кабеля. Никаких концентраторов не требуется, а цены на некоторые интерфейсы FDDI в настоящее время упали до уровня менее \$1500.

### **Конфигурация клиент/сервер и региональные сети**

В последнее время резко увеличилось число приложений, в которых фронтальные системы и серверы СУБД могут или должны размещаться в географически разнесенных местах. Такие системы должны соединяться между собой с помощью глобальных сетей. Обычно средой передачи данных для таких сетей являются арендованные линии с синхронными последовательными интерфейсами. Эти линии обычно работают со скоростями 56-64 Кбит/с, 1.5-2.0 Мбит/с и 45 Мбит/с. Хотя скорость передачи данных в такой среде значительно ниже, чем обычные скорости локальных сетей, подобных Ethernet, природа последовательных линий такова, что они могут поддерживать очень высокий уровень загрузки. Линия T1 предлагает пропускную способность 1.544 Мб/с (2.048 Мбит/с за пределами США). По сравнению с 3.5 Мбит/с, обеспечиваемых Ethernet в обычном окружении, линия T1 предлагает пропускную способность, которая количественно практически не отличается от Ethernet. Неполные линии T3 часто доступны со скоростями 10-20 Мбит/с, очевидно соперничая с сетями Ethernet и Token Ring. В ряде случаев можно найти приложения, которые могут работать успешно в конфигурации клиент/сервер даже через сеть со скоростью 56 Кбит/с.

Для традиционных бизнес-приложений трафик конфигурации клиент/сервер обычно содержит сравнительно низкий объем данных, пересылаемых между фронтальной системой и сервером СУБД, и для них более низкая пропускная способность сети может оказаться достаточной. В большинстве случаев задержка сети не является большой проблемой, однако если глобальная сеть имеет особенно большую протяженность, или если сама среда передачи данных имеет очень большую задержку (например, спутниковые каналы связи), приложение должно быть протестировано для определения его чувствительности к задержкам пакетов.

Для сокращения трафика клиент/сервер до абсолютного минимума могут использоваться мониторы обработки транзакций.

### **Трафик символьного терминала**

Общей ошибкой пользователей является представление о том, что сетевой трафик, связанный с терминальными серверами, может перегрузить Ethernet. Это неправильно. Рассмотрим, например, 64-портовый сетевой терминальный сервер, который может управлять каждым портом, работающим со скоростью 38400 бод, или 38400 символов в секунду. (В каждой байте данных содержится 8 бит информации, но рейтинг бод включает биты старта и стопа). Если каждый порт работает на полной скорости 38400 бод, то всего за одну секунду по сети будет пересылаться 2457600 байт данных (или примерно 1.9 Мбит, т.е. около 20% максимальной загрузки Ethernet) с главной системы в терминальный сервер для дальнейшего распределения. Конечно имеются некоторые накладные расходы (дополнительные байты TCP/IP), связанные с этим уровнем трафика, но они составляют примерно 50 байт на пакет, т.е. примерно 4% для этого уровня трафика. Это сценарий

наихудшего случая, например, когда на полной скорости работают 64 принтера. Типичный же уровень трафика намного ниже: один 2000-символьный экран может отображаться один раз в минуту. При этих условиях 64-портовый терминальный сервер обрабатывает примерно 35 байт в секунду на один порт или всего примерно 2 Кбайт/с.

Операции по вводу символов с клавиатуры терминала можно вообще не рассматривать, поскольку даже самая быстрая машинистка печатает только 20 символов в секунду (более типичный случай 1.0 - 1.5 символов в секунду). Даже если операции ввода обрабатываются в режиме sbreak, наибольшая нагрузка, которая будет генерироваться всеми пользователями, может составлять 1300 символов в секунду (по 20 символов в секунду на каждый порт при 64 портах). После учета максимальных накладных расходов TCP/IP это дает общий поток в 80 Кбайт/с. Типичные нагрузки (64 порта по 1.5 символа в секунду) будут составлять порядка 15 Кбайт/с с учетом накладных расходов.

### **Заключительные рекомендации по конфигурированию сетевого ввода/вывода**

- Для конфигураций клиент/сервер, где клиенты работают на удаленных ПК или рабочих станциях, следует конфигурировать по 20-50 клиентов на одну сеть Ethernet. Количество клиентов в одной сети 16 Мбит/с Token Ring может достигать 100 благодаря прекрасной устойчивости к деградации под нагрузкой.
- Если фронтальная система обслуживает большое количество клиентов, то между фронтальной системой и сервером СУБД следует предусмотреть специальную выделенную сеть. При этом если приходится манипулировать очень большими объектами данных (более 500 Кбайт на объект), то вместо сетей Ethernet или Token Ring следует применять FDDI.
- Фронтальная система и сервер СУБД могут объединяться с помощью глобальных сетей, обеспечивающих "достаточную" полосу пропускания. В идеале это предполагает по крайней мере использование неполной линии T1. Такие приложения должны быть относительно малочувствительны к задержке сети.
- При конфигурировании систем клиент/сервер с использованием глобальных сетей следует исследовать возможность применения мониторов обработки транзакций для снижения трафика клиент/сервер до минимума.
- В частных сетях обычно нет никаких проблем с использованием терминальных серверов, если только сами разделяемые сети не загружены тяжело для других целей.



## Практическая работа №10. Расчет показателей и критериев оценивания ИС.

*Качество* ИС связано с дефектами, заложенными на этапе проектирования и проявляющимися в процессе эксплуатации. Свойства ИС, в том числе и дефектологические, могут проявляться лишь во взаимодействии с внешней средой, включающей технические средства, персонал, информационное и программное окружение.

В зависимости от целей исследования и этапов жизненного *цикла* ИС дефектологические свойства разделяют на *дефектогенность*, *дефектабельность* и *дефектоскопичность*.

*Дефектогенность* определяется влиянием следующих факторов:

- численностью разработчиков ИС, их профессиональными психофизиологическими характеристиками;
- условиями и организацией процесса разработки ИС;
- характеристиками инструментальных средств и комплексов ИС;
- сложностью задач, решаемых ИС;
- степенью агрессивности внешней среды (потенциальной возможностью внешней среды вносить преднамеренные дефекты, например, воздействие вирусов).

*Дефектабельность* характеризует наличие дефектов ИС и определяется их количеством и местонахождением. Другими факторами, влияющими на *дефектабельность*, являются:

- структурно-конструктивные особенности ИС;
- интенсивность и характеристики ошибок, приводящих к дефектам.

*Дефектоскопичность* характеризует возможность проявления дефектов в виде отказов и сбоев в процессе отладки, испытаний или эксплуатации. На *дефектоскопичность* влияют:

- количество, типы и характер распределения дефектов;
- устойчивость ИС к проявлению дефектов;
- характеристики средств контроля и диагностики дефектов;
- квалификация обслуживающего персонала.

Оценка *качества* ИС - задача крайне сложная из-за многообразия интересов пользователей. Поэтому невозможно предложить одну универсальную меру *качества* и приходится использовать ряд характеристик, охватывающих весь спектр предъявляемых требований. Наиболее близки к задачам оценки *качества* ИС *модели качества программного обеспечения*, являющегося одним из важных составных частей ИС. В настоящее время используется несколько абстрактных *моделей качества программного обеспечения*, основанных на определениях характеристики *качества*, показателя *качества*, критерия и *метрики*.

*Критерий* может быть определен как независимый *атрибут* ИС или процесса ее создания. С помощью такого критерия может быть измерена характеристика *качества* ИС на основе той или иной *метрики*. Совокупность нескольких критериев

определяет *показатель качества*, формируемый исходя из требований, предъявляемых к ИС. В настоящее время наибольшее распространение получила *иерархическая модель* взаимосвязи компонентов *качества* ИС. Вначале определяются характеристики *качества*, в числе которых могут быть, например:

- общая полезность;
- исходная полезность;
- *удобство эксплуатации*.

Далее формируются показатели, к числу которых могут быть отнесены:

- практичность;
- целостность;
- корректность;
- удобство обслуживания;
- оцениваемость;
- гибкость;
- адаптируемость;
- мобильность;
- возможность взаимодействия.

Каждому показателю *качества* ставится в соответствие *группа* критериев. Для указанных показателей приведем возможные критерии. Надо отметить, что один и тот же критерий может характеризовать несколько показателей:

- *практичность* - работоспособность, возможность обучения, коммуникативность, объем ввода, скорость ввода-вывода;
- *целостность* - регулирование доступа, контроль доступа;
- *эффективность* - эффективность использования памяти, эффективность функционирования;
- *корректность* - трассируемость, завершенность, согласованность;
- *надежность* - точность, устойчивость к ошибкам, согласованность, простоту;
- *удобство обслуживания* - согласованность, простоту, краткость, информативность, модульность;
- *оцениваемость* - простоту, наличие измерительных средств, информативность, модульность;
- *гибкость* - распространяемость, общность, информативность, модульность;
- *адаптируемость* - общность, информативность, модульность, аппаратную независимость, программную независимость;
- *мобильность* - информативность, модульность, аппаратную независимость, программную независимость;
- *возможность взаимодействия* - модульность, унифицируемость процедур связи, унифицируемость данных.

С помощью *метрик* можно дать количественную или качественную оценку *качества* ИС. Различают следующие виды метрических шкал для измерения критериев.

Первый тип - *метрики*, которые используют интервальную шкалу, характеризующую относительными величинами реально измеряемых физических показателей, например, временем наработки на отказ, вероятностью ошибки, объемом информации и других.

Второй тип - *метрики*, которым соответствует порядковая *шкала*, позволяющая ранжировать характеристики путем сравнения с опорными значениями.

Третий тип - *метрики*, которым соответствуют номинальная, или *категоризованная шкала*, определяющая наличие рассматриваемого свойства или признака у рассматриваемого объекта без учета градаций по этому признаку. Так, например, *интерфейс* может быть "простым для понимания", "умеренно простым", "сложным для понимания".

Развитием иерархического подхода является представленная на [рис. 12.1](#) модель классификации критериев *качества* информационных систем. С помощью *функциональных критериев* оценивается степень выполнения ИС основных целей или задач. Конструктивные критерии предназначены для оценки *компонент* ИС, не зависящих от целевого назначения.

Одним из путей обеспечения *качества* ИС является *сертификация*. В США Радиотехническая комиссия по авионавигации в своем руководящем документе определяет процесс *сертификации* следующим образом:



**Рис. 1.** Модель классификации критериев качества информационных систем

" *Сертификация* - процесс официально выполняемой функции системы ... путем удостоверения, что *функция* ... удовлетворяет требованиям заказчика, а также государственным нормативным документам".

В настоящее время не существует *стандартов*, полностью удовлетворяющих оценке *качества* ИС. В западноевропейских странах имеется ряд *стандартов*, определяющих основы *сертификации* программных систем. *Стандарт* Великобритании (BS750) описывает структурные построения программных систем, при соблюдении которых может быть получен документ, гарантирующий *качество* на государственном уровне. Имеется международный аналог указанного *стандарта* (ISO9000) и аналог для стран-членов НАТО (AQAP1). Существующая в нашей стране система нормативно-технических документов относит *программное обеспечение* к "продукции производственно-технического назначения", которая рассматривается как материальный объект. Однако *программное обеспечение* является скорее абстрактной нематериальной сферой. Существующие ГОСТы (например, ГОСТ 28195-89 "Оценка *качества* программных средств. Общие положения") явно устарели и являются неполными.

### **Стандарты управления качеством промышленной продукции**

Международные *стандарты* серии *ISO 9000* разработаны для управления *качеством* продукции, их дополняют *стандарты* серии *ISO14000*, отражающие экологические требования к производству промышленной продукции. Хотя эти *стандарты* непосредственно не связаны с *CALS- стандартами*, их цели - совершенствование промышленного производства, повышение его эффективности - совпадают.

Очевидно, что управление *качеством* тесно связано с его контролем. *Контроль качества* традиционно основан на измерении показателей *качества* продукции на специальных технологических операциях контроля и выбраковке негодных изделий. Однако есть и другой подход к управлению *качеством*, который основан на контроле качественных показателей не самих изделий, а *проектных процедур* и технологических процессов, используемых при создании этих изделий.

Такой подход во многих случаях более эффективен. Он требует меньше затрат, поскольку позволяет обойтись без стопроцентного контроля продукции и благодаря предупреждению появления брака снижает производственные издержки. Именно этот подход положен в основу *стандартов ISO 9000*, принятых *ISO* в 1987 г. и проходящих корректировку приблизительно каждые пять лет.

Таким образом, методической основой для управления *качеством* являются международные *стандарты* серии *ISO 9000*. Они определяют и регламентируют инвариантные вопросы создания, развития, применения и *сертификации* систем *качества* в промышленности. В них устанавливается форма требований к системе *качества* в целях демонстрации поставщиком своих возможностей и оценки этих возможностей внешними сторонами.

Основной причиной появления *стандартов ISO 9000* была потребность в общем для всех участников международного рынка базисе для контроля и управления *качеством* товаров. Американское общество контроля *качества* определило цели *ISO 9000* как помощь в развитии международного обмена товарами и услугами и кооперации в сфере интеллектуальной, научной, технологической и деловой активности.

В *стандартах ISO 9000* используется определение *качества* из *стандарта ISO 8402*: " *Качество* - совокупность характеристик продукта, относящихся к его способности удовлетворять установленные или предполагаемые потребности". Аналогичное определение содержится в ГОСТ 15467-79: " *Качество* продукции - это совокупность свойств продукции, обуславливающих ее пригодность удовлетворять определенные потребности в соответствии с ее назначением". В *ISO 9000* вводится понятие системы *качества* (QS - *Quality System*), под которой понимают документальную систему с руководствами и описаниями процедур достижения *качества*. Другими словами, система *качества* есть совокупность организационной структуры, ответственности, процедур, процессов и ресурсов, обеспечивающая осуществление общего

руководства *качеством*. Система качества обычно представляет собой совокупность трех слоев документов:

- описание политики управления для каждого системного элемента;
- описание процедур управления *качеством* (что, где, кем и когда должно быть сделано);
- тесты, планы, инструкции и т. п.

*Сертификация* предприятий по *стандартам ISO 9001-9003* выполняется некоторой уполномоченной внешней организацией. Наличие сертификата *качества* - одно из важных условий для успеха коммерческой деятельности предприятий.

*Вторичные стандарты* включают в себя:

- *ISO 9000* - основные понятия, руководство по применению *ISO 9001*;
- *ISO 9004* - элементы систем управления *качеством*. *Поддерживающие стандарты* предназначены для развития и установки систем *качества*:
- *ISO 10011* - аудит, критерии для аудита систем *качества* ;
- *ISO 10012* - требования для измерительного оборудования;
- *ISO 10013* - пособие для развития руководств по управлению *качеством*.

Часть этих *стандартов* утверждена как государственные *стандарты* Российской Федерации. В частности, к ним относятся:

- ГОСТ Р ИСО 9001-96 "Системы *качества*. Модель обеспечения качества при проектировании, разработке, производстве, монтаже и обслуживании";
- ГОСТ Р ИСО 9002-96 "Системы *качества*. Модель обеспечения качества при производстве, монтаже и обслуживании";
- ГОСТ Р ИСО 9003-96 "Системы *качества*. Модель обеспечения качества при окончательном контроле и испытаниях".

В настоящее время разработана новая версия *стандартов* серии *ISO 9000* под названием *ISO 9000:2000 Quality management systems* (системы управления *качеством* ), в которую включены следующие документы:

- *ISO 9000:2000 Fundamentals and vocabulary* (основы и терминология);
- *ISO 9001:2000 Requirements* (требования);
- *ISO 9004:2000 Guidelines for performance improvement* (руководство по развитию).

Главное отличие новой версии от предыдущей состоит в том, что она обусловлена стремлением упростить практическое использование *стандартов*, направлена на их лучшую гармонизацию и заключаются в следующем.

В *стандарте ISO 9001* минимизируется объем требований к системе *качества*. *Стандарты ISO 9002-9003* из новой версии исключаются. Расширяется круг контролируемых ресурсов, в их число включены такие элементы, как *информация*, коммуникации, *инфраструктура*.

Введенные в *стандарте ISO 9004* двадцать элементов *качества* сворачиваются в четыре группы:

- распределение ответственности (management responsibility);
- управление ресурсами (resource management);
- реализация продукции и услуг (product and/or service realization);
- измерения и анализ (measurement, analysis, and improvement).

*Сертификация* предприятий по *стандартам ISO 9001-9003* выполняется некоторой уполномоченной внешней организацией. Наличие сертификата *качества* - одно из важных условий для успеха коммерческой деятельности предприятий.

*Стандарты ISO 14000* являются также системой управления влиянием на окружающую среду; они, как и *ISO 9000*, реализуются в процессе *сертификации* предприятий, задают процедуры управления и *контроль* документации, *аудит*, подразумевают соответствующее обучение и сбор статистики. Кроме требований заказчиков и покупателей, в них воплощаются внутренние требования организации.

## Контрольные вопросы

1. Чем определяется *качество* ИС?
2. Какие характеристики *качества* можно определить?
3. Что определяет показатель *качества*?
4. Охарактеризуйте дефектологические свойства в зависимости от целей исследования и этапов жизненного цикла  
ИС: *дефектогенность, дефектабельность и дефектоскопичность*.
5. Как формируется показатель *качества*?
6. Какие существуют виды метрических шкал для измерения критериев?
7. Поясните модель классификации критериев *качества* информационных систем (рис.1)
8. Что оценивается с помощью *функциональных критериев*?
9. Для чего предназначены конструктивные критерии?
10. Расскажите о нормативных документах по оценке *качества* информационных систем.
11. На чем традиционно основан контроль *качества*?
12. Что является методической основой для управления *качеством* ИС?
13. Что представляет собой совокупность документов системы *качества*?
14. Что включают в себя вторичные *стандарты* системы *качества*?
15. Для чего предназначены поддерживающие *стандарты*?