

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ

**«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**Кафедра управления в технических системах**

УТВЕРЖДАЮ:

Проректор по учебной работе

\_\_\_\_\_ Е.И. Луковникова

« \_\_\_\_\_ » \_\_\_\_\_ 201 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ  
СИСТЕМ УПРАВЛЕНИЯ**

**Б1.В.ДВ.06.02**

**НАПРАВЛЕНИЕ ПОДГОТОВКИ**

**27.03.04 Управление в технических системах**

**ПРОФИЛЬ ПОДГОТОВКИ**

**Управление и информатика в технических системах**

Программа академического бакалавриата

Квалификация выпускника: бакалавр

<b>1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ .....</b>	<b>3</b>
<b>2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ .....</b>	<b>3</b>
<b>3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ .....</b>	<b>4</b>
3.1 Распределение объёма дисциплины по формам обучения.....	4
3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости .....	4
<b>4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ .....</b>	<b>5</b>
4.1 Распределение разделов дисциплины по видам учебных занятий .....	5
4.2 Содержание дисциплины, структурированное по разделам и темам .....	7
4.3 Лабораторные работы.....	73
4.4 Семинары / практические занятия.....	73
4.5. Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат.....	73
<b>5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>74</b>
<b>6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ</b>	<b>75</b>
<b>7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>	<b>75</b>
<b>8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>76</b>
<b>9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....</b>	<b>77</b>
9.1. Методические указания для обучающихся по выполнению лабораторных работ/ семинаров / практических работ .....	77
<b>10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ .....</b>	<b>85</b>
<b>11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ .....</b>	<b>85</b>
<b>Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....</b>	<b>86</b>
<b>Приложение 2. Аннотация рабочей программы дисциплины .....</b>	<b>91</b>
<b>Приложение 3. Протокол о дополнениях и изменениях в рабочей программе .....</b>	<b>92</b>

# 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

## Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к проектно-конструкторскому виду профессиональной деятельности выпускника в соответствии с компетенцией и видами деятельности, указанными в учебном плане.

## Цель дисциплины

Изложение базовых принципов создания прикладных программных средств и информационного обеспечения систем управления в целом, а также формирование практических навыков по проектированию базы данных как основы информационного обеспечения систем управления.

## Задачи дисциплины

Подготовить обучающихся к самостоятельной работе по решению практических задач, связанных с построением и функционированием информационного обеспечения систем управления (ИОСУ), с использованием методов и средств создания проектов автоматизированных систем управления технологическими процессами с применением современных программных средств, с проектированием моделей ИОСУ реальных объектов регулирования.

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
1	2	3
ПК-6	способность производить расчёты и проектирование отдельных блоков и устройств систем автоматизации и управления и выбирать стандартные средства автоматизации, измерительной и вычислительной техники для проектирования систем автоматизации и управления	<b>знать:</b> – принципы проектирования и разработки информационного обеспечения систем управления; <b>уметь:</b> - выбирать стандартные средства автоматизации, измерительной и вычислительной техники для проектирования систем автоматизации и управления; <b>владеть:</b> - навыками расчётов и проектирования отдельных блоков и устройств систем автоматизации и управления.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.В.ДВ.6.2 «Информационное обеспечение систем управления» относится к вариативной части, дисциплина по выбору.

Дисциплина «Информационное обеспечение систем управления» базируется на знаниях, полученных при изучении таких учебных дисциплин, как Б1.Б.05 Математика, Б1.В.ДВ.05.01 Математическая статистика, Б1.Б.15 Теория автоматического управления, Б1.В.16 Системы управления базами данных.

Дисциплина «Информационное обеспечение систем управления» представляет основу для Б2.В.03(П) Производственной (преддипломной) практики, Б1.Б.01 Защиты выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты.

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

### 3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ

#### 3.1. Распределение объема дисциплины по формам обучения

<i>Форма обучения</i>	<i>Курс</i>	<i>Семестр</i>	<i>Трудоёмкость дисциплины в часах</i>						<i>Контрольная работа</i>	<i>Вид промежуточной аттестации</i>
			<i>Всего часов (с экз.)</i>	<i>Аудиторных часов</i>	<i>Лекции</i>	<i>Лабораторные работы</i>	<i>Практические занятия</i>	<i>Самостоятельная работа</i>		
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>Очная</b>	4	8	108	48	24	-	24	60	-	зачёт
<b>Заочная</b>	4	-	108	14	6	-	8	90	-	зачёт
<b>Заочная (ускоренное обучение)</b>	-	-	-	-	-	-	-	-	-	-
<b>Очно-заочная</b>	-	-	-	-	-	-	-	-	-	-

#### 3.2. Распределение объема дисциплины по видам учебных занятий и трудоёмкости

<i>Вид учебных занятий</i>	<i>Трудоёмкость (час.)</i>	<i>в т.ч. в интерактивной, активной, инновационной формах, (час.)</i>	<i>Распределение по семестрам, час</i>
			<b>8</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>I. Контактная работа обучающихся с преподавателем (всего)</b>	48	10	48
Лекции (Лк)	24	6	24
Практические занятия (ПЗ)	24	4	24
Групповые (индивидуальные) консультации	+	-	+
<b>II. Самостоятельная работа обучающихся (СР)</b>	60	-	60
Подготовка к практическим занятиям	45	-	45
Подготовка к зачёту	15	-	15
<b>III. Промежуточная аттестация</b> зачёт	+	-	+
Общая трудоёмкость дисциплины ..... час.	108	-	108
	зач. ед.	3	3

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной формы обучения:

№ раз- дела и темы	Наименование раздела и тема дисциплины	Трудоем- кость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоя тельная работа обучаю- щихся
			лекции	практи- ческие занятия	
1	2	3	4	5	6
<b>1.</b>	<b>Введение. Предмет и задачи курса</b>	<b>12</b>	<b>3</b>	<b>-</b>	<b>9</b>
1.1.	Введение	2,5	0,5	-	2
1.2.	Основные понятия курса	2,5	0,5		2
1.3.	Основные процессы преобразования информации и их характеристика	4	1	-	3
1.4	Системы информационного обмена	3	1	-	2
<b>2.</b>	<b>Управление информацией и системы баз данных</b>	<b>16</b>	<b>3</b>	<b>4</b>	<b>9</b>
2.1.	Задачи и функции ИС	2	0,5	-	1,5
2.2.	Типология ИС	2	0,5	-	1,5
2.3.	Состав и структура ИС	3	0,5	1	1,5
2.4.	АИПС, банки и базы данных	3	0,5	1	1,5
2.5.	Информационные сети	3	0,5	1	1,5
2.6.	Обеспечивающие подсистемы ИС и их характеристика	3	0,5	1	1,5
<b>3.</b>	<b>Жизненный цикл информационных систем</b>	<b>15</b>	<b>3</b>	<b>4</b>	<b>8</b>
3.1.	Модели жизненного цикла (ЖЦ) ИС	2	0,5	-	1,5
3.2.	Стандарты на проектирование ИС	1,5	0,5	-	1,0
3.3.	Процессы ЖЦ ПО	1,5	0,5	-	1
3.4.	Каноническое проектирование ИС	2	0,5	-	1,5
3.5.	Содержание технического задания на ИС	4	0,5	2	1,5
3.6.	Содержание технического проекта ИС	4	0,5	2	1,5
<b>4.</b>	<b>Технология и методология проектирования автоматизированных информационных систем</b>	<b>15</b>	<b>3</b>	<b>4</b>	<b>8</b>
4.1.	Специфика информационных программных систем	5	1	1	3
4.2.	Трудности, встречающиеся при проектировании ИС	5	1	1	3
4.3	Критерии создания ИС	5	1	2	2
<b>5.</b>	<b>Требования к организации диалога и представлению данных</b>	<b>15</b>	<b>3</b>	<b>4</b>	<b>8</b>

5.1.	Стратегии построения ИС	5,5	1,5	-	4
5.2.	Методологии и технологии проектирования ИС	3	1,0	-	2
5.3.	Структурное представление ИС при их создании	6,5	0,5	4	2
<b>6.</b>	<b>Принципы построения и организации баз данных</b>	<b>9</b>	<b>3</b>	<b>-</b>	<b>6</b>
6.1.	Численные и информационные прикладные системы	2,1	0,6	-	1,5
6.2.	Файловые системы	1,6	0,6	-	1
6.3.	Области применения файлов	2,1	0,6	-	1,5
6.4.	Потребности информационных систем	1,6	0,6	-	1
6.5.	СУБД и представление данных в базах данных	1,6	0,6	-	1
<b>7.</b>	<b>Дореляционные структуры и модели данных</b>	<b>13</b>	<b>3</b>	<b>4</b>	<b>6</b>
7.1.	Различные представления о данных в базах данных	4,5	1,5	-	3
7.2.	Основные этапы проектирования базы данных	8,5	1,5	4	3
<b>8.</b>	<b>Реляционная модель данных</b>	<b>13</b>	<b>3</b>	<b>4</b>	<b>6</b>
8.1.	Модели данных	3	1	1	1
8.2.	Типы моделей данных	10	2	3	5
	<b>ИТОГО</b>	<b>108</b>	<b>24</b>	<b>24</b>	<b>60</b>

- для заочной формы обучения:

№ раздела и темы	Наименование раздела и тема дисциплины	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся
			лекции	практические занятия	
1	2	3	4	5	6
1.	Введение. Предмет и задачи курса	13	1	-	12
2.	Управление информацией и системы баз данных	13	0,5	1,5	11
3.	Жизненный цикл информационных систем	12,5	0,5	1	11
4.	Технология и методология проектирования автоматизированных информационных систем	12,5	0,5	1	11
5.	Требования к организации диалога и представлению данных	14,5	1	1,5	12
6.	Принципы построения и организации баз данных	12	1	-	11
7.	Дореляционные структуры и модели данных	13	0,5	1,5	11
8.	Реляционная модель данных	13,5	1	1,5	11
	<b>ИТОГО</b>	<b>104</b>	<b>6</b>	<b>8</b>	<b>90</b>

## **4.2. Содержание дисциплины, структурированное по разделам и темам**

### **Раздел 1. ВВЕДЕНИЕ. ПРЕДМЕТ И ЗАДАЧИ КУРСА**

#### **Тема 1.1. Введение**

Развитие общества всегда сопровождалось усложнением и интенсификацией связей во всех областях деятельности человека, постоянным накоплением знаний, повышением наукоемкости производимой обществом продукции. В ходе такого процесса информация и знания приобретали постепенно характер самостоятельных видов ресурсов общества, т.е. информационных ресурсов, обеспечивающих экономию практически всех общественных видов ресурсов (материальных, интеллектуальных, трудовых и т.д.). Это привело к тому, что на современном этапе темпы дальнейшего развития общества стали определяться совершенством его информационной инфраструктуры, эффективностью формирования, размещения и использования информационных ресурсов. Происходящее в течение последних 30-40 лет качественное изменение роли информации в жизни общества выявило глубокие диспропорции производительности труда и инструментальнооснащенности в сферах материального производства и информационной сфере, т.е. в сфере информационного обеспечения общества. Сегодня в промышленно развитых странах инструментальнооснащенность информационной сферы более чем в десятки раз ниже (в стоимостном выражении) инструментальнооснащенности сферы материального производства. Еще контрастнее картина диспропорций производительности труда в указанных сферах. Так, в течение 80-х годов темпы роста производительности труда в сферах обрабатывающей промышленности развитых стран в 20 раз превышали темпы роста производительности труда в информационной сфере. В результате низкая производительность труда в информационной сфере, с одной стороны, и возросшая роль информации во всех сферах деятельности человека, с другой, привели к росту относительной численности работников, занятых в информационной сфере.

В современных условиях темпы научно-технического прогресса в значительной степени определяются степенью информационного обеспечения общества, оперативностью и эффективностью использования информации. Основу информационной структуры общества, обеспечивающей высокие темпы научно-технического прогресса, составляют автоматизированные информационные системы (АИС) и сети с распределенными банками и базами данных, интегрированные и интегральные АИС, автоматизированные информационные системы управления разного уровня, реализующие высокоэффективные информационные технологии (ИТ) на базе электронно-вычислительной и телекоммуникационной техники.

Сегодня в нашей стране проводится огромная работа по созданию информационной инфраструктуры, формированию, размещению и эффективному использованию информационных ресурсов на базе современной сетевой автоматизированной информационной технологии, которая, являясь по существу совокупностью средств и методов реализации технологических процессов в распределенных информационных системах (РИС), дает большие возможности повышения эффективности функционирования информационных систем и сетей за счет более совершенной организации и координации их технологических процессов и расширение рынка сбыта информации путем подключения к узлам информационных систем (ИС) и сетей удаленных пользователей. При этом абонент, обращаясь к услугам РИС, не должен знать ни топологию размещения ресурсов (информационных, технических, программных), ни тем более маршрутизацию своего запроса, алгоритмы и процедуры поиска информации. Однако она должна быть построена таким образом, чтобы обеспечить, с одной стороны, оптимальное размещение всех видов ее ресурсов, соответствующее заданной системе ограничений и критериев качества, а с другой стороны, - оптимальную маршрутизацию конкретного запроса, алгоритм и процедуры поиска информации.

Сетевая информационная технология (СИТ), принципы и возможности которой заложены уже в самой природе производственных отношений, не является чем-то новым. Новым являются технические, телекоммуникационные и программные средства СИТ, которые создают принципиально новые возможности реализации сетевой информационной технологии в современных условиях. Поэтому совершенствование информационной инфраструктуры страны в этих условиях требует разработки конкретных методов и средств координации и управления распределенными ИС и сетями, адекватных современной автоматизированной информационной технологии и обеспечивающих оптимизацию процессов формирования, размещения и использования информационных ресурсов РИС и сетей.

## **Тема 1.2. Основные понятия курса**

Информатика - научная дисциплина, изучающая движение, структуру и свойства семантической, т.е. смысловой информации. Информатика в определенной степени связана с математической теорией информации, основы которой были разработаны Клодом Шенноном и методы которой широко применяются в информатике. Однако следует различать математическую теорию информации и информатику. Центральным моментом математической теории информации являются понятия информации и меры для измерения ее количества. Указанная теория целиком и полностью отвлекается от содержательной (семантической) стороны информации, тогда как в информатике именно эта сторона является наиболее существенной. Информатика кроме количественных характеристик информации, рассматривает также ее качественную особенность, ее смысловое (семантическое) значение, важность информации для потребителя.

В настоящее время наиболее распространенным убеждением является то, что универсального понятия информации не существует. Поэтому рассмотрим некоторые из них.

Если специфика информации определяется в первую очередь основной целью функционирования системы, то с этой точки зрения информацией являются все сведения об объекте, полезные «приемнику» (человеку, коллективу, человеко-машинной системе) для решения задачи (достижения цели). Если данные сведения не нужны, то они представляют собой «шум», а не информацию. В случае, когда данные сведения способствуют принятию неправильного решения, то они представляют собой дезинформацию.

В отличие от информации имеющиеся данные – это зарегистрированные на любых носителях сведения об объекте (реальном или вымышленном) независимо от того, дошли они до какого-либо приемника и интересуют ли они его. В такой трактовке информация понимается как данные, ценные для получателя (приемника). Это определение оказывается наиболее целесообразным для анализа информационных процессов и подчеркивает относительную «важность» той или иной информации для решения конкретных задач. В таком случае данные представляют собой потенциальную информацию, и с этой точки зрения в информационных системах накапливается не информация, а данные (потенциальная информация). Информацией они становятся по предоставлению их некоторому потребителю. В таком определении информация не отождествляется со знаниями, поскольку информация есть собрание данных, тогда как знание предполагает постижение действительности сознанием, организующим данные путем их анализа.

Наряду с понятиями информация и данные широко используется понятие научная информация, т.е. получаемая в процессе познания логическая информация, адекватно отображающая закономерности объективного мира и используемая в общественно-исторической практике. Выделяют четыре наиболее существенных ее признака:

1) Поскольку научной может быть лишь информация, получаемая в процессе познания закономерностей объективной действительности, то источником научной информации может быть не только исследовательская деятельность ученых, но и производственная, хозяйственная деятельность инженеров, рабочих-новаторов и работников сельского хозяйства: их многочисленные изобретения, усовершенствования и рационализаторские предложения.



2) Информация становится научной лишь тогда, когда она подвергнута обработке и обобщению абстрактно-логическим мышлением; именно этим научная информация отличается от сведений, получаемых человеком в процессе чувственного познания мира.

3) Адекватность отображения объективной действительности. Поскольку в процессе познания люди могут получать логическую информацию, дающую искаженные представления об окружающем мире, то ошибочная гипотеза или теория вполне научна, если ведется систематическое изучение, обобщение и проверка на практике ее положений. В этом и состоит принципиальное отличие ошибочных научных гипотез и теорий от лженаучных.

4) Непременное ее использование в общественно-исторической практике, что обеспечивает постоянную проверку истинности логической информации и препятствует отнесению к категории научной информации устаревших истин.

В качестве меры количества семантической информации обычно рассматривают степень изменения тезауруса получателя при приеме некоторого сообщения. Если сообщение не имеет ничего общего с тезаурусом потребителя, не пересекается с ним, то принимаемая информация равна нулю.

Под тезаурусом потребителя информации понимается запас знаний потребителя, представленный в виде специального словаря (тезауруса).

Независимо от сферы деятельности человека выполнение любой работы и решение любой проблемы всегда связано с использованием существующей и созданием новой информации. С этой точки зрения любая основная деятельность человека связана с его информационной деятельностью, т.е. с деятельностью по сбору и обработке существующей и созданию новой информации. Таким образом, информационная деятельность человека неразрывно связана с основной деятельностью, а субъект основной деятельности (СОД) всегда выполняет три взаимосвязанных функции: потребителя информации; собственно исполнителя основной работы; генератора (источника) информации. Следовательно, понятие информации тесно связано с понятием потребителя (приемника) информации, генератора (источника) информации, информационной потребности и другими понятиями, определяющими степень соответствия информации, полученной потребителем, его информационным потребностям.

Под потребителем (приемником) информации понимают отдельное лицо, коллектив, машинную или человеко-машинную (организационную) систему, использующие информацию в целях выполнения определенной работы в процессе основной деятельности.

В отличие от потребителя информации генератором (источником) информации является отдельное лицо, коллектив, машинная или человеко-машинная система, создающая сообщения в результате выполнения той или иной деятельности.

Под информационной потребностью (ИП) понимают совокупность элементов информации, необходимую и достаточную для эффективного выполнения заданной работы субъектом основанной деятельности. ИП зависит: от степени подготовленности СОД к выполнению заданной работы; его знаний и умения; наличия уже собранной информации; от трудоемкости и сложности заданной работы; ее внутреннего многообразия и взаимосвязи с внешней средой.

Информационная потребность СОД удовлетворяется в результате поиска сообщений в информационной среде по информационному запросу, сформулированному на естественном языке и в той или иной мере отражающему информационную потребность.

Степень адекватности соответствия информационного запроса информационной потребности определяется различными факторами, основным из которых является *способность СОД сформулировать* свою информационную потребность на естественном языке с учетом специфики как стоящей перед ним проблемы, так и информационной среды. Найденные в результате информационного поиска сообщения в той или иной мере соответствуют информационному запросу и информационной потребности.

Характеристику степени соответствия сообщения информационной потребности называют пертинентность, а характеристику степени соответствия сообщения информационному запросу называют релевантность.

Поскольку информационный запрос практически никогда не соответствует полной и точной ИП, то релевантность информации всегда отличается от ее пертинентности. Понятия

пертинентности информации и релевантности информации тесно связаны с понятием количества информации и ее ценности с точки зрения решаемой СОД проблемы.

Если условно изобразить тезаурусы потребителя через ТП, решаемой задачи - ТЗ, информационной потребности - ТИП, информационного запроса - ТИЗ и найденного информационного сообщения - ТИС в виде плоских геометрических фигур, то пересечения этих фигур между собой достаточно наглядно покажут взаимоотношения указанных понятий (рис. 1.1).

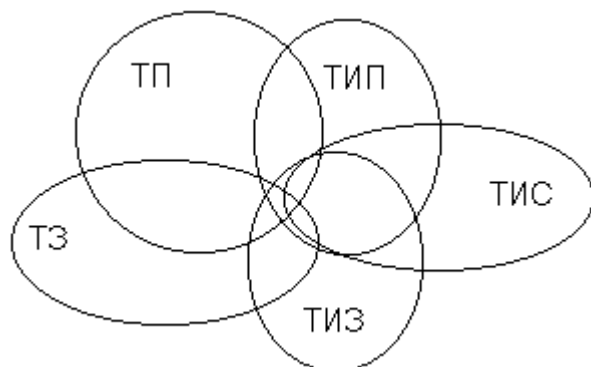


Рис. 1.1. Схема пересечения тезаурусов

Из рис. 1.1 видно, что тезаурус найденного сообщения может иметь мало общего с тезаурусами как задачи, так и информационной потребности. Цель информационного поиска состоит в нахождении *множества* сообщений, совокупный тезаурус которых должен соответствовать тезаурусу информационной потребности и который, в свою очередь, вместе с тезаурусом потребителя должен покрывать тезаурус решаемой задачи.

Информационный запрос и его тезаурус служат промежуточным звеном между потребителем и информационной средой, обеспечивающим поиск необходимой информации. Количество информации (по Шрейдеру Ю.Л.) определяется степенью изменения тезауруса потребителя (ТП) под воздействием тезауруса сообщения (ТИС).

В отличие от количества информации, ценность сообщения (информации) определяется по отношению к решаемой задаче и, следовательно, ее можно оценить при рассмотрении взаимоотношения тезаурусов сообщения и задачи.

### **Тема 1.3. Основные процессы преобразования информации и их характеристика**

С информационной точки зрения любая деятельность человека или человеко-машинной системы включает следующие этапы:

- 1) поиск (получение сообщения);
- 2) интерпретация сообщения;
- 3) решение (выполнение) задачи (разработки);
- 4) создание сообщения;
- 5) распространение сообщения.

Целью первого этапа является отбор из внешнего потока сообщений тех из них, которые могут быть использованы при выполнении основной деятельности. На этом этапе используются услуги различных информационных систем и неформальные каналы, доступные конкретному потребителю информации.

Второй этап заключается в адаптации сообщений, т.е. в извлечении из сообщений информации, необходимой для решения поставленной задачи. Этап 2 приводит к созданию информационного обеспечения решаемой задачи, которое должно привести к повышению эффективности ее решения.

На третьем этапе, используя полученную информацию, собственные знания и опыт, а также материальные, энергетические, трудовые и иные ресурсы, СОД вырабатывает новую информацию, которая является результатом решения задачи.

На этапе 4 СОД подготавливает сообщение в стандартной форме, принятой на данном этапе развития системы научной коммуникации вообще и систем научно-технической информации (НТИ), в частности. Это может быть подготовка статьи, отчета, выступления на научной конференции, семинаре и т.п.

Следует отметить, что информация, полученная на этапе 3, без дополнительных затрат, как правило, не представляет ценности при выходе за пределы конкретной задачи, являясь достоянием СОД. Однако результаты четвертого этапа потенциально представляют собой общественную ценность для решения большого круга задач, так или иначе связанных с задачами, в результате решения которой информация получена.

На пятом этапе автор сообщения вступает в активное взаимодействие с системой коммуникации, затрачивая усилия (в основном организационного характера) по вводу сообщения в один (или несколько) из доступных каналов коммуникации (депонирование сообщения, публикация, выступление и т.п.). Эффективность этапа определяется как степенью усилий, предпринимаемых СОД, так и теми возможностями, которые ему предоставляет система коммуникации.

1-й и 5-й этапы приведенной декомпозиции деятельности являются этапами собственной информационной деятельности.

3-й этап - собственно основная деятельность.

2-й и 4-й этапы являются пограничными и могут быть отнесены как к основной, так и к информационной деятельности.

#### **Тема 1.4. Системы информационного обмена**

В связи с многозначностью понятия «информация» информационной можно назвать любую систему, представляющую собой совокупность элементов информационной деятельности. Таким образом система информационного обмена (СИОБ) - это взаимосвязанная совокупность поставщиков (источников) информации, органов ее преобразования или передачи (в том числе информационных и издательских органов), а также потребителей (приемников) информации. В таком понимании система информационного обмена включает в себя как *основную*, так и *информационную* деятельность в их органическом единстве. В зависимости от характера информационных связей с внешней средой выделяют три разновидности систем информационного обмена:

Если система не имеет информационных связей с другими системами (внешней средой) и следовательно по отношению к ним не является ни источником, ни приемником информации, то ее называют замкнутой. Примером такой системы может служить любая изолированная саморазвивающаяся система (любое государство или группа людей, изолированные от внешнего мира; ученый, в одиночку работающий над решением некоторой проблемы и не использующий информацию внешней среды).

Система, открытая для поступления информации *извне*, но закрытая для выхода информации во внешнюю среду, т.е. которая является только приемником по отношению к внешним системам, называется закрытой.

Система, играющая роль одновременно и источника и приемника информации по отношению к внешним системам, называется открытой системой.

Примерами закрытых систем информационного обмена являются военно-промышленные комплексы, которые только потребляют информацию и держат в секрете результаты своих разработок. Примерами открытых систем являются высокоразвитые в области «производства» новой информации системы, стоящие в авангарде научных исследований и разработок.

Из определения системы информационного обмена следует, что всякая интеграция систем сопровождается переходом от открытых и закрытых систем к замкнутым и открытым системам, что соответствует переходу от простого потребления информации к процессам ее обмена между участниками системы.

Усложнение организационных систем различных уровней и процессов их функционирования и развития приводит к необходимости организационного обособления

основной и информационной деятельности. Этот процесс столь же закономерен, как и процесс иерархического построения самих организационных систем, и по сути, является его следствием.

Если в чисто технических системах, где четко выделяется объект управления (ОУ) и устройство управления (УУ), что соответствует разделению деятельности на основную и информационную, то в организационных системах провести грань между основной и информационной деятельностью значительно труднее. Но, несмотря на это, в любой организационной системе верхние уровни ее иерархии в основном заняты информационной деятельностью, что позволяет говорить об ее организационной обособленности.

Информационной системой (ИС) называют систему, предназначенную для сбора, обработки и распространения информации.

Цель функционирования ИС – информационное обслуживание основной деятельности системы информационного обмена (СИОБ). Функционирование ИС предполагает наличие: исходных, промежуточных и конечных информационных продуктов, технологических процессов и ресурсов.

Необходимой составляющей информационной системы является подсистема управления, обеспечивающая ее эффективное функционирование. В отличие от системы информационного обмена в ИС не включены как поставщики, так и потребители информации. ИС объединяет только преобразующие информацию элементы, т.е. ввода, обработки, хранения, поиска, вывода и распространения информации (рис. 1.2).



Рис. 1.2. Информационная структура СИОБ

Исходя из сказанного, информационная система – это взаимосвязанная совокупность элементов ввода, обработки, переработки, хранения, поиска, вывода и распространения информации, цель функционирования которой состоит в информационном обеспечении эффективной деятельности организационной системы, подсистемой которой она является. Выделение ИС из системы информационного обмена как самостоятельной системы является следствием организационного обособления основной и информационной деятельности.

Существующие сегодня автоматизированные документальные и фактографические информационно-поисковые системы (ИПС), автоматизированные системы управления (АСУ), банки данных (БнД), отделы научно-технической информации (ОНТИ), системы научно-технической информации (НТИ) в целом организационно обособлены. Но при этом стоит вопрос как рассматривать поставщиков и потребителей этих систем: как элементы собственно систем или как элементы внешней среды? Вопрос решается выделением внутренних и внешних как потребителей так и поставщиков информации. Если считать, что потребители информации являются элементами систем НТИ, АСУ, БнД и т.д., то необходимо предположить, что данные системы могут управлять их информационными потребностями. В действительности же системы могут только изучать эти потребности и

приспосабливаться к ним. Функции управления по отношению к потребителям реализует система более высокого уровня – система информационного обмена.

Для системы существенно взаимодействие ее элементов для достижения общей цели. Если этого взаимодействия нет, то нет и системы. С другой стороны ОНТИ, БТИ (бюро технической информации), АСУ, БНД и т.д. организационно не принадлежат этим системам. Возникает вопрос, а как же можно эффективно управлять функционированием системы, имеющей элементы, организационно не принадлежащие ей. Ответ один - рассматривать эти элементы как элементы внешней среды. То же самое можно сказать и в отношении поставщиков информации. Таким образом, внутренними источниками и потребителями информационных систем являются те, которые входят в состав элементов данной системы информационного обмена, т.е. системы более высокого уровня, по отношению к которой ИС является подсистемой. Элементы внешней по отношению к данной системе информационного обмена среды являются внешними источниками и потребителями информации.

## **Раздел 2. УПРАВЛЕНИЕ ИНФОРМАЦИЕЙ И СИСТЕМЫ БАЗ ДАННЫХ**

Лекция проводится в интерактивной форме с текущим контролем (3 час.)

### **Тема 2.1. Задачи и функции ИС**

При организационном обособлении ИС решают две группы задач. Первая связана с чисто информационным обеспечением основной деятельности: отбор необходимых сообщений и их обработка, хранение, поиск и выдача субъекту основной деятельности с заранее заданной полнотой, точностью и оперативностью в наиболее приемлемой для СОД форме.

Вторая группа задач связана с обработкой полученной информации в соответствии с теми или иными алгоритмами или программами с целью подготовки решения задач, стоящих перед субъектом основной деятельности.

Для решения таких задач ИС должна обладать необходимой информацией о предметной области (ПО) субъекта основной деятельности, стоящих перед ним проблемах, должна уметь использовать существующие модели решения задач СОД или самостоятельно строить такие модели, а также обладать определенным искусственным или естественным интеллектом.

Предметная область – это множество объектов и отношений между ними, ограниченное потребностями конкретного субъекта основной деятельности.

С развитием вычислительной техники (ВТ) и ее программного обеспечения, упрощением технологии работы с ВТ задачи второй группы все чаще выполняют сами субъекты основной деятельности. Однако существует определенный круг задач второй группы любой из сфер основной деятельности, решение которых целесообразно возлагать на информационные системы. Если задачи первой группы – это задачи информатизации общества «вширь», то задачи второй группы – задачи информатизации общества «вглубь».

Для решения поставленных задач, т.е. задач первой и второй группы, ИС должна выполнять следующие функции:

- отбор сообщений из внутренней и внешней среды, необходимых для реализации основной деятельности;
- ввод информации в ИС;
- хранение информации в памяти, ее актуализация и поддержание целостности;
- обработка, поиск и выдача информации в соответствии с заданными субъектом основной деятельности требованиями. Обработка может включать и подготовку вариантов решения пользовательских прикладных задач по соответствующим алгоритмам (программам).

## Тема 2.2. Типология ИС

Когда говорят об информационных системах, то всегда имеют в виду, что информационная система не существует сама по себе. Она является подсистемой более общей системы – системы информационного обмена. Практически бесконечное многообразие организационных систем порождает огромное разнообразие ИС как их подсистем. Еще сложнее обстоит дело, когда речь идет об автоматизированных и автоматических ИС.

Под автоматизированной ИС (АИС) понимают ИС, выполнение некоторых функций которой автоматизировано (с помощью электронной или любой иной техники). Степень автоматизации может быть разной. Наиболее легко автоматизируются функции этапов 1 и 5 (см. раздел 3). Более сложна автоматизация функций этапов 2 и 4, поскольку такая автоматизация может потребовать алгоритмизации и(или) моделирования некоторых составляющих и(или) аспектов основной деятельности.

Автоматической ИС (АВИС) называют ИС, в которой *все* функции выполняются автоматически. В настоящее время большей степенью автоматизации отличаются ИС технических систем и технологических процессов. В организационных системах, имеющих дело с неструктурируемыми проблемами, высокие уровни автоматизации ИС пока еще недостижимы. Из сказанного следует, что классификация ИС в современных условиях практически невозможна. Более целесообразно говорить о типологии ИС, характеризуя тип системы множеством существующих параметров, определяющих место конкретной ИС в множестве существующих и потенциально возможных ИС.

Исходя из определения ИС, как информационной системы, обеспечивающей функционирование системы информационного обмена, а также из рассмотрения специфических особенностей в составе и структуре ИС, изложенных в разделах 6 и 7, рассматривают следующие существенные параметры ИС:

- масштаб ИС (определяется масштабом СИОБ);
- область обслуживаемой СИОБ;
- характер решаемой ИС задач;
- совокупность выполняемых функций ИС;
- степень автоматизации функций ИС;
- характер (степень структурируемости) обрабатываемой информации;
- вид информации.

1. С точки зрения масштаба выделяют информационные системы: всемирные, международные, республиканские, региональные, отраслевые, объединений, предприятий и подразделений.

2. По отрасли (области) деятельности существуют информационные системы: медицины, транспорта, связи, строительства, отраслей промышленности, межотраслевые, ИС комплексных проблем, проблем, задач и т.д.

В настоящее время в мире существует ИС всех видов, перечисленных в пунктах 1 и 2.

В последние годы активно развиваются проблемно-ориентированные ИС (ПОИС).

3. С точки зрения характера решаемых ИС задач, выделяют:

3.1. Информационные системы информационного обеспечения и сопровождения основной деятельности. Эти системы составляют широкий спектр систем научно-технической информации (СНТИ), систем экономической информации (статистика, финансы, кадры, товары, услуги).

3.2. Информационные системы информационного обеспечения и сопровождения управления основной деятельностью. К ним относится огромное разнообразие автоматизированных систем управления (АСУ), АСУТП, систем обеспечения принятия решений (СОПР).

3.3. Информационные системы информационного обеспечения и сопровождения основной деятельности и управления основной деятельностью. Эти системы стали развиваться лишь в последние годы. Сложность создания интегрированных ИС (ИИС) связана с большими трудностями объединения ИС типов 3.1 и 3.2, базирующихся на достаточно различающихся технологиях обработки, хранения и поиска информации.

4. С точки зрения совокупности управляемых функций ИС, выделяют следующие их типы:

4.1. Системы справочно-информационного обеспечения, выполняющие функции собственно информационной деятельности (этапы 1 и 5 в разделе 4). Среди ИС этого типа существует определенная специализация по следующим функциям:

4.1.1. Подготовки и издания первичной информации (издательства, РИО, статистические бюро, пункты сбора и накопления информации и т.д.);

4.1.2. Обработки первичной информации с целью подготовки вторичной информации или метаинформации (генераторы баз данных, реферативные издательства и т.д.);

4.1.3. Обработки, хранения и поиска вторичной информации с целью информационного обслуживания (ОНТИ, бюро научно-технической информации (БНТИ), системы информационного обслуживания и т.д.);

4.1.4. Хранения и выдачи первичной информации (библиотеки и т.д.).

4.2. Системы информационного сопровождения основной деятельности, берущие на себя помимо функций этапов 1 и 5 (см. раздел 4), часть функций этапов 2 и 4, т.е. функций обобщенной основной деятельности (подготовка аналитических и тематических справок и обзоров, информационный анализ и синтез, анализ тенденций развития, оценка качества и технического уровня изделий, построение и анализ информационных моделей, подготовка вариантов решений и т.д.). К данному типу ИС относятся системы обеспечения принятия решений, системы автоматизации проектирования (САПР). Системы данного типа выполняют функции по математической, логической и даже интеллектуальной переработке информации. С этой точки зрения их еще называют информационно-логическими ИС. В литературе часто встречаются такие понятия как информационно-справочные системы (ИСИ), информационно-советующие системы (ИСС), информационно-управляющие системы (ИУС) и экспертные системы (ЭС).

Информационно-справочные системы – это системы, которые выдают только ту информацию, которая заранее в них введена.

ИСС и ИУС – это системы, способные выдавать новую информацию, являющуюся результатом переработки входной информации. В рамках введенной типологии информационно-справочные системы относятся к ИС типа 4.1, в то время как информационно-советующие и информационно-управляющие - к типу 4.2.

5. По степени автоматизации функций обычно рассматривают ИС трех типов: ручные (неавтоматизированные), автоматизированные и автоматические.

6. По составу и характеру перерабатываемой информации, предъявляющих жесткие требования к аппарату ее описания, организации и поиска, различают:

- документальные ИС (слабоструктурируемая информация);
- фактографические ИС (жесткоструктурированная информация);
- документально-фактографические ИС.

7. По видам обрабатываемой информации выделяют: ИС публикуемой информации и ИС непубликуемой информации.

### Тема 2.3. Состав и структура ИС

Исходя из функций, в информационных системах выделяют три самостоятельные функциональные подсистемы (рис. 2.1).

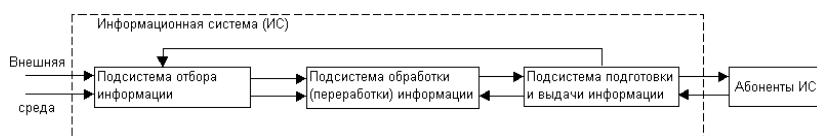


Рис. 2.1. Структурная схема процесса функционирования ИС

**1. Подсистема отбора информации.** ИС может обрабатывать (перерабатывать) только ту информацию, которая в нее введена. При этом качество работы ИС определяется не

только ее способностью находить и перерабатывать нужную информацию в собственном массиве и выдавать ее пользователю, но и способностью отбирать релевантную (см. раздел 2) информацию из внешней среды. Такой отбор осуществляет рассматриваемая подсистема, которая накапливает данные об информационных потребностях пользователей ИС (внутренних и внешних), анализирует и упорядочивает эти данные, образуя информационный профиль ИС. Аналогично на основании данных о потоках информационной среды формируется описание входных потоков информации. При заданном критерии качества функционирования ИС и соответствующей системы ограничений в процессе управления ИС решается задача оптимизации комплектования информационного массива ИС, которая определяет алгоритм отбора информации. Указанный алгоритм (оператор) осуществляет преобразование входных потоков в информационный массив ИС. Функции *именно этой* подсистемы ИС практически не поддаются автоматизации за исключением систем информационного обеспечения управления технологическими процессами и техническими системами.

**2. Подсистема ввода, обработки (переработки) и хранения информации.** Эта подсистема осуществляет преобразование входной информации и запросов, организацию их хранения и переработки с целью удовлетворения информационной потребности абонента ИС. Реализация функций данной подсистемы предполагает:

- наличие аппарата описания информации, а именно информационно-поискового языка (ИПЯ), систем кодирования и языка описания данных (ЯОД);
- организацию и ведение информации (логическая и физическая организация, процедуры ведения и защиты информации и т.д.);
- аппарата обработки и переработки информации (алгоритмы, модели и т.д.).

Все три указанные составляющие определяются двумя параметрами ИС:

- характером обработки информации;
- функциями ИС.

Если для описания информации в документальных ИС используют ИПЯ и системы индексирования, то в фактографических - ЯОД, методологии построения и использования которых существенно отличаются друг от друга. Кроме того, различны и аппараты обработки и переработки документальной и фактографической информации. Если в фактографических ИС преимущественно используются математические алгоритмы, то в документальных - эвристические процедуры, требующие затрат интеллектуальной энергии. (Эвристика – наука, занимающаяся построением эмпирических правил с помощью которых человек (эксперт) реализует поставленные цели при отсутствии жесткого алгоритма или формул).

**3. Подсистема подготовки и выдачи информации.** Эта подсистема непосредственно реализует удовлетворение информационных потребностей как внутренних, так и внешних пользователей информационных систем. Для выполнения этой задачи подсистема проводит изучение и анализ информационных потребностей, определяет формы и методы их удовлетворения, оптимальный состав и структуру выходных информационных продуктов, организует сам процесс информационного обеспечения и сопровождения. Для выполнения этих функций требуется аппарат описания и анализа информационных потребностей и их выражения на языке ИС (в том числе ЯОД, ИПЯ, язык индексирования и т.д.); аппарат непосредственно информационного обеспечения (процедуры поиска и выдачи информации, языки манипулирования данными и т.д.). Следует отметить, что при одинаковости выполнения функций этой подсистемой в ИС разных типов они существенно отличаются между собой. Особенно это заметно при сравнении документальных и фактографических ИС.

При автоматизации функций ИС используют формализованное представление как самих функций, так и обрабатываемой в ИС информации. Однако любая формализация (структуризация) характеризуется тем или иным уровнем адекватности создаваемого образца реальной действительности (модели) в самой действительности.

Различают три уровня структурируемости информации:

1. Жесткоструктурируемая информация - информация, формализованное представление которой современными средствами ее структурирования (в частности языками описания



данных) не приводит к потере адекватности создаваемого образа информации (модели) самой исходной информации (например, таблицы, графики). Такую информацию называют данными.

2. Слабоструктурируемая информация - информация, формализованное представление которой современными средствами описания информации (в частности, ИПЯ) приводит к значительной потере адекватности модели информации самой исходной информации (например тексты статьи, книги). Обработка и поиск такой информации предполагает специальные меры по оценке степени адекватности модели информации (в автоматизированных информационно-поисковых системах этой цели служат меры смысло-выразительной способности, т.е. семантической силы ИПЯ).

3. Неструктурируемая информация - информация, для которой в настоящее время не существует средств и форм представления с приемлемым на практике уровнем адекватности (например, картины, скульптуры и т.п.). Средства представления такой информации должны обладать высокими смысло-выразительными способностями. Разработка таких средств в настоящее время идет по линии создания языков описания знаний и ИПЯ с высокой семантической силой.

С точки зрения возможностей алгоритмизации, функции ИС разбивают на три категории, аналогичные уровням структурируемости информации, т.е. на:

- жесткоструктурируемые (ввод, обработка и хранение информации), выполнение которых не требует затрат интеллектуального труда и которые легко алгоритмируются;

- слабоструктурируемые (поиск и переработка информации, учет и анализ потребностей, анализ потоков информации и т.д.), выполнение которых можно алгоритмизировать без существенных потерь в качестве их выполнения;

- неструктурируемые, и требующие для своего выполнения существенных затрат интеллектуальной энергии, алгоритмизация которых приводит к невосполнимым потерям в качестве их выполнения. Эти функции связаны с научным анализом и/или синтезом информации, с оценкой уровня продукции и прогнозированием развития различных направлений науки и техники.

С рассмотренных позиций следует, что большинство жесткоструктурируемых функций сосредоточено в подсистеме ввода, обработки и хранения информации. Две другие подсистемы связаны с реализацией, в основном, слабо- и неструктурируемых функций. Легкость автоматизации функций второй подсистемы ИС на основе использования электронно-вычислительной и телекоммуникационной техники ввода, обработки, хранения и передачи информации привела к неоправданно быстрому развитию этих составляющих ИС в ущерб развитию двух других не менее, а может быть и более важных ее составляющих. Следует отметить, что в большинстве современных АИС эти две подсистемы настолько неразвиты, что по сути дела это уже не АИС, а организационно-обособленные подсистемы ввода, хранения, обработки и поиска информации. А потому их целесообразно называть не АИС, а банки данных или АИПС.

#### **Тема 2.4. АИПС, банки и базы данных**

Учитывая то, что «степень структурируемости» информации является весьма важным при выяснении единства и различий существующих АИС, в литературе предлагаются следующие определения элементов АИС:

**База данных (БД)** – совокупность специально организованных и логически взаимосвязанных элементов фактографической и/или документальной информации, представленных на машиночитаемых носителях.

**Система управления базами данных (СУБД)** – комплекс языковых, логических и программных средств, предназначенных для описания, ввода, переработки, ведения, хранения и выборки специально организованной и логически взаимосвязанной жесткоструктурируемой информации в целях их многоаспектного коллективного использования.

**Банк данных (БД)** – комплекс языковых, логических, программных, информационных (в том числе базы данных); технических и организационных средств ввода, переработки, ведения, хранения, поиска и выдачи информации в целях ее многоаспектного коллективного использования.

**Автоматизированная информационно-поисковая система (АИПС)** – взаимосвязанная совокупность критерия смыслового соответствия, языковых, логических, программных, информационных, технических и организационных средств ввода, переработки, ведения, хранения, поиска и выдачи специально организованной и логически взаимосвязанной слабоструктурируемой информации в целях ее многоаспектного коллективного использования.

**Фактографическая АИПС** – то же, что и банк данных, но по обработке жесткоструктурируемой информации.

Программная система – комплекс программ реализации на ЭВМ некоторой совокупности информационных процессов ввода, переработки, ведения, хранения, выборки и поиска слабоструктурируемой информации, подготовки и выдачи выходных информационных продуктов, т.е. это пакет прикладных программ реализации технологических процессов информационной системы.

Из рассмотренных определений следует:

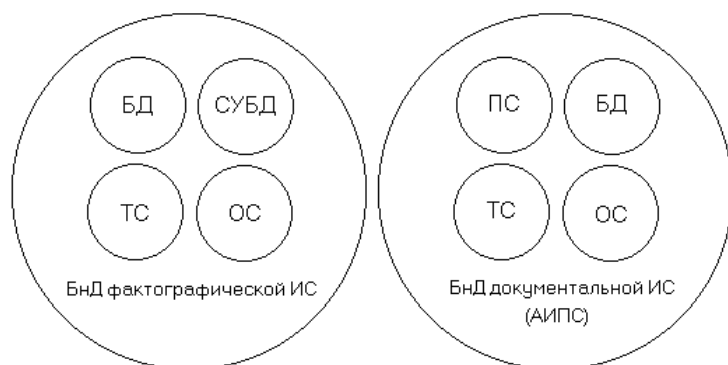
1. Банк данных и АИПС являются основными составляющими АИС в целом, с той лишь разницей, что банк данных оперирует с жесткоструктурируемой информацией (данными), а АИПС – со слабоструктурируемой.

2. База данных – основной элемент как банка данных, так и АИПС, и может содержать как жесткоструктурируемую (при работе в составе банка данных), так и слабоструктурируемую информацию (при работе в составе АИПС).

3. СУБД осуществляет управление базами данных, содержащими жесткоструктурируемую (в случае банка данных) или слабоструктурируемую информацию (в случае АИПС).

4. Определенные различия между банком данным и АИПС порождаются различиями в языковых и логических средствах *адекватного* представления слабо- и жесткоструктурируемой информации

Рис. 2.2. иллюстрирует взаимосвязь приведенных понятий.



ТС – технические средства,  
ОС – организационные средства,  
ПС – программная система.

Рис. 2.2. Состав банка данных

5. Из сказанного в п. 1 вовсе не следует, что банк данных не может работать со слабоструктурируемой информацией, а АИПС – с жесткоструктурируемой. Обе они в любом случае, работают в той или иной мере со *структурированной*, т.е. формализованно представленной информацией, но их аппараты, т.е. языковые и логические средства структурирования, различны. Так, применение аппарата банка данных для представления слабоструктурируемой информации приводит к *определенному снижению* эффективности

обработки и поиска информации. К аналогичным последствиям приводит и применение АИПС для обработки и поиска жесткоструктурируемой информации.

6. Поскольку в определении банка данных нет упоминания о структурируемости информации, то банк данных может иметь дело и со слабоструктурируемой информацией. С этой точки зрения АИПС – это тоже банк данных, но связанный с обработкой слабоструктурируемой информации. Поэтому, если необходимо подчеркнуть характер обрабатываемой информации, то говорят о документальной АИПС или фактографической ФИПС.

Тенденция современного развития АИС требует создания интегрированных систем, способных одинаково эффективно оперировать как жесткоструктурируемой, так и со слабоструктурируемой информацией. Это предполагает создание логических и языковых средств *адекватного и эффективного* представления информации независимо от степени ее структурируемости.

## Тема 2.5. Информационные сети

Сеть информационного обмена – это совокупность взаимодействующих информационных систем, предполагающая:

- существование канала связи между ИС, обеспечивающего физическую возможность передачи исходных, промежуточных и конечных продуктов ИС;
- существование договоренности между ИС, регламентирующей процессы их функционирования.

При объединении ИС в сеть эффективность информационной деятельности повышается за счет устранения непроизводительного дублирования процессов обработки и хранения информации. В отличие от ИС информационная сеть может не иметь единого органа управления ее функционированием.

Различают однородные и неоднородные сети. Сеть считается однородной, если все составляющие ее ИС выполняют одинаковые функции, имеют одни и те же выходы, обладают одинаковыми статусами. Сеть неоднородна, если каждой составляющей ее ИС свойственны собственные функции, определенные выходы и статус. Если существует иерархия статусов, то говорят об иерархической сети. Частным случаем неоднородной сети является сеть с центральной ИС и совокупностью ее филиалов.

Сеть с *одним* процессом, установленным в центральной ИС, и совокупностью удаленных терминалов, называют вырожденной. Следует отметить, что каждый из вариантов взаимодействия ИС возможен только при обеспечении соответствующих форм совместимости – информационной, языковой, программной и технической (одной или нескольких одновременно). В противном случае требуется разработка соответствующих «интерпретаторов» и собственно процесса «интерпретации».

## Тема 2.6. Обеспечивающие подсистемы информационных систем и их характеристика

Любую ИС, независимо от сферы применения, можно представить в виде совокупности обеспечивающих подсистем, среди которых выделяют: информационное, техническое, математическое, программное, организационное и правовое обеспечение.

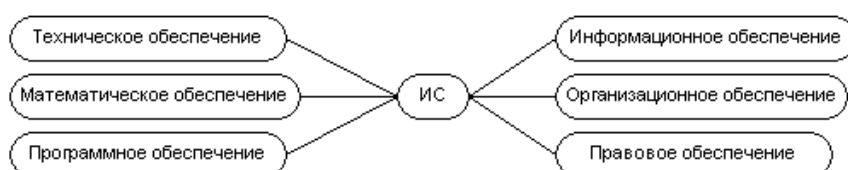


Рис. 3.

## **Информационное обеспечение**

Информационное обеспечение – это совокупность:

- единой системы классификации и кодирования информации;
- унифицированных систем документации (УСД);
- схем информационных потоков, циркулирующих в организации;
- методологии построения баз данных.

*Унифицированные системы документации* создаются на государственном, отраслевом и региональном уровнях. Главная цель – обеспечение сопоставимости показателей различных сфер общественного производства. Однако, несмотря на унифицированные системы документации, при обслуживании большинства организаций, постоянно выявляется целый комплекс недостатков:

- чрезвычайно большой объем документов для ручной обработки;
- одни и те же показатели часто дублируются в разных документах;
- работа с большим количеством документов отвлекает специалистов от решения непосредственных задач;
- имеются показатели, которые создаются, но не используются.

Поэтому устранение указанных недостатков является одной из важных задач, стоящих при создании информационного обеспечения ИС.

*Схемы информационных потоков* отражают маршруты движения информации и ее объемы, места возникновения первичной информации и использования результативной информации. За счет анализа структуры подобных схем можно выработать меры по совершенствованию всей системы управления.

Примером простейшей схемы потоков данных является схема, отражающая все этапы прохождения служебной записки или записи в БД о приеме сотрудника на работу – от момента ее создания до выхода приказа о его зачислении на работу.

Построение схем информационных потоков, позволяющих выявить объемы информации и провести ее детальный анализ, обеспечивают:

- исключение дублирующей и неиспользуемой информации;
- классификацию и рациональное представление информации.

При этом подробно должны рассматриваться вопросы взаимосвязи движения информации по уровням управления (см. рис. 1.1), а также выявляться необходимые для принятия управленческих решений показатели, т.к. к каждому исполнителю должна поступать только та информация, которая используется.

*Методология построения баз данных* базируется на теоретических основах их проектирования и реализуется на практике в виде двух последовательных этапов.

*Первый этап* – обследование всех функциональных подразделений фирмы с целью:

- понять специфику и структуру ее деятельности;
- построить схему информационных потоков;
- проанализировать существующую систему документооборота;
- определить информационные объекты и соответствующий состав реквизитов (параметров, характеристик), описывающих их свойства и назначение.

*Второй этап* – построение концептуальной информационно-логической модели данных для обследованной на первом этапе сферы деятельности. В этой модели должны быть установлены и оптимизированы все связи между объектами и их реквизитами. Информационно-логическая модель является фундаментом, на котором создается база данных. Таким образом, для создания информационного обеспечения необходимо:

- ясное понимание цели, задач, функций всей системы управления организацией;
- выявление движения информации от момента возникновения и до ее использования на различных уровнях управления, представленной для анализа в виде схем информационных потоков;
- совершенствование системы документооборота;
- наличие и использование системы классификации и кодирования;
- владение методологией создания концептуальных информационно-логических моделей, отражающих взаимосвязь информации;

- создание массивов информации на машинных носителях, что требует наличия современного технического обеспечения.

### **Техническое обеспечение**

Техническое обеспечение - это комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы. Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линии связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы.

Документацией оформляется предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных и технологическое оснащение. Документацию условно делят на три группы:

- общесистемную, включающую государственные отраслевые стандарты по техническому обеспечению;
- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;
- нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

### **Математическое и программное обеспечение**

Математическое и программное обеспечение - это совокупность математических методов, моделей, алгоритмов и программ для решения целей и задач информационной системы, а также нормального функционирования комплекса технических средств. К средствам математического обеспечения относят:

- средства моделирования процесса управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

В состав программного обеспечения входят общесистемные и специальные программные продукты, а также техническая документация. К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных. Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ (ППП), реализующие *разработанные модели* разной степени адекватности, отражающие функционирование реального объекта. Техническая документация на разработку программных средств должна содержать описание задач, задание на алгоритмизацию, экономико-математическую модель задачи и контрольные примеры.

### **Организационное обеспечение**

Организационное обеспечение - это совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационных систем. Это обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;

- разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на первом этапе построения БД, цели которого рассмотрены в информационном обеспечении.

### **Правовое обеспечение**

Правовое обеспечение – это совокупность правовых норм, определяющих создание, юридический статус и функционирование ИС, регламентирующих порядок получения, преобразования и использования информации. Главной целью этого обеспечения является укрепление законности. В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении выделяют:

- общую часть, регулирующую функционирование любой информационной системы;
- локальную часть, регулирующую функционирование конкретной ИС.

Правовое обеспечение этапов функционирования ИС включает:

- статус ИС;
- права, обязанности и ответственность персонала;
- правовые положения отдельных видов процесса управления;
- порядок создания и использования информации и др.

## **Раздел 3. ЖИЗНЕННЫЙ ЦИКЛ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде жизненного цикла (ЖЦ) ИС, представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Жизненный цикл ИС можно представить как ряд событий, происходящих с системой в процессе ее создания и использования.

### **Тема 3.1. Модели жизненного цикла (ЖЦ) ИС**

Модель жизненного цикла отражает различные состояния системы, начиная с момента возникновения необходимости в данной ИС и заканчивая моментом ее полного выхода из употребления. *Модель жизненного цикла* - структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования.

В настоящее время известны и используются следующие модели жизненного цикла:

- *каскадная модель* (рис. 3.1) предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе;

- *поэтапная модель* с промежуточным контролем (рис. 3.2). Разработка ИС ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки;

- *спиральная модель* (рис. 3.3). На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и

планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки - анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).



Рис. 3.1. Каскадная модель ЖЦ ИС



Рис. 3.2. Поэтапная модель с промежуточным контролем

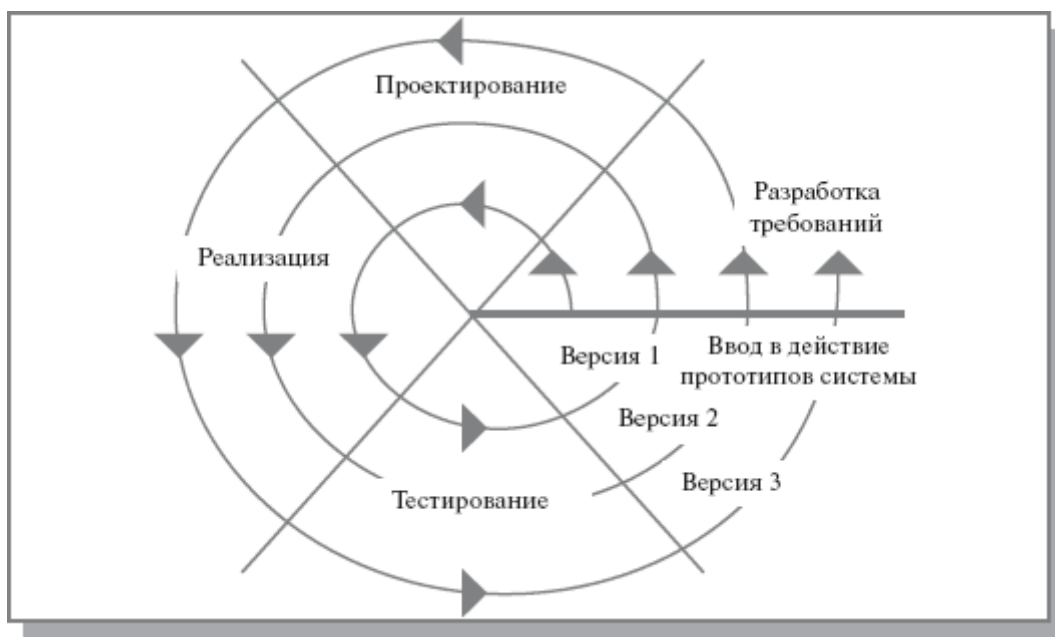


Рис. 3.3. Спиральная модель ЖЦ ИС

На практике наибольшее распространение получили две основные модели жизненного цикла:

- каскадная модель (характерна для периода 1970-1985 гг.);
- спиральная модель (характерна для периода после 1986 г.).

В ранних проектах достаточно простых ИС каждое приложение представляло собой единый, функционально и информационно независимый блок. Для разработки такого типа приложений эффективным оказался каскадный способ. Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.

*Можно выделить следующие положительные стороны применения каскадного подхода:*

- на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Каскадный подход хорошо зарекомендовал себя при построении относительно простых ИС, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования к системе. Основным *недостатком этого подхода является* то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания ИС оказывается соответствующим поэтапной модели с промежуточным контролем.

Однако и эта схема не позволяет оперативно учитывать возникающие изменения и уточнения требований к системе. Согласование результатов разработки с пользователями производится только в точках, планируемых после завершения каждого этапа работ, а общие требования к ИС зафиксированы в виде технического задания на все время ее создания. Таким образом, пользователи зачастую получают систему, не удовлетворяющую их реальным потребностям.

*Спиральная модель ЖЦ* была предложена для преодоления перечисленных проблем. На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется путем создания прототипов. Каждый виток спирали соответствует созданию работоспособного фрагмента или версии системы. Это позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали. Таким образом углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который удовлетворяет действительным требованиям заказчика и доводится до реализации.

*Итеративная разработка* отражает объективно существующий спиральный цикл создания сложных систем. Она позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем и решить главную задачу - как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

*Основная проблема спирального цикла* - определение момента перехода на следующий этап. Для ее решения вводятся временные ограничения на каждый из этапов жизненного цикла, и переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. Планирование производится на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

Несмотря на настойчивые рекомендации компаний - вендоров и экспертов в области проектирования и разработки ИС, многие компании продолжают использовать каскадную модель вместо какого-либо варианта итерационной модели. Основные причины, по которым каскадная модель сохраняет свою популярность, следующие:

1. **Привычка** - многие ИТ-специалисты получали образование в то время, когда изучалась только каскадная модель, поэтому она используется ими и в наши дни.

2. **Иллюзия снижения рисков** участников проекта (заказчика и исполнителя). Каскадная модель предполагает разработку законченных продуктов на каждом этапе: технического задания, технического проекта, программного продукта и пользовательской документации. Разработанная документация позволяет не только определить требования к продукту следующего этапа, но и определить обязанности сторон, объем работ и сроки, при этом окончательная оценка сроков и стоимости проекта производится на начальных этапах, после завершения обследования. Очевидно, что если требования к информационной системе меняются в ходе реализации проекта, а качество документов оказывается невысоким (требования неполны и/или противоречивы), то в действительности использование каскадной модели создает лишь иллюзию определенности и на деле увеличивает риски, уменьшая лишь ответственность участников проекта. При формальном подходе менеджер



проекта реализует только те требования, которые содержатся в спецификации, опирается на документ, а не на реальные потребности бизнеса. Есть два основных типа контрактов на разработку ПО. Первый тип предполагает выполнение определенного объема работ за определенную сумму в определенные сроки (*fixed price*). Второй тип предполагает повременную оплату работы (*time work*). Выбор того или иного типа контракта зависит от степени определенности задачи. Каскадная модель с определенными этапами и их результатами лучше приспособлена для заключения контракта с оплатой по результатам работы, а именно этот тип контрактов позволяет получить полную оценку стоимости проекта до его завершения. Более вероятно заключение контракта с повременной оплатой на небольшую систему, с относительно небольшим весом в структуре затрат предприятия. Разработка и внедрение интегрированной информационной системы требует существенных финансовых затрат, поэтому используются контракты с фиксированной ценой, и, следовательно, каскадная модель разработки и внедрения. Спиральная модель чаще применяется при разработке информационной системы силами собственного отдела ИТ предприятия.

**3. Проблемы внедрения** при использовании итерационной модели. В некоторых областях спиральная модель не может применяться, поскольку невозможно использование/тестирование продукта, обладающего неполной функциональностью (например, военные разработки, атомная энергетика и т.д.). Поэтапное итерационное внедрение информационной системы для бизнеса возможно, но сопряжено с организационными сложностями (перенос данных, интеграция систем, изменение бизнес-процессов, учетной политики, обучение пользователей). Трудозатраты при поэтапном итерационном внедрении оказываются значительно выше, а управление проектом требует настоящего искусства. Предвидя указанные сложности, заказчики выбирают каскадную модель, чтобы «внедрять систему один раз».

### **Тема 3.2. Стандарты на проектирование ИС**

Каждая из стадий создания системы предусматривает выполнение определенного объема работ, которые представляются в виде процессов ЖЦ. Процесс определяется как совокупность взаимосвязанных действий, преобразующих входные данные в выходные. Описание каждого процесса включает в себя перечень решаемых задач, исходных данных и результатов.

Существует целый ряд стандартов, регламентирующих ЖЦ ПО, а в некоторых случаях и процессы разработки.

Значительный вклад в теорию проектирования и разработки информационных систем внесла компания IBM, предложив еще в середине 1970-х годов методологию BSP (Business System Planning - методология организационного планирования). Метод структурирования информации с использованием матриц пересечения бизнес-процессов, функциональных подразделений, функций систем обработки данных (информационных систем), информационных объектов, документов и баз данных, предложенный в BSP, используется сегодня не только в ИТ-проектах, но и проектах по реинжинирингу бизнес-процессов, изменению организационной структуры. Важнейшие шаги процесса BSP, их последовательность (получить поддержку высшего руководства, определить процессы предприятия, определить классы данных, провести интервью, обработать и организовать данные интервью) можно встретить практически во всех формальных методиках, а также в проектах, реализуемых на практике.

Среди наиболее известных стандартов можно выделить следующие:

- ГОСТ 34.601-90 - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится описание содержания работ на каждом этапе. Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют каскадной модели жизненного цикла;

- ISO/IEC 12207:1995 - стандарт на процессы и организацию жизненного цикла. Распространяется на все виды заказного ПО. Стандарт не содержит описания фаз, стадий и этапов;

- Custom Development Method (методика Oracle) по разработке прикладных информационных систем - технологический материал, детализированный до уровня заготовок проектных документов, рассчитанных на использование в проектах с применением Oracle. Применяется CDM для классической модели ЖЦ (предусмотрены все работы/задачи и этапы), а также для технологий «быстрой разработки» (Fast Track) или «облегченного подхода», рекомендуемых в случае малых проектов;

- Rational Unified Process (RUP) предлагает итеративную модель разработки, включающую четыре фазы: начало, исследование, построение и внедрение. Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования. Прохождение через четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы. Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы. Суть работы в рамках RUP - это создание и сопровождение моделей на базе UML [6];

- Microsoft Solution Framework (MSF) сходна с RUP, так же включает четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования. MSF в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений;

- Extreme Programming (XP). Экстремальное программирование (самая новая среди рассматриваемых методологий) сформировалось в 1996 году. В основе методологии командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС, а разработка ведется с использованием последовательно дорабатываемых прототипов.

### Тема 3.3. Процессы ЖЦ ПО

В соответствии с базовым международным стандартом ISO/IEC 12207 все процессы ЖЦ ПО делятся на три группы:

#### 1) Основные процессы:

- а) приобретение;
- б) поставка;
- в) разработка;
- г) эксплуатация;
- д) сопровождение.

#### 2) Вспомогательные процессы:

- а) документирование;
- б) управление конфигурацией;
- в) обеспечение качества;
- г) разрешение проблем;
- д) аудит;
- е) аттестация;
- ж) совместная оценка;
- з) верификация.

#### 3) Организационные процессы:

- а) создание инфраструктуры;
- б) управление;
- в) обучение;
- г) усовершенствование.

В таблице 3.1 приведены ориентировочные описания основных процессов ЖЦ. Вспомогательные процессы предназначены для поддержки выполнения основных процессов, обеспечения качества проекта, организации верификации, проверки и тестирования ПО.

Организационные процессы определяют действия и задачи, выполняемые как заказчиком, так и разработчиком проекта для управления своими процессами.

Для поддержки практического применения стандарта ISO/IEC 12207 разработан ряд технологических документов: Руководство для ISO/IEC 12207 (ISO/IEC TR 15271:1998 Information technology - Guide for ISO/IEC 12207) и Руководство по применению ISO/IEC 12207 к управлению проектами (ISO/IEC TR 16326:1999 Software engineering - Guide for the application of ISO/IEC 12207 to project management).

Таблица 3.1 – Содержание основных процессов ЖЦ ПО ИС (ISO/IEC 12207)

Процесс (исполнитель процесса)	Действия	Вход	Результат
Приобретение (заказчик)	<ul style="list-style-type: none"> <li>- Инициирование</li> <li>- Подготовка заявочных предложений</li> <li>- Подготовка договора</li> <li>- Контроль деятельности поставщика</li> <li>- Приемка ИС</li> </ul>	<ul style="list-style-type: none"> <li>- Решение о начале работ по внедрению ИС</li> <li>- Результаты обследования деятельности заказчика</li> <li>- Результаты анализа рынка ИС/ тендера</li> <li>- План поставки/ разработки</li> <li>- Комплексный тест ИС</li> </ul>	<ul style="list-style-type: none"> <li>- Техничко-экономическое обоснование внедрения ИС</li> <li>- Техническое задание на ИС</li> <li>- Договор на поставку/ разработку</li> <li>- Акты приемки этапов работы</li> <li>- Акт приемно-сдаточных испытаний</li> </ul>
Поставка (разработчик ИС)	<ul style="list-style-type: none"> <li>- Инициирование</li> <li>- Ответ на заявочные предложения</li> <li>- Подготовка договора</li> <li>- Планирование исполнения</li> <li>- Поставка ИС</li> </ul>	<ul style="list-style-type: none"> <li>- Техническое задание на ИС</li> <li>- Решение руководства об участии в разработке</li> <li>- Результаты тендера</li> <li>- Техническое задание на ИС</li> <li>- План управления проектом</li> <li>- Разработанная ИС и документация</li> </ul>	<ul style="list-style-type: none"> <li>- Решение об участии в разработке</li> <li>- Коммерческие предложения/ конкурсная заявка</li> <li>- Договор на поставку/ разработку</li> <li>- План управления проектом</li> <li>- Реализация/ корректировка</li> <li>- Акт приемно-сдаточных испытаний</li> </ul>
Разработка (разработчик ИС)	<ul style="list-style-type: none"> <li>- Подготовка</li> <li>- Анализ требований к ИС</li> <li>- Проектирование архитектуры ИС</li> <li>- Разработка требований к ПО</li> <li>- Проектирование архитектуры ПО</li> <li>- Детальное проектирование ПО</li> <li>- Кодирование и тестирование ПО</li> <li>- Интеграция ПО и квалификационное тестирование ПО</li> <li>- Интеграция ИС и квалификационное тестирование ИС</li> </ul>	<ul style="list-style-type: none"> <li>- Техническое задание на ИС</li> <li>- Техническое задание на ИС, модель ЖЦ</li> <li>- Подсистемы ИС</li> <li>- Спецификации требования к компонентам ПО</li> <li>- Архитектура ПО</li> <li>- Материалы детального проектирования ПО</li> <li>- План интеграции ПО, тесты</li> <li>- Архитектура ИС, ПО, документация на ИС, тесты</li> </ul>	<ul style="list-style-type: none"> <li>- Используемая модель ЖЦ, стандарты разработки</li> <li>- План работ</li> <li>- Состав подсистем, компоненты оборудования</li> <li>- Спецификации требования к компонентам ПО</li> <li>- Состав компонентов ПО, интерфейсы с БД, план интеграции ПО</li> <li>- Проект БД, спецификации интерфейсов между компонентами ПО, требования к тестам</li> <li>- Тексты модулей ПО, акты автономного тестирования</li> <li>- Оценка соответствия комплекса ПО требованиям ТЗ</li> <li>- Оценка соответствия ПО, БД, технического комплекса и комплекта документации требованиям ТЗ</li> </ul>

Позднее был разработан и в 2002 г. опубликован стандарт на процессы жизненного цикла систем (ISO/IEC 15288 System life cycle processes). К разработке стандарта были привлечены специалисты различных областей: системной инженерии, программирования, управления качеством, человеческими ресурсами, безопасностью и пр. Был учтен практический опыт создания систем в правительственных, коммерческих, военных и

академических организациях. Стандарт применим для широкого класса систем, но его основное предназначение - поддержка создания компьютеризированных систем.

Согласно стандарту ISO/IEC серии 15288 [7] в структуру ЖЦ следует включать следующие группы процессов:

- 1) **Договорные процессы:**
  - а) приобретение (внутренние решения или решения внешнего поставщика);
  - б) поставка (внутренние решения или решения внешнего поставщика).
- 2) **Процессы предприятия:**
  - в) управление окружающей средой предприятия;
  - г) инвестиционное управление;
  - д) управление ЖЦ ИС;
  - е) управление ресурсами;
  - ж) управление качеством.
- 3) **Проектные процессы:**
  - з) планирование проекта;
  - и) оценка проекта;
  - к) контроль проекта;
  - л) управление рисками;
  - м) управление конфигурацией;
  - н) управление информационными потоками;
  - о) принятие решений.
- 4) **Технические процессы:**
  - п) определение требований;
  - р) анализ требований;
  - с) разработка архитектуры;
  - т) внедрение;
  - у) интеграция;
  - ф) верификация;
  - х) переход;
  - ц) аттестация;
  - ч) эксплуатация;
  - ш) сопровождение;
  - щ) утилизация.
- 5) **Специальные процессы:**
  - ы) определение и установка взаимосвязей исходя из задач и целей.

Стадии создания системы, предусмотренные в стандарте ISO/IEC 15288, несколько отличаются от рассмотренных выше. Перечень стадий и основные результаты, которые должны быть достигнуты к моменту их завершения, приведены в таблице 3.2.

Таблица 3.2. – Стадии создания систем (ISO/IEC 15288)

№ п/п	Стадия	Описание
1	Формирование концепции	Анализ потребностей, выбор концепции и проектных решений
2	Разработка	Проектирование системы
3	Реализация	Изготовление системы
4	Эксплуатация	Ввод в эксплуатацию и использование системы
5	Поддержка	Обеспечение функционирования системы
6	Снятие с эксплуатации	Прекращение использования, демонтаж, архивирование системы

### Тема 3.4. Каноническое проектирование ИС

Организация канонического проектирования ИС ориентирована на использование главным образом каскадной модели жизненного цикла ИС. Стадии и этапы работы описаны в стандарте ГОСТ 34.601-90.

В зависимости от сложности объекта автоматизации и набора задач, требующих решения при создании конкретной ИС, стадии и этапы работ могут иметь различную трудоемкость. Допускается объединять последовательные этапы и даже исключать некоторые из них на любой стадии проекта. Допускается также начинать выполнение работ следующей стадии до окончания предыдущей.

Стадии и этапы создания ИС, выполняемые организациями-участниками, прописываются в договорах и технических заданиях на выполнение работ:

#### Стадия 1. Формирование требований к ИС.

На начальной стадии проектирования выделяют следующие этапы работ:

- обследование объекта и обоснование необходимости создания ИС;
- формирование требований пользователей к ИС;
- оформление отчета о выполненной работе и тактико-технического задания на разработку.

#### Стадия 2. Разработка концепции ИС.

- изучение объекта автоматизации;
- проведение необходимых научно-исследовательских работ;
- разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей;
- оформление отчета и утверждение концепции.

#### Стадия 3. Техническое задание.

- разработка и утверждение технического задания на создание ИС.

#### Стадия 4. Эскизный проект.

- разработка предварительных проектных решений по системе и ее частям;
- разработка эскизной документации на ИС и ее части.

#### Стадия 5. Технический проект.

- разработка проектных решений по системе и ее частям;
- разработка документации на ИС и ее части;
- разработка и оформление документации на поставку комплектующих изделий;
- разработка заданий на проектирование в смежных частях проекта.

#### Стадия 6. Рабочая документация.

- разработка рабочей документации на ИС и ее части;
- разработка и адаптация программ.

#### Стадия 7. Ввод в действие.

- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
- строительные-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации;
- проведение приемочных испытаний.

#### Стадия 8. Сопровождение ИС.

- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

*Обследование* – это изучение и диагностический анализ организационной структуры предприятия, его деятельности и существующей системы обработки информации. Материалы, полученные в результате обследования, используются для:

- обоснования разработки и поэтапного внедрения систем;
- составления технического задания на разработку систем;

- разработки технического и рабочего проектов систем.

*На этапе обследования целесообразно выделить две составляющие:* определение стратегии внедрения ИС и детальный анализ деятельности организации.

Основная задача первого этапа обследования - оценка реального объема проекта, его целей и задач на основе выявленных функций и информационных элементов автоматизируемого объекта высокого уровня. Эти задачи могут быть реализованы или заказчиком ИС самостоятельно, или с привлечением консалтинговых организаций. Этап предполагает тесное взаимодействие с основными потенциальными пользователями системы и бизнес-экспертами. Основная задача взаимодействия - получить полное и однозначное понимание требований заказчика. Как правило, нужная информация может быть получена в результате интервью, бесед или семинаров с руководством, экспертами и пользователями.

По завершении этой стадии обследования появляется возможность определить вероятные технические подходы к созданию системы и оценить затраты на ее реализацию (затраты на аппаратное обеспечение, закупаемое программное обеспечение и разработку нового программного обеспечения).

*Результатом этапа определения стратегии* является документ (техничко-экономическое обоснование проекта), где четко сформулировано, что получит заказчик, если согласится финансировать проект, когда он получит готовый продукт (график выполнения работ) и сколько это будет стоить (для крупных проектов должен быть составлен график финансирования на разных этапах работ). В документе желательно отразить не только затраты, но и выгоду проекта, например время окупаемости проекта, ожидаемый экономический эффект (если его удастся оценить).

Ориентировочное содержание этого документа:

- ограничения, риски, критические факторы, которые могут повлиять на успешность проекта;
- совокупность условий, при которых предполагается эксплуатировать будущую систему: архитектура системы, аппаратные и программные ресурсы, условия функционирования, обслуживающий персонал и пользователи системы;
- сроки завершения отдельных этапов, форма приемки / сдачи работ, привлекаемые ресурсы, меры по защите информации;
- описание выполняемых системой функций;
- возможности развития системы;
- информационные объекты системы;
- интерфейсы и распределение функций между человеком и системой;
- требования к программным и информационным компонентам ПО, требования к СУБД;
- что не будет реализовано в рамках проекта.

*На этапе детального анализа деятельности организации* изучаются задачи, обеспечивающие реализацию функций управления, организационная структура, штаты и содержание работ по управлению предприятием, а также характер подчиненности вышестоящим органам управления. На этом этапе должны быть выявлены:

- инструктивно-методические и директивные материалы, на основании которых определяются состав подсистем и перечень задач;
- возможности применения новых методов решения задач.
- аналитики собирают и фиксируют информацию в двух взаимосвязанных формах:
- функции - информация о событиях и процессах, которые происходят в бизнесе;
- сущности - информация о вещах, имеющих значение для организации и о которых что-то известно.

При изучении каждой функциональной задачи управления определяются:

- наименование задачи; сроки и периодичность ее решения;
- степень формализуемости задачи;
- источники информации, необходимые для решения задачи;
- показатели и их количественные характеристики;
- порядок корректировки информации;
- действующие алгоритмы расчета показателей и возможные методы контроля;

- действующие средства сбора, передачи и обработки информации;
- действующие средства связи;
- принятая точность решения задачи;
- трудоемкость решения задачи;
- действующие формы представления исходных данных и результатов их обработки в виде документов;
- потребители результатной информации по задаче.

Одной из наиболее трудоемких, хотя и хорошо формализуемых задач этого этапа является *описание документооборота организации*. При обследовании документооборота составляется схема маршрута движения документов, которая должна отразить:

- количество документов;
- место формирования показателей документа;
- взаимосвязь документов при их формировании;
- маршрут и длительность движения документа;
- место использования и хранения данного документа;
- внутренние и внешние информационные связи;
- объем документа в знаках.

*По результатам обследования* устанавливается перечень задач управления, решение которых целесообразно автоматизировать, и очередность их разработки.

На этапе обследования следует классифицировать планируемые функции системы по степени важности. Один из возможных форматов представления такой классификации - MuSCoW. Эта аббревиатура расшифровывается так: Must have - необходимые функции; Should have - желательные функции; Could have - возможные функции; Won't have - отсутствующие функции.

Функции первой категории обеспечивают критичные для успешной работы системы возможности.

Реализация функций второй и третьей категорий ограничивается временными и финансовыми рамками: разрабатывается то, что необходимо, а также максимально возможное в порядке приоритета число функций второй и третьей категорий.

Последняя категория функций особенно важна, поскольку необходимо четко представлять границы проекта и набор функций, которые будут отсутствовать в системе.

*Модели деятельности организации создаются в двух видах:*

- модель «как есть» («as-is») - отражает существующие в организации бизнес-процессы;
- модель «как должно быть» («to-be») - отражает необходимые изменения бизнес-процессов с учетом внедрения ИС.

На этапе анализа необходимо привлекать к работе группы тестирования для решения следующих задач:

- получения сравнительных характеристик предполагаемых к использованию аппаратных платформ, операционных систем, СУБД, иного окружения;
- разработки плана работ по обеспечению надежности информационной системы и ее тестирования.

Привлечение тестировщиков на ранних этапах разработки является целесообразным для любых проектов. Если проектное решение оказалось неудачным и это обнаружено слишком поздно (на этапе разработки или, что еще хуже, на этапе внедрения в эксплуатацию), то исправление ошибки проектирования обходится очень дорого. Чем раньше группы тестирования выявляют ошибки в информационной системе, тем ниже стоимость сопровождения системы. Время на тестирование системы и на исправление обнаруженных ошибок следует предусматривать не только на этапе разработки, но и на этапе проектирования.

Для автоматизации тестирования следует использовать системы отслеживания ошибок (bug tracking). Это позволяет иметь единое хранилище ошибок, отслеживать их повторное появление, контролировать скорость и эффективность исправления ошибок, видеть наиболее нестабильные компоненты системы, а также поддерживать связь между группой

разработчиков и группой тестирования (уведомления об изменениях по e-mail и т.п.). Чем больше проект, тем сильнее потребность в bug tracking.

### Тема 3.5. Содержание технического задания на ИС

Результаты обследования представляют объективную основу для формирования технического задания на информационную систему.

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

При разработке технического задания необходимо решить следующие задачи:

- установить общую цель создания ИС, определить состав подсистем и функциональных задач;
- разработать и обосновать требования, предъявляемые к подсистемам;
- разработать и обосновать требования, предъявляемые к информационной базе, математическому и программному обеспечению, комплексу технических средств (включая средства связи и передачи данных);
- установить общие требования к проектируемой системе;
- определить перечень задач создания системы и исполнителей;
- определить этапы создания системы и сроки их выполнения;
- провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности ее внедрения.

Типовые требования к составу и содержанию технического задания приведены в таблице 3.3.

Таблица 3.3 – Состав и содержание технического задания (ГОСТ 34.602- 89)

№ п/п	Раздел	Содержание
1	Общие сведения	а) полное наименование системы и ее условное обозначение б) шифр темы или шифр (номер) договора; в) наименование предприятий разработчика и заказчика системы, их реквизиты г) перечень документов, на основании которых создается ИС д) плановые сроки начала и окончания работ е) сведения об источниках и порядке финансирования работ ж) порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств
2	Назначение и цели создания (развития) системы	з) вид автоматизируемой деятельности и) перечень объектов, на которых предполагается использование системы к) наименования и требуемые значения технических, технологических, производственно-экономических и др. показателей объекта, которые должны быть достигнуты при внедрении ИС
3	Характеристика объектов автоматизации	л) краткие сведения об объекте автоматизации м) сведения об условиях эксплуатации и характеристиках окружающей среды
4	Требования к системе	Требования к системе в целом: н) требования к структуре и функционированию системы (перечень подсистем, уровни иерархии, степень централизации, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы) о) требования к персоналу (численность пользователей, квалификация, режим работы, порядок подготовки) п) показатели назначения (степень приспособляемости системы к изменениям процессов управления и значений параметров) р) требования к надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию



		<p>и ремонту, защите и сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации</p> <p>Требования к функциям (по подсистемам) :</p> <p>с) перечень подлежащих автоматизации задач</p> <p>г) временной регламент реализации каждой функции</p> <p>у) требования к качеству реализации каждой функции, к форме представления выходной информации, характеристики точности, достоверности выдачи результатов</p> <p>ф) перечень и критерии отказов</p> <p>Требования к видам обеспечения:</p> <p>х) математическому (состав и область применения мат. моделей и методов, типовых и разрабатываемых алгоритмов)</p> <p>ц) информационному (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые классификаторы, СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным документам)</p> <p>ч) лингвистическому (языки программирования, языки взаимодействия пользователей с системой, системы кодирования, языки ввода- вывода)</p> <p>ш) программному (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ)</p> <p>щ) техническому</p> <p>ы) метрологическому</p> <p>э) организационному (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала)</p> <p>ю) методическому (состав нормативно- технической документации)</p>
5	Состав и содержание работ по созданию системы	<p>а) перечень стадий и этапов работ</p> <p>б) сроки исполнения</p> <p>с) состав организаций — исполнителей работ</p> <p>д) вид и порядок экспертизы технической документации</p> <p>е) программа обеспечения надежности</p> <p>ф) программа метрологического обеспечения</p>
6	Порядок контроля и приемки системы	<p>г) виды, состав, объем и методы испытаний системы</p> <p>h) общие требования к приемке работ по стадиям</p> <p>i) статус приемной комиссии</p>
7	Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие	<p>j) преобразование входной информации к машиночитаемому виду</p> <p>k) изменения в объекте автоматизации</p> <p>l) сроки и порядок комплектования и обучения персонала</p>
8	Требования к документированию	<p>m) перечень подлежащих разработке документов</p> <p>n) перечень документов на машинных носителях</p>
9	Источники разработки	<p>о) документы и информационные материалы, на основании которых разрабатывается ТЗ и система</p>

Эскизный проект предусматривает разработку предварительных проектных решений по системе и ее частям.

Выполнение стадии эскизного проектирования не является строго обязательной. Если основные проектные решения определены ранее или достаточно очевидны для конкретной ИС и объекта автоматизации, то эта стадия может быть исключена из общей последовательности работ.

Содержание эскизного проекта задается в ТЗ на систему. Как правило, на этапе эскизного проектирования определяются:

- функции ИС;
- функции подсистем, их цели и ожидаемый эффект от внедрения;
- состав комплексов задач и отдельных задач;
- концепция информационной базы и ее укрупненная структура;
- функции системы управления базой данных;

- состав вычислительной системы и других технических средств;
- функции и параметры основных программных средств.

По результатам проделанной работы оформляется, согласовывается и утверждается документация в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию системы.

### Тема 3.6. Содержание технического проекта ИС

На основе технического задания (и эскизного проекта) разрабатывается *технический проект ИС*. Технический проект системы - это техническая документация, содержащая общесистемные проектные решения, алгоритмы решения задач, а также оценку экономической эффективности автоматизированной системы управления и перечень мероприятий по подготовке объекта к внедрению.

На этом этапе осуществляется комплекс научно-исследовательских и экспериментальных работ для выбора основных проектных решений и расчет экономической эффективности системы.

Состав и содержание технического проекта приведены в таблице 3.4.

Таблица 3.4 – Содержание технического проекта		
№ п/п	Раздел	Содержание
1	Пояснительная записка	а) основания для разработки системы б) перечень организаций разработчиков в) краткая характеристика объекта с указанием основных технико-экономических показателей его функционирования и связей с другими объектами г) краткие сведения об основных проектных решениях по функциональной и обеспечивающим частям системы
2	Функциональная и организационная структура системы	д) обоснование выделяемых подсистем, их перечень и назначение е) перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания ж) схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы
3	Постановка задач и алгоритмы решения	з) организационно-экономическая сущность задачи (наименование, цель решения, краткое содержание, метод, периодичность и время решения задачи, способы сбора и передачи данных, связь задачи с другими задачами, характер использования результатов решения, в которых они используются) и) экономико-математическая модель задачи (структурная и развернутая форма представления) к) входная оперативная информация ( характеристика показателей, диапазон изменения, формы представления) л) нормативно-справочная информация ( НСИ) (содержание и формы представления) м) информация, хранимая для связи с другими задачами н) информация, накапливаемая для последующих решений данной задачи о) информация по внесению изменений ( система внесения изменений и перечень информации, подвергающейся изменениям) п) алгоритм решения задачи (последовательность этапов расчета, схема, расчетные формулы) р) контрольный пример (набор заполненных данными форм входных документов, условные документы с накапливаемой и хранимой информацией, формы выходных документов, заполненные по результатам решения экономико-технической задачи и в соответствии с разработанным алгоритмом расчета)
4	Организация информационной базы	с) источники поступления информации и способы ее передачи г) совокупность показателей, используемых в системе у) состав документов, сроки и периодичность их поступления ф) основные проектные решения по организации фонда НСИ

		<ul style="list-style-type: none"> <li>х) состав НСИ, включая перечень реквизитов, их определение, диапазон изменения и перечень документов НСИ</li> <li>ц) перечень массивов НСИ, их объем, порядок и частота корректировки информации</li> <li>ч) структура фонда НСИ с описанием связи между его элементами; требования к технологии создания и ведения фонда</li> <li>ш) методы хранения, поиска, внесения изменений и контроля</li> <li>щ) определение объемов и потоков информации НСИ</li> <li>ы) контрольный пример по внесению изменений в НСИ</li> <li>э) предложения по унификации документации</li> </ul>
5	Альбом форм документов	
6	Система математического обеспечения	<ul style="list-style-type: none"> <li>a) обоснование структуры математического обеспечения</li> <li>b) обоснование выбора системы программирования</li> <li>c) перечень стандартных программ</li> </ul>
7	Принцип построения комплекса технических средств	<ul style="list-style-type: none"> <li>d) описание и обоснование схемы технологического процесса обработки данных</li> <li>e) обоснование и выбор структуры комплекса технических средств и его функциональных групп</li> <li>f) обоснование требований к разработке нестандартного оборудования</li> <li>g) комплекс мероприятий по обеспечению надежности функционирования технических средств</li> </ul>
8	Расчет экономической эффективности системы	<ul style="list-style-type: none"> <li>h) сводная смета затрат, связанных с эксплуатацией систем</li> <li>i) расчет годовой экономической эффективности, источниками которой являются оптимизация производственной структуры хозяйства (объединения), снижение себестоимости продукции за счет рационального использования производственных ресурсов и уменьшения потерь, улучшения принимаемых управленческих решений</li> </ul>
9	Мероприятия по подготовке объекта к внедрению системы	<ul style="list-style-type: none"> <li>j) перечень организационных мероприятий по совершенствованию бизнес-процессов</li> <li>k) перечень работ по внедрению системы, которые необходимо выполнить на стадии рабочего проектирования, с указанием сроков и ответственных лиц</li> </ul>
10	Ведомость документов	

В завершение стадии технического проектирования производится разработка документации на поставку серийно выпускаемых изделий для комплектования ИС, а также определяются технические требования и составляются ТЗ на разработку изделий, не изготавливаемых серийно.

На стадии «рабочая документация» осуществляется создание программного продукта и разработка всей сопровождающей документации. Документация должна содержать все необходимые и достаточные сведения для обеспечения выполнения работ по вводу ИС в действие и ее эксплуатации, а также для поддержания уровня эксплуатационных характеристик (качества) системы. Разработанная документация должна быть соответствующим образом оформлена, согласована и утверждена.

Для ИС, которые являются разновидностью автоматизированных систем, устанавливают следующие основные виды испытаний: предварительные, опытная эксплуатация и приемочные. При необходимости допускается дополнительно проведение других видов испытаний системы и ее частей.

В зависимости от взаимосвязей частей ИС и объекта автоматизации испытания могут быть автономные или комплексные. Автономные испытания охватывают части системы. Их проводят по мере готовности частей системы к сдаче в опытную эксплуатацию. Комплексные испытания проводят для групп взаимосвязанных частей или для системы в целом.

Для планирования проведения всех видов испытаний разрабатывается документ «Программа и методика испытаний». Разработчик документа устанавливается в договоре или ТЗ. В качестве приложения в документ могут включаться тесты или контрольные примеры.

Предварительные испытания проводят для определения работоспособности системы и решения вопроса о возможности ее приемки в опытную эксплуатацию. Предварительные

испытания следует выполнять после проведения разработчиком отладки и тестирования поставляемых программных и технических средств системы и представления им соответствующих документов об их готовности к испытаниям, а также после ознакомления персонала ИС с эксплуатационной документацией.

Опытную эксплуатацию системы проводят с целью определения фактических значений количественных и качественных характеристик системы и готовности персонала к работе в условиях ее функционирования, а также определения фактической эффективности и корректировки, при необходимости, документации.

Приемочные испытания проводят для определения соответствия системы техническому заданию, оценки качества опытной эксплуатации и решения вопроса о возможности приемки системы в постоянную эксплуатацию.

## **Раздел 4. ТЕХНОЛОГИЯ И МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ**

### **Тема 4.1. Специфика информационных программных систем (систем обработки данных)**

Известно, что ИС в зависимости от конкретной области применения довольно сильно различаются по своим функциям, архитектуре и реализации. Однако, можно выделить по крайней мере два свойства, которые являются общими для всех ИС.

Во-первых, любая ИС предназначена для сбора, хранения и обработки информации. Поэтому в основе любой ИС лежит среда хранения и доступа к данным. Среда должна обеспечивать уровень надежности хранения и эффективность доступа в соответствующей области применения ИС. Однако в вычислительных программных системах наличие такой среды не является обязательным. Основным требованием к программе, выполняющей численные расчеты, является ее быстрдействие. Здесь нужно, чтобы программа произвела достаточно точные результаты за установленное время. При этом внешняя память используется для периодического (нечастого) сохранения промежуточных результатов вычислений, чтобы в случае сбоя компьютера можно было продолжить работу программы от сохраненной контрольной точки.

Во-вторых, ИС ориентируется на конечного пользователя, например, банковского клерка. Такие пользователи могут быть далеки от компьютеров. Для них терминал, ПК или рабочая станция представляют собой лишь орудие их собственной профессиональной деятельности. Поэтому ИС должна обладать простым, удобным, легко осваиваемым интерфейсом, который должен предоставить конечному пользователю все необходимые для его работы функции, но в то же время не дать ему возможность выполнить какие-либо лишние действия. Сегодня очень популярны графические интерфейсы с меню, кнопками, подсказками и т.п., а потому многие современные средства разработки информационных приложений ориентированы на разработки таких интерфейсов.

Следует заметить, что если для любой современной ИС наличие развитых интерфейсных средств является обязательным, то для вычислительных программных систем их наличие не обязательно. Обусловлено это тем, что серьезные вычислительные программы, как правило, почти уникальны, расчеты выполняются либо разработчиками программ, либо людьми из того же окружения, для которых гораздо важнее быстрдействие вычислений, чем удобство запуска программы, а наличие развитого интерфейса предполагает существенный расход компьютерных ресурсов. Поскольку эффективность использования систем зависит от качества интерфейса, а пользователи весьма часто судят о качестве ИС в целом, исходя также из качества ее интерфейса, то при построении полного интерфейса необходимо учитывать требования эстетичности и удобства, а также принимать во внимание особенности конкретной области применения ИС.

## **Тема 4.2. Трудности, встречающиеся при проектировании ИС**

При создании ИС весьма часто принимались допущения, которые фактически не оправдывались. К ним относятся:

1. Чем больше, тем лучше, т.е. управленческому персоналу часто предоставлялось информации больше, чем ему требовалось. Однако данные следовало бы проверять на релевантность и концентрировать их в соответствии с действительными потребностями. Например, следовало бы давать контрольные итоги, а не детальные данные.

2. Управленческий персонал нуждается во всей той информации, которую он запрашивает. Администраторы часто впадают в искушение запросить всю доступную информацию, однако системный аналитик должен гарантировать, что потребности в информации будут удовлетворяться только лишь после тщательного исследования процесса принятия решения. При этом информация не должна поставляться только по той причине, что она доступна. Поэтому системному аналитику необходимо знать о действительных потребностях администраторов.

3. Процесс принятия решений способствует развитию представляемой информации. В действительности администраторы могут фактически игнорировать получаемую информацию, хотя она и является релевантной, полагаясь только на свою интуицию, опыт, знания. Они должны знать потенциальные возможности системы, и им должно быть гарантировано эффективное использование этих возможностей.

4. Повышение коммуникабельности ведет к росту эффективности работы. Фактически администраторы, работающие с ИС, могут иметь доступ к информации, относящейся к другим подразделениям или подсистемам организации. Но это, в конце концов, может привести к нежелательным конфликтным ситуациям между подразделениями. Подобные последствия должны быть предусмотрены до того, как будет разрешено свободное пользование информацией.

5. Администратор должен только уметь использовать систему. Однако для того чтобы иметь возможность оценить систему и осуществлять управление ею, администратору необходимо обладать некоторыми знаниями о механизме работы ИС.

Таким образом, проектирование ИС – весьма тонкий и сложный процесс, а потому не может быть поручен только техническим работникам.

## **Тема 4.3. Критерии создания ИС**

Основным критериями, которым должна удовлетворять ИС при ее построении являются:

1. Релевантность, т.е. получаемая информация должна быть релевантна запросам любого лица, принимающего решения. Эта информация различается в зависимости от потребителей, которыми могут быть, например, директор фирмы или техник. Существуют ключевые различия между данными и информацией. Информация - это данные релевантные потребителю.

2. Управление по отклонениям. Административному работнику не требуется большого количества детальной информации, но он должен быть хорошо осведомлен о критических факторах, влияющих на успех фирмы или организации.

3. Точность. Данные, на основе которых формируется информация, должны быть правильны с тем, чтобы решения принимались на основе точной информации, требуемой для рассматриваемой задачи. Точность также предполагает, что информация отражает текущее состояние и, следовательно, она не должна основываться на устаревших данных.

4. Своевременность, т.е. информация должна предоставляться именно тогда, когда она необходима потребителю.

5. Приспособляемость. Система должна быть способна удовлетворять различные запросы пользователя. Эти запросы могут принимать форму контроля итоговых сумм, например, объема сбыта по определенному подразделению или быть более детализированными, например, запрос об объеме сбыта, реализованном отдельным агентом.

В системе должны быть также предусмотрены возможности перепроектирования в соответствии с изменяющимися потребностями пользователей.

6. Установление набора целей. Особая важность этого требования состоит в том, что «если с самого начала четко определены цели системы, то можно судить о том, достигаются ли они в действительности; это, следовательно, дает некоторую меру оценки работоспособности системы и степень пригодности ее для организации» (Шодербек П. П.). Следует отметить, что сказанное применительно к ИС в целом, применимо и к ее любой отдельно взятой подсистеме.

## **РАЗДЕЛ 5. ТРЕБОВАНИЯ К ОРГАНИЗАЦИИ ДИАЛОГА И ПРЕДСТАВЛЕНИЮ ДАННЫХ**

Лекция проводится в интерактивной форме с разбором конкретных ситуаций (3 час.)

### **Тема 5.1. Стратегии построения ИС**

Поскольку различия между ИС чрезвычайно велики, а проблемы проектирования и реализации определяются проблемами практического характера вне зависимости от того, являются ли они политическими, финансовыми или психологическими, то лучше всего определить стратегию проектирования для каждой конкретной организации, что позволит учесть указанные проблемы до начала разработки конкретной информационной модели. Это связано еще и с тем, что стратегия, или система наших взглядов относительно организации, оказывает существенное влияние на разрабатываемую модель. Поэтому, если подход определен, то характеристики реальной организации уже можно отразить непосредственно на информационной модели, а следовательно, разработанный проект системы будет менее абстрактным и более релевантным потребностям организации. Если же стратегия окажется ошибочной, а практические проблемы не будут учтены в достаточной степени, то получится чисто теоретический проект.

Известен целый ряд общих стратегий, которые не являются взаимоисключающими и на практике любая из них может применяться вместе с другими.

#### **Подход от организационной структуры**

Этот подход применим в системе, которая базируется на существующих границах организации и ее структуре. К функциональным областям деятельности организации обычно относят финансы, производство продукции, персонал, участие в рынке и заказы. Этот подход не является радикально новым и реализация ИС не требует перестройки существующей системы управления (СУ). Основной недостаток – может быть упущена возможность совершенствования организации, а также устаревшие методы, теряющие жизнеспособность через определенные промежутки времени, вероятнее всего окажутся перенесенными в новую систему. Поэтому подход, основанный на организационной структуре, наиболее применим в случае, когда сфера деятельности фирмы достаточно консервативна, и что этот подход дает время для того, чтобы воспринять новую технологию.

#### **Подход с отложенной интеграцией**

Этот подход разумен, когда не существуют вполне ясного представления о том, как будет развиваться ИС в организации, поскольку порождает меньше проблем, чем чрезмерно амбициозный план. Трудности применения данного подхода заключаются в том, что независимые подсистемы могут развиваться в большие системы и последующая интеграция, если она будет возможна, может оказаться сложной и дорогой. Поэтому задача окончательного объединения системы может столкнуться с более серьезными проблемами, чем просто задержка процесса интеграции.

### **Подход, базирующийся на сборе данных**

В рамках такого подхода на первом этапе проектирования ИС особое значение придается сбору всех данных, которые могут быть использованы в системе. Затем данные тщательно классифицируются, что помогает понять, как будут использоваться данные, что окажет определенное воздействие на способы последующего использования данных.

С коммерческой точки зрения идея использования сбора данных в случае, когда результат может оказаться полезным на некотором ограниченном отрезке времени в будущем, имеет мало смысла. Однако в большинстве коммерческих систем данные собираются как *побочные* продукты функционирования этих систем. Как правило, это сообщения, состоящие из текущих выверенных данных, и на их основе может быть разработана хорошая ИС. Основная трудность кроется не только в области детальной классификации, но и в выборе источника данных, достоверности данных и возможности удаления данных после потери их значимости.

### **Подход, основанный на использовании баз данных**

Этот подход также предполагает осуществление сбора, хранения и поддержки большого количества данных. Данные должны быть детализированы до такой степени, чтобы содержать все необходимое для операционного и административного управления в деловой сфере. Соответствующая база данных будет использоваться всеми подсистемами и абонентами, которые по мере необходимости будут осуществлять к ней доступ. БД поддерживается достаточно развитым ПО-СУБД. Но его (ПО) использование связано с организацией достаточно сложной службы, обеспечивающей безопасность, секретность и точность данных. Подход, основанный на современной технологии ведения БД, заслуживает еще большего внимания по двум причинам: во-первых, даже если мы и не получим завершенной ИС, то за счет независимости данных упрощается развитие системы, и во-вторых, при таком подходе обеспечивается возможность с помощью языка запросов предоставить пользователям непосредственный доступ к информации.

### **Подход «сверху вниз»**

Этот подход влечет за собой определение информационных потребностей для всей последовательности уровней управления, начиная от оценки потребностей управления и общих целей всего бизнеса. Если информация, требуемая на высшем уровне, остается относительно устойчивой по степени детальности, содержанию и частоте использования, то системы могут удовлетворить этим требованиям.

Рассматриваемый подход может быть обоснованным и там, где существует разница в типе информации, требуемой на различных уровнях. Однако, при этом теряется два основных преимущества БД, содержащих пооперационные данные, а именно, преимущества, вытекающие из того, что ценность информации определяется операцией и что достоверность данных может быть установлена совместно с порождающей их операцией.

Например, если управленческий аппарат высшего уровня получит сведения об общем числе служащих из платежной ведомости, то данные, вероятно, будут точны, а способ их получения – дешев. Если же для этой цели будет использоваться отдельная ИС, в которой общее число служащих подсчитывается на основе общего числа карточек, то способ получения данных может оказаться очень дорогим, а результат – более трудным в смысле установления достоверности.

### **Общесистемный подход**

Этот подход основывается на предположении, что еще до реализации системы возможно определить и распознать взаимосвязи между всей базовой информацией. При этом процессы сбора, хранения и обработки данных проектируются и реализуются в рамках всей системы в целом. И хотя этот подход является идеальным, его применение в полном объеме может оказаться весьма трудным из-за объема практических, политических и социальных проблем. Однако в организациях, которые еще не имеют разработанных систем, действующих и считающихся удовлетворительными, рассматриваемый подход может быть успешно применен. Лучшим примером применения этого подхода является NASA, которая среди

других целей поставила цель посещения Луны человеком. В ходе его (подхода) реализации были разработаны новые превосходные стандарты надежности и управления проектом. В данном случае не было необходимости создавать специализированную организацию по решению проблемы прилунения путем преобразования существующей. Здесь весь служебный персонал был набран заново и все верили, что цели, на которые направлены их усилия, достижимы. Для применения этого подхода имеется ряд других случаев, например, когда компания строит новую фабрику или филиал, а также, когда заново строится больница и т.п.

Следует помнить, что этот подход является идеализированным и не может в полном объеме применяться в существующей организации.

Из рассмотренных подходов следует, что стратегия выбора подхода должна специально формироваться для конкретной системы. При этом необходимо принимать в расчет такие факторы, как размер организации, природа ее деловых операций и опыт. Существенно, что выбор стратегии должен быть произведен после тщательной оценки степени риска и преимуществ возможных подходов.

## Тема 5.2. Методологии и технологии проектирования ИС

Основу проекта любой ИС составляют методологии, технологии и инструментальные средства проектирования. Методология реализуется через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства обеспечивают выполнение процессов жизненного цикла. Технология проектирования определяется как совокупность трех составляющих:

1. Пошаговой процедуры, определяющей последовательность технологических операций проектирования. Последнюю обычно представляют следующим образом (рис. 5.1),



Рис. 5.1

где ВД - входные данные: исходные данные в стандартном представлении, т.е. документы, рабочие материалы, результаты предыдущей операции;

ВР - выходные данные: результаты в стандартном представлении;

М - механизм: исполнители, программные и технические средства;

У - управление: методические материалы, инструкции, нормативы и стандарты, критерии оценки результатов.

2. Критериев и правил, используемых для оценки результата выполнения технологических операций.

3. Графических и текстовых средств (нотаций), используемых для описания проектируемой системы.

Технологические инструкции, составляющие основное содержание технологий, должны состоять:

- из описания последовательности технологических операций;
- условий, в зависимости от которых выполняется та или иная операция;
- описания самих операций.

Реальное применение любой технологии проектирования, разработки и сопровождения ИС в конкретной организации и конкретном проекте не возможно без выработки ряда стандартов (правил, соглашений), которые должны соблюдаться всеми участниками проекта. К таким стандартам относят:

1. Стандарт проектирования, который должен устанавливать:



1.1. Набор необходимых моделей (диаграмм) на каждой стадии проектирования и степень их детализации.

1.2. Правила фиксации проектных решений на диаграммах, в том числе:

а) правила именования объектов, включая соглашения по терминологии;

б) набор атрибутов для всех объектов;

в) правила оформления диаграмм, включая требования к форме и размерам объектов.

1.3. Требования к конфигурации рабочих мест разработчиков, включая настройки операционной системы, настройки CASE-средств и т.д.

1.4. Механизм обеспечения совместной работы над проектом, в том числе:

г) правила интеграции подсистем проекта;

д) правила поддержания проекта в одинаковом для всех разработчиков состоянии (регламент обмена проектной информацией, механизм фиксации общих проектов);

е) правила проверки проектных решений на непротиворечивость и др.

2. Стандарт оформления проектной документации, устанавливающий:

2.1. Комплектность, состав и структуру документации на каждой стадии проектирования.

2.2. Требования к ее оформлению, включая требования к содержанию разделов, подразделов, пунктов, таблиц и т.д.

2.3. Правила подготовки, рассмотрения, согласования и утверждения документации с указанием предельных сроков для каждой стадии.

2.4. Требования к настройке издательской системы, используемой в качестве встроенного средства подготовки документации.

2.5. Требования к настройке CASE-средств для обеспечения подготовки документации в соответствии с установленными требованиями.

3. Стандарт пользовательского интерфейса, который должен устанавливать:

3.1. Правила оформления экрана (шрифты и цветовая палитра), состав и расположение окон и элементов управления.

3.2. Правила использования клавиатуры и мыши.

3.3. Правила оформления текстов помощи.

3.4. Перечень стандартных сообщений.

3.5. Правила обработки реакции пользователя.

### **Тема 5.3. Структурное представление ИС при их создании**

Сущность такого представления ИС заключается в ее декомпозиции (разбиении) на автоматизированные функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяющиеся на задачи и т.д. Процесс разбиения продолжается вплоть до конкретных процедур, при этом автоматизируемая система сохраняет *целостное* представление, в котором все составляющие компоненты взаимосвязаны.

Все наиболее распространенные методологии структурного представления базируются на ряде общих принципов. В качестве двух базовых принципов используют:

- принцип «разделяй и властвуй» – принцип решения сложных проблем путем их разделения на множество меньших *независимых* задач; легких для понимания и решения;

- принцип иерархического упорядочивания – принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

- Однако выделение двух основных принципов не означает, что остальные принципы являются второстепенными, т.к. игнорирование любого из них может привести к непредсказуемым последствиям, в том числе и к провалу всего проекта. Основными из этих принципов являются следующие:

- принцип абстрагирования, заключающийся в выделении существенных аспектов системы и отвлечения от несущественных;

- принцип формализации, сущность которого состоит в необходимости строго методического подхода к решению проблемы;
- принцип непротиворечивости, заключающийся в обоснованности и согласованности элементов;
- принцип структурирования данных, заключающийся в том, что данные должны быть структурированы и иерархически организованы;
- принцип логической независимости, заключающийся в концентрации внимания на логическом описании системы и обеспечении независимости от ее физической реализации;
- принцип независимости данных, заключающийся в том, что модели данных могут быть проанализированы и спроектированы независимо от процессов их логической обработки, а также от их физической структуры и распределения.

Соблюдение всех указанных принципов необходимо на начальных этапах жизненного цикла системы (анализе и проектировании) независимо от типа разрабатываемой ИС и используемой при этом методологии. Обусловлено это относительно невысокой сложностью и трудозатратами на последующих этапах.

### **Структурный анализ**

Структурный анализ – это метод исследования систем с помощью ее графического модельного представления, который начинается с общего обзора и затем детализируется, приобретает иерархическую структуру со все большим числом уровней.

Этот анализ является первым этапом создания ИС, на котором требования заказчика формализуются, уточняются и документируются. Фактически на этом этапе дается ответ на вопрос: «Что должна делать система?». Именно в анализе лежит ключ к успеху всего проекта. Целью анализа является преобразование общих, расплывчатых знаний об исходной ПО в *точные* определения и спецификации, а также генерация функционального описания системы. На этом этапе специфицируются:

- внешние условия работы системы;
- функциональная структура системы;
- распределение функций между человеком и системой, интерфейсы;
- требования к техническим, информационным и программным компонентам системы;
- условия эксплуатации.

Разработка перечисленных выше спецификаций при создании ИС, предназначенных для автоматизации управленческих процессов, в общем случае проходит четыре стадии.

Структурный анализ начинается с исследования того, как организована система управления предприятием и с обследования функциональной и информационной структуры системы. Иными словами, на первой стадии структурного анализа вначале выясняется:

- для чего создана организация;
- к чему она стремится;
- каковы направления ее деятельности;
- какова структура организации;
- какие функции выполняют ее подразделения;
- как они взаимодействуют между собой.

Поэтому целями исследования являются:

1. Формулировка и описание функций каждого подразделения организации, а также решаемые ею задачи;
2. Описание технологии работы «как есть» каждого из подразделений организации и понимания того, что необходимо автоматизировать и в какой последовательности;
3. Описание технологии работы «как надо» каждого из подразделений и связанных с ними информационных потоков, исходя из миссии и направления деятельности организации;
4. Отображение технологии «как надо» на структуру организации, определение ее функционального состава и количества рабочих мест в каждом структурном подразделении организации, а также описание функций, которые выполняются (автоматизируются) на каждом рабочем месте;

5. Описание основных путей и алгоритмы прохождения входящих, внутренних и исходящих документов, а также технологии их обработки.

Следует отметить, что первые два пункта целей исследования реализуются на первой стадии структурного анализа, а последующие пункты – на второй и третьей стадиях. По результатам обследования аналитик на первой стадии анализа строит обобщенную логическую модель *исходной* ПО, отображающую ее функциональную структуру, особенности основной деятельности и информационное пространство, в котором эта деятельность осуществляется.

Вторая стадия работы, к которой привлекаются заинтересованные представители заказчика, а при необходимости и независимые эксперты, состоит:

- в анализе модели «как есть»;
- в выявлении ее недостатков и узких мест;
- определение путей совершенствования системы управления на основе выделенных критериев качества.

Третья стадия анализа, содержащая элементы проектирования, – это создание усовершенствованной обобщенной логической модели, отображающей реорганизованную предметную область или ее часть, которая подлежит автоматизации. Эту модель называют моделью «как надо».

Заканчивается процесс (четвертая стадия) разработкой карты автоматизации, представляющей собой модель реорганизованной предметной области, на которой обозначены «границы автоматизации». Следует отметить, что на практике рассмотренную общую схему структурного анализа и проектирования, включающую стадию планирования реорганизованной деятельности организации, приходится встречать крайне редко. Такую работу для заказчиков могут выполнять лишь крупные специализированные фирмы. В большинстве случаев модель «как есть» улучшается системным аналитиком за счет устранения очевидных несоответствий, а полученный таким образом вариант модели считается моделью «как надо».

Основным документом, отражающим результаты работ первого этапа создания ИС, является техническое задание на проект, содержащее, кроме выше перечисленных определений и спецификаций, также сведения об очередности создания системы, сведения о выделенных ресурсах, директивных сроках проведения отдельных этапов работ, организационных процедурах и мероприятиях по приемке этапов, защите проектной информации. Из рассмотренного следует, что:

1. На стадии исследований формулируется возможность реализации (осуществимости) системы, для чего решаются главные вопросы:

- а) действительно ли имеется проблема или есть безпроблемные альтернативы;
- б) целесообразно ли создание новой ИС;
- в) действительно ли ее можно реализовать (создать), а также оформляется отчет об осуществимости системы.

2. Структурный анализ включает:

- г) структурные исследования;
- д) собственно структурный анализ.

3. В результате структурного анализа:

е) формируются функциональные требования к будущей системе с составлением документа, отражающего информационные потребности конечных пользователей, влияние окружающей среды, роль существующей ИС;

ж) определяются и документируются требования не только к системе в целом, но и к ее ресурсам: оборудованию, телекоммуникациям, программному обеспечению, базе данных, а также информационным процессам (ввод данных, хранение, обработка, управление данными).

В настоящее время для целей структурного анализа и моделирования систем вообще используется три группы средств, отображающих:

- функции, которые система должна выполнять;
- процессы, обеспечивающие выполнение указанных функций;

- данные, используемые при выполнении функций и отношения между этими данными.  
Среди всего многообразия средств решения указанных задач в методологиях структурного анализа наиболее часто и эффективно применяемыми являются:

- FDD (Function Decomposition Diagrams) – диаграммы функциональной декомпозиции;
- DFD (Data Flow Diagrams) – диаграммы потоков;
- ERD (Entity-Relation Diagrams) – диаграммы «Сущность-Связь».

Все эти инструментальные средства содержат как графические, так и текстовые средства моделирования. Графические средства служат для удобства демонстрации основных компонентов моделей и их связей, а текстовые – для обеспечения точного определения компонентов и связей. При помощи этих средств строятся как логические модели исходной и реорганизованной систем управления, так и логическая модель автоматизированной системы управления (АСУ), т.е. подробное описание того, что и как должна делать система, при условии освобождения от рассмотрения путей реализации.

### Структурное проектирование

Главный документ, который должен иметь специалист (специалисты) перед началом проектирования, является та часть технического задания, в которой отражены «Требования к системе».

Разработка системных спецификаций составляет *цель* этого этапа проектирования, а сами спецификации являются его *продуктом*, как основного элемента для этапа внедрения ИС. Данный этап проектирования завершается формулированием *детального* (рабочего) *проекта* ИС. Проект представляет собой серию спецификаций и инструкций пользователя. На рис. 5.2 представлены типы спецификаций и их содержание.



Рис. 5.2. Типы и содержание спецификаций

В процессе проектирования различают (рис. 5.3):

- логическое проектирование;
- физическое проектирование;
- спецификации.

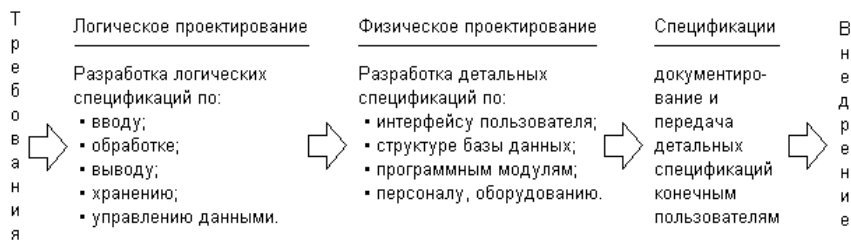


Рис. 5.3. Схема последовательности шагов и их содержание

Логические и физические аспекты более всего относятся к проектированию баз данных. Структура БД находится в центре внимания всего проектирования *прикладных ИС*. Главное здесь – выявить и создать описание информационных объектов, их структуры, а также связей между объектами и их элементами. Практическое проектирование прикладной ИС (приложения) обычно представляется как разработка трех главных видов ее элементов:

- информационного обеспечения (концептуальных и внешних моделей БД, словаря элементов БД);
- пользовательского интерфейса (командных линеек, панелей меню, диалоговых окон, сообщений, экранов входных и выходных форм);
- программного обеспечения (программ и процедур).

#### *Разработка информационного обеспечения*

Проектирование информационного обеспечения выявляет и документирует три ее главные стадии:

- сущности, т.е. объекты реально существующие в ИС (описание сущностей по специальным правилам является важнейшим исходным условием многих CASE-технологий);
- связи объектов, которые описывают наличие отношений между разными элементами одного объекта или между объектами, а также количественные особенности отношений;
- правила интеграции сущностей и описание их ролей.

Правила интеграции логических элементов в отдельные записи фиксируются в специальной таблице соответствия записей и их элементов (например, в реляционной таблице). Словарь содержит текстовые характеристики (толкование) всех элементов для обеспечения *однозначного* понимания их смысла.

#### *Разработка пользовательского интерфейса*

Эта процедура фактически реализует организационную модель приложения, связывая ее с информационным обеспечением (БД) и программным обеспечением (программными модулями).

Пользовательский интерфейс моделирует также отношения между пользователем и компьютерной системой посредством готовых экранных форм ввода данных и вывода отчетов. При проектировании интерфейса разрабатывается иерархическая структура диалогов, создаются перечни диалогов, шагов, использованных объектов, физический вид каждого шага диалога.

#### *Разработка прикладного программного обеспечения*

Такая процедура в настоящее время осуществляется на логическом уровне с использованием возможностей объектно-ориентированных программных языков четвертого поколения. Проектирование структуры БД и общение с современными СУБД также осуществляется на логическом уровне, т.е. с использованием «прозрачных» (легких для понимания их смысла) идентификаторов и имен данных.

Физическое размещение БД на устройствах памяти формируется автоматически посредством СУБД. Кроме программ разрабатываются также и процедуры, например, архивации. Таким образом, разработка программ и процедур прочно увязывает пользовательский интерфейс, информационное обеспечение и характеристики оборудования.

## Роль инструментов и стандартов в проектировании

Практика показала, что создать большую, устойчивую, динамичную и эффективную ИС невозможно, если не использовать стандарты отображения структуры системы, алгоритмов и т.п.

Многолетний опыт проектирования ИС породил довольно удобные и главным образом графические средства (инструменты) для наглядного изображения элементов проекта. Это табличные формы, диаграммы, блок-схемы и другие графические способы представления ресурсов и процессов.

Различают ручные и машинные инструменты. Например, пластиковые линейки с набором стандартных графических символов (широко используются до сих пор при ручном вычерчивании элементов). Однако это не единственный способ стандартизации. Стандартизируются также технология, принципы проектирования и др.

Альтернативой ручному проектированию сегодня созданы стандартные средства, ориентированные на компьютеры (стандарты отдельных CASE-технологий). Например, в США, используются:

- стандарты «Архитектура интегрированных приложений» от Digital Equipment Corporation;
- System's Application Architecture (SAA) – архитектура прикладных систем от IBM;
- стандарты Application Integration Architecture (AIA) – New Ware (от Hewlett Packard) и др.

## Внедрение ИС

На этом этапе разработанный проект внедряется в реальную среду предприятия или организации, т.е. приспособляется к текущей работе пользователей. Составляющие этого процесса приведены на рис. 5.4.

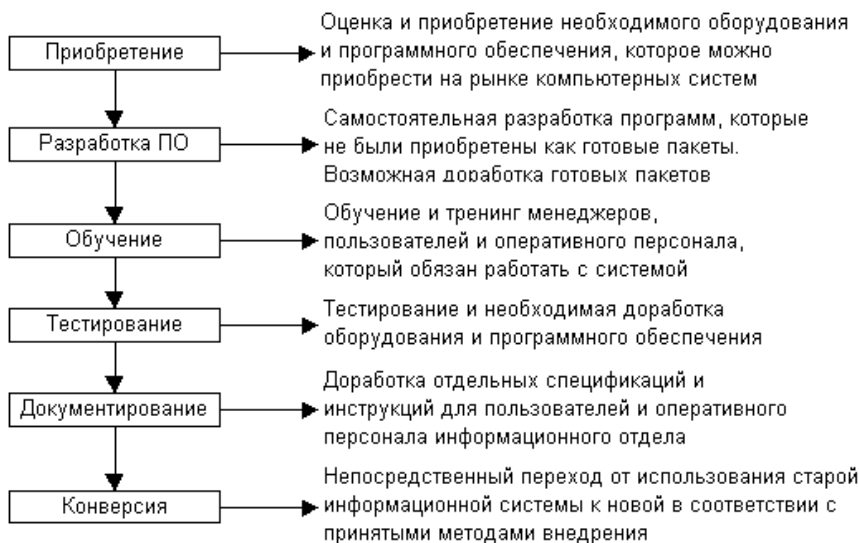


Рис. 5.4

Из приведенного выше рисунка следует, что основными мероприятиями, проводимыми на этапе внедрения проекта ИС, являются:

- приобретение (или разработка) и установление оборудования и программного обеспечения;
- тестирование системы и ее документирование;
- обучение персонала;
- переход на новую систему.

В зависимости от того, сохраняется ли старая технология обработки данных на период внедрения новой системы, различают варианты внедрения:

- с дублированием работ старым способом;
- без дублирования.

Внедрение первого варианта более надежно в случаях отказа новой системы или каких-либо ее нарушений. Однако он требует дополнительных затрат на двойную технологию обработки данных. Если в процессе внедрения системы основным приоритетом является сохранение и не ухудшение качества обслуживания клиентов, то следует выбрать метод внедрения с дублированием несмотря на затраты, если приоритет – снижение издержек, то лучше выбрать метод без дублирования. В этом случае следует заблаговременно провести тестирование оборудования и программного обеспечения, а также предварительное обучение персонала с контролем обучения.

При наличии нескольких структурных подразделений, в которых следует внедрять проект системы, переход на новую систему может осуществляться последовательным, параллельным или комбинированным методом. При последовательном методе проект внедряется поочередно в каждом подразделении, суммарное время внедрения проекта при этом максимальное. Однако для непервоочередных подразделений имеется возможность вносить коррективы, необходимость внедрения которых обнаруживается при внедрении в первоочередных подразделениях. При параллельном методе проект внедряется одновременно во всех подразделениях, поэтому время ввода здесь минимальное, а внесение коррективов усложняется и умножается с числом подразделений. Компромиссным и оптимальным вариантом является комбинированный метод, когда сначала проект внедряется последовательно в одном-двух подразделениях, затем по результатам внедрения в проект вносят необходимые коррективы и, наконец, откорректированный проект одновременно (параллельно) внедряется во всех остальных подразделениях организации.

## **Раздел 6. ПРИНЦИПЫ ПОСТРОЕНИЯ И ОРГАНИЗАЦИИ БАЗ ДАННЫХ**

Часто довольно трудно понять, какой механизм управления данными во внешней памяти необходим и достаточен для решения конкретной прикладной задачи. Если раньше приходилось сталкиваться с ситуацией, когда возможности файлов переоценивались, то теперь сплошь да рядом применяют базы данных в случаях, для которых вполне достаточно файлов. Поэтому кажется полезным еще раз перечислить возможности и альтернативные решения файловых систем, показать наиболее типичные области их применения и отметить те ситуации, когда эти возможности оказываются недостаточными для организации прикладных систем.

### **Тема 6.1. Численные и информационные прикладные системы**

Во всей истории вычислительной техники можно проследить две основных области ее использования. Первая область - применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную. Развитие этой области способствовало интенсификации методов численного решения сложных математических задач, развитию класса языков программирования, ориентированных на удобную запись численных алгоритмов, становлению обратной связи с разработчиками новых архитектур ЭВМ.

Вторая область, которая непосредственно относится к теме наших лекций, - это использование средств вычислительной техники в автоматических или автоматизированных информационных системах. В самом широком смысле информационная система представляет собой программно-аппаратный комплекс, функции которого состоят в надежном хранении информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса. Обычно такие системы имеют дело с большими объемами информации, и эта информация имеет достаточно сложную структуру. Классическими примерами информационных систем являются банковские системы, системы резервирования авиационных или железнодорожных билетов, мест в гостиницах и т. д.

Вторая область использования вычислительной техники возникла несколько позже первой. Это связано с тем, что на заре вычислительной техники возможности компьютеров по хранению информации были очень ограниченными. Говорить о надежном и долговременном хранении информации можно только при наличии запоминающих устройств, сохраняющих информацию после выключения электрического питания. Оперативная (основная) память компьютеров этим свойством обычно не обладает. В первых компьютерах использовались два вида устройств внешней памяти - магнитные ленты и барабаны. Емкость магнитных лент была достаточно велика, но по своей физической природе они обеспечивали последовательный доступ к данным. Магнитные же барабаны (они больше всего похожи на современные магнитные диски с фиксированными головками) давали возможность произвольного доступа к данным, но были ограниченного размера.

Эти ограничения не являлись слишком существенными для чисто численных расчетов. Даже если программа должна обработать (или произвести) большой объем информации, при программировании можно продумать расположение этой информации во внешней памяти (например, на последовательной магнитной ленте), обеспечивающее эффективное выполнение этой программы.

Но для информационных систем, в которых потребность в текущих данных определяется конечным пользователем, наличие только магнитных лент и барабанов неудовлетворительно. Представьте себе покупателя билета, который, стоя у кассы, должен дожидаться полной перемотки магнитной ленты. Одним из естественных требований к таким системам является удовлетворительная средняя скорость выполнения операций.

Как кажется, именно требования нечисленных приложений вызвали появление съемных магнитных дисков с подвижными головками, что явилось революцией в истории вычислительной техники. Эти устройства внешней памяти обладали существенно большей емкостью, чем магнитные барабаны, обеспечивали удовлетворительную скорость доступа к данным в режиме произвольной выборки, а возможность смены дискового пакета на устройстве позволяла иметь практически неограниченный архив данных.

С появлением магнитных дисков началась история систем управления данными во внешней памяти. До этого каждая прикладная программа, которой требовалось хранить данные во внешней памяти, сама определяла расположение каждой порции данных на магнитной ленте или барабане и выполняла обмены между оперативной памятью и устройствами внешней памяти с помощью программно-аппаратных средств низкого уровня (машинных команд или вызовов соответствующих программ операционной системы). Такой режим работы не позволяет или очень затрудняет поддержание на одном внешнем носителе нескольких архивов долговременно хранимой информации. Кроме того, каждой прикладной программе приходилось решать проблемы именования частей данных и структуризации данных во внешней памяти.

## **Тема 6.2. Файловые системы**

Историческим шагом явился переход к использованию централизованных систем управления файлами. С точки зрения прикладной программы, файл - это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные. Правила именования файлов, способ доступа к данным, хранящимся в файле, и структура этих данных зависят от конкретной системы управления файлами и, возможно, от типа файла. Система управления файлами берет на себя распределение внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным.

Первая получившая широкое распространение файловая система была разработана фирмой ИВМ для серии компьютеров 360. К настоящему времени она очень устарела, и мы не будем рассматривать детали ее организации. Заметим лишь, что в этой системе поддерживались как чисто последовательные, так и индексно-последовательные файлы, а реализация во многом опиралась на возможности только появившихся к этому времени контроллеров управления дисковыми устройствами. Если учесть к тому же, что понятие



файла в OS/360 являлось базовой абстракцией, которому соответствовал любой внешний объект, включая внешние устройства, то работать с файлами на уровне пользователя было очень неудобно. Требовался целый ряд громоздких и перегруженных деталями конструкций. Все это хорошо знакомо программистам среднего и старшего поколения, которые прошли через использование отечественных аналогов компьютеров IBM.

Поэтому дальше мы будем говорить о более современных организациях файловых систем и начнем со структур файлов.

### **Структуры файлов**

Прежде всего, практически во всех современных компьютерах основными устройствами внешней памяти являются магнитные диски с подвижными головками, и именно они служат для хранения файлов. Такие магнитные диски представляют собой пакеты магнитных пластин (поверхностей), между которыми на одном рычаге движется пакет магнитных головок. Шаг движения пакета головок является дискретным, и каждому положению пакета головок логически соответствует цилиндр магнитного диска. На каждой поверхности цилиндр «высекает» дорожку, так что каждая поверхность содержит число дорожек, равное числу цилиндров. При разметке магнитного диска (специальном действии, предшествующем использованию диска) каждая дорожка размечается на одно и то же количество блоков таким образом, что в каждый блок можно записать по максимуму одно и то же число байтов. Таким образом, для выполнения обмена с магнитным диском на уровне аппаратуры нужно указать номер цилиндра, номер поверхности, номер блока на соответствующей дорожке и число байтов, которое нужно записать или прочитать от начала этого блока.

Однако эта возможность обмениваться с магнитными дисками порциями меньше объема блока в настоящее время не используется в файловых системах. Это связано с двумя обстоятельствами. Во-первых, аппаратура управления дисками является электронно-механической, и при выполнении обмена с диском выполняются три основных действия: подвод головок к нужному цилиндру, поиск на дорожке нужного блока и собственно обмен с этим блоком. Из всех этих действий в среднем наибольшее время занимает первое, механическое действие. Поэтому существенный выигрыш в суммарном времени обмена за счет считывания или записывания только части блока получить практически невозможно. Во-вторых, для того чтобы работать с частями блоков, файловая система должна обеспечить соответствующего размера буфера оперативной памяти, что существенно усложняет распределение оперативной памяти.

Поэтому во всех современных файловых системах явно или неявно выделяется некоторый базовый уровень. На этом уровне обеспечивается работа с файлами, состоящими из блоков, прямо адресуемых в адресном пространстве файла. Размер этих логических блоков файла совпадает или кратен размеру физического блока диска и обычно выбирается равным размеру страницы виртуальной памяти, поддерживаемой аппаратурой компьютера совместно с операционной системой.

В некоторых файловых системах базовый уровень доступен пользователю, но более часто прикрывается некоторым более высоким уровнем, стандартным для пользователей. Распространены два основных подхода.

При первом подходе, применяемом, например, в файловых системах операционных систем фирмы DEC RSX и VMS, пользователи представляют файл как последовательность записей. Каждая запись - это байтовая последовательность постоянного или переменного размера. Записи можно читать или записывать последовательно или позиционировать файл на запись с указанным номером.

Некоторые файловые системы позволяют разбивать записи на поля и объявлять некоторые поля ключами записи. В таких файловых системах можно потребовать выборку записи из файла по ее заданному ключу. Естественно, что в этом случае файловая система поддерживает в том же (или другом, служебном) базовом файле дополнительные невидимые пользователю служебные структуры данных. Распространенные способы организации ключевых файлов основываются на технике хэширования и В-деревьев. Существуют и многоключевые способы организации файлов.

Второй подход, ставший распространенным вместе с операционной системой UNIX, состоит в том, что любой файл представляется как последовательность байтов. Из файла можно прочитать указанное число байтов, либо начиная с его начала, либо предварительно произведя его позиционирование на байт с указанным номером. Аналогично, можно записать указанное число байтов в конец файла, либо предварительно выполнив позиционирование файла. Заметим, что, тем не менее, во всех разновидностях файловых систем ОС UNIX поддерживается базовое блочное представление файла, которое скрыто от пользователей.

Конечно, для обоих подходов можно обеспечить набор преобразующих функций, приводящих представление файла к некоторому другому виду. Один из примеров - наличие стандартной файловой среды системы программирования на языке Си в операционных системах фирмы DEC.

### **Именованние файлов**

Все современные файловые системы поддерживают многоуровневое именованние файлов за счет поддержания во внешней памяти дополнительных файлов со специальной структурой - каталогов. Каждый каталог содержит имена каталогов и/или файлов, содержащихся в данном каталоге. Таким образом, полное имя файла состоит из списка имен каталогов плюс имя файла в каталоге, непосредственно указывающем на данный файл. Разница между способами именованния файлов в разных файловых системах состоит в том, с чего начинается эта цепочка имен.

Имеются два крайних варианта. Во многих системах управления файлами требуется, чтобы каждый архив файлов (полное дерево справочников) целиком располагался на одном дисковом пакете (или логическом диске, разделе физического дискового пакета, представляемом с помощью средств операционной системы как отдельный диск). В этом случае полное имя файла начинается с имени дискового устройства, на котором установлен соответствующий диск. Такой способ именованния используется в файловых системах фирмы DEC, очень близко к этому находятся и файловые системы персональных компьютеров. Можно назвать эту организацию поддержанием изолированных файловых систем.

Другой крайний вариант был реализован в файловых системах операционной системы Multics. Эта система заслуживает отдельного большого разговора, в ней был реализован целый ряд оригинальных идей, но мы остановимся только на особенностях организации архива файлов. В файловой системе Multics пользователи представляли всю совокупность каталогов и файлов как единое дерево. Полное имя файла начиналось с имени корневого каталога, и пользователь не обязан был заботиться об установке на дисковое устройство каких-либо конкретных дисков. Сама система, выполняя поиск файла по его имени, запрашивала оператора об установке необходимых дисков. Такую файловую систему можно назвать полностью централизованной.

Конечно, во многом централизованные файловые системы удобнее изолированных: система управления файлами принимает на себя больше рутинной работы. Но в таких системах возникают существенные проблемы, если кому-то требуется перенести поддерево файловой системы на другую вычислительную установку.

Компромиссное решение применено в файловых системах ОС UNIX. На базовом уровне в этих файловых системах поддерживаются изолированные архивы файлов. Один из этих архивов объявляется корневой файловой системой. После запуска системы можно «смонтировать» корневую файловую систему и ряд изолированных файловых систем в одну общую файловую систему. Технически это производится с помощью создания в корневой файловой системе специальных пустых каталогов. Специальный системный вызов mount ОС UNIX позволяет подключить к одному из этих пустых каталогов корневой каталог указанного архива файлов. После монтирования общей файловой системы именованние файлов производится так же, как если бы она с самого начала была централизованной. Если учесть, что обычно монтирование файловой системы производится при раскрутке системы, то пользователи ОС UNIX обычно и не задумываются об исходном происхождении общей файловой системы.

## **Защита файлов**

Поскольку файловые системы являются общим хранилищем файлов, принадлежащих, вообще говоря, разным пользователям, системы управления файлами должны обеспечивать авторизацию доступа к файлам. В общем виде подход состоит в том, что по отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю. Существовали попытки реализовать этот подход в полном объеме. Но это вызывало слишком большие накладные расходы как по хранению избыточной информации, так и по использованию этой информации для контроля правомочности доступа.

Поэтому в большинстве современных систем управления файлами применяется подход к защите файлов, впервые реализованный в ОС UNIX. В этой ОС каждому зарегистрированному пользователю соответствует пара целочисленных идентификаторов: идентификатор группы, к которой относится этот пользователь, и его собственный идентификатор в группе. При каждом файле хранится полный идентификатор пользователя, который создал этот файл, и фиксируется, какие действия с файлом может производить его создатель, какие действия с файлом доступны для других пользователей той же группы и что могут делать с файлом пользователи других групп. Эта информация очень компактна, требующиеся при проверке действия невелики, а такой способ контроля доступа удовлетворителен в большинстве случаев.

## **Режим многопользовательского доступа**

Последнее, на чем мы остановимся в связи с файлами, это способы их использования в многопользовательской среде. Если операционная система поддерживает многопользовательский режим, вполне реальна ситуация, когда два или более пользователя одновременно пытаются работать с одним и тем же файлом. Если все пользователи собираются только читать файл, ничего страшного не произойдет. Но если хотя бы один из них будет изменять файл, для корректной работы этих пользователей требуется взаимная синхронизация.

В системах управления файлами обычно применялся следующий подход. В операции открытия файла (первой и обязательной операции, с которой должен начинаться сеанс работы с файлом) среди прочих параметров указывался режим работы (чтение или изменение). Если к моменту выполнения этой операции от имени некоторого пользовательского процесса А файл уже находился в открытом состоянии от имени некоторого другого процесса В, причем файл был открыт в режиме, который несовместим с желаемым режимом открытия (совместимы только режимы чтения), то в зависимости от особенностей системы процессу А либо сообщалось о невозможности открытия файла в желаемом режиме, либо он блокировался до тех пор, пока в процессе В не выполнялась операция закрытия файла.

Заметим, что в ранних версиях файловой системы ОС UNIX вообще не были предусмотрены какие бы то ни было средства синхронизации параллельного доступа к файлам. Операция открытия файла выполнялась всегда для любого существующего файла, если пользователь, от имени которого выполнялся процесс, имел соответствующие права доступа. При совместной работе синхронизацию приходилось производить вне файловой системы (и специальных средств для этого ОС UNIX не предоставляла). В современных реализациях файловых систем ОС UNIX по выбору поддерживается синхронизация при открытии файлов. Кроме того, существует возможность синхронизации нескольких процессов, параллельно модифицирующих один и тот же файл. Для этого введен специальный механизм синхронизационных блокировок диапазонов адресов открытого файла.

### **Тема 6.3. Области применения файлов**

После этого краткого экскурса в историю и возможности систем управления файлами рассмотрим возможные области их применения. Конечно, прежде всего файлы применяются для хранения текстовых данных: документов, текстов программ и т. д. Такие файлы обычно образуются и модифицируются с помощью различных текстовых редакторов. Структура текстовых файлов обычно очень проста: это либо последовательность записей, содержащих строки текста, либо последовательность байтов, среди которых встречаются специальные символы (например, символы конца строки).

Файлы с текстами программ являются входными параметрами компиляторов, которые, в свою очередь, формируют файлы, содержащие объектные модули. С точки зрения файловой системы, объектные файлы также обладают абсолютно стандартной структурой - последовательности записей или байтов. Система программирования накладывает на эту структуру более сложную и специфичную для этой системы структуру объектного модуля. Подчеркнем, что логическая структура объектного модуля неизвестна файловой системе, эта структура поддерживается программами системы программирования.

Аналогично обстоит дело с файлами, формируемыми редакторами связей (компоновщиками выполняемых программ) и содержащими образы выполняемых программ. Логическая структура таких файлов остается известной только редактору связей и программе-загрузчику, являющейся компонентом операционной системы.

Заметим, что в отмеченных выше случаях вполне достаточно тех средств защиты файлов и синхронизации параллельного доступа, которые обеспечивают системы управления файлами. Очень редко возникает потребность параллельной модификации файлов, и как правило, каждый пользователь может обойтись своей частной копией.

Другими словами, файловые системы обычно обеспечивают хранение слабо структурированной информации, оставляя дальнейшую структуризацию прикладным программам. В перечисленных выше случаях использования файлов это даже хорошо, потому что при разработке любой новой прикладной системы, опираясь на простые, стандартные и сравнительно дешевые средства файловой системы, можно реализовать те структуры хранения, которые наиболее естественно соответствуют специфике данной прикладной области.

### **Тема 6.4. Потребности информационных систем**

Однако для информационных систем ситуация коренным образом отличается. Эти системы главным образом ориентированы на хранение, выбор и модификацию постоянно хранимой информации. Структура информации обычно очень сложна, и хотя структуры данных различны в разных информационных системах, между ними часто бывает много общего. На начальном этапе использования вычислительной техники проблемы структуризации данных решались индивидуально в каждой информационной системе. Производились необходимые надстройки (библиотеки программ) над файловыми системами, подобно тому, как это делается в компиляторах, редакторах и т. д.

Но поскольку в информационных системах требуется поддержка сложных структур данных, эти индивидуальные средства управления данными составляли существенную часть информационных систем, практически повторяясь (как программные компоненты) от одной системы к другой. Стремление выделить общую часть информационных систем, ответственную за управление сложно структурированными данными явилось, на наш взгляд, первой побудительной причиной создания СУБД, которая, возможно, могла бы представлять некоторую общую библиотеку программ, доступную каждой информационной системе.

Однако очень скоро стало понятно, что невозможно обойтись такой общей библиотекой программ, реализующей над стандартной базовой файловой системой более сложные методы хранения данными. Поясим это на примере. Предположим, что мы хотим реализовать простую информационную систему, поддерживающую учет сотрудников некоторой

организации. Система должна выдавать списки сотрудников в соответствии с указанными номерами отделов, поддерживать регистрацию перевода сотрудника из одного отдела в другой, приема на работу новых сотрудников и увольнения работающих. Для каждого отдела должна поддерживаться возможность получения имени руководителя этого отдела, общей численности отдела, общей суммы выплаченной в последний раз зарплаты и т. д. Для каждого сотрудника должна поддерживаться возможность выдачи номера удостоверения по полному имени сотрудника, выдачи полного имени по номеру удостоверения, получения информации о текущем соответствии занимаемой должности сотрудника и о размере зарплаты.

Предположим, что мы решили реализовать эту информационную систему на основе файловой системы и пользоваться при этом одним файлом, расширив базовые возможности файловой системы за счет специальной библиотеки функций. Поскольку минимальной информационной единицей в нашем случае является сотрудник, естественно потребовать, чтобы в этом файле содержалась одна запись для каждого сотрудника. Очевидно, что поля таких записей должны содержать полное имя сотрудника (СОТР\_ИМЯ), номер его удостоверения (СОТР\_НОМЕР), информацию о его соответствии занимаемой должности (СОТР\_СТАТ - для простоты, «да» или «нет»), размер зарплаты (СОТР\_ЗАРП), номер отдела (СОТР\_ОТД\_НОМЕР). Поскольку мы хотим ограничиться одним файлом, эта же запись должна содержать имя руководителя отдела (СОТР\_ОТД\_РУК).

Для выполнения функций нашей информационной системы требуется возможность многоключевого доступа к этому файлу по уникальным ключам (недублируемым в разных записях) СОТР\_ИМЯ и СОТР\_НОМЕР. Кроме того, должна обеспечиваться возможность выбора всех записей с общим заданным значением СОТР\_ОТД\_НОМЕР, то есть доступ по неуникальному ключу. Чтобы получить численность отдела или общий размер зарплаты, информационная система должна будет каждый раз выбирать все записи о сотрудниках отдела и подсчитывать соответствующие общие значения.

Таким образом, для реализации даже такой простой системы на базе файловой системы, во-первых, требуется создание достаточно сложной надстройки, обеспечивающей многоключевой доступ к файлам, и во-вторых, неизбежны существенная избыточность хранения (для каждого сотрудника данного отдела повторяется имя руководителя отдела) и выполнение массовой выборки и вычислений для получения сводной информации об отделах. Кроме того, если в ходе эксплуатации системы возникнет потребность, например, выдавать списки сотрудников, получающих заданную зарплату, то придется либо полностью просматривать файл, либо реструктуризовывать его, объявляя ключевым поле СОТР\_ЗАРП.

Первое, что приходит на ум, - начать поддерживать два многоключевых файла СОТРУДНИКИ и ОТДЕЛЫ: первый файл должен содержать поля СОТР\_ИМЯ, СОТР\_НОМЕР, СОТР\_СТАТ, СОТР\_ЗАРП и СОТР\_ОТД\_НОМЕР, а второй - ОТД\_НОМЕР, ОТД\_РУК, СОТР\_ЗАРП (общий размер зарплаты) и ОТД\_РАЗМЕР (общее число сотрудников в отделе). Тогда большая часть неудобств, перечисленных в предыдущем абзаце, будет преодолена. Каждый из файлов будет содержать только недублируемую информацию, необходимость в динамических вычислениях сводной информации не возникнет. Но заметим, что после такого перехода информационная система будет обладать некоторыми новыми возможностями, сближающими ее с СУБД.

Прежде всего, теперь система должна знать, что она работает с двумя информационно связанными файлами (это шаг в сторону схемы базы данных), ей должны быть известны структура и смысл каждого поля (например, что СОТР\_ОТД\_НОМЕР в файле СОТРУДНИКИ и ОТД\_НОМЕР в файле ОТДЕЛЫ означают одно и то же), а также понимать, что в ряде случаев изменение информации в одном файле должно вызывать модификацию второго файла, чтобы общее содержимое файлов было согласованным. Например, если на работу принимается новый сотрудник, то необходимо добавить запись в файл СОТРУДНИКИ, а также соответствующим образом изменить поля ОТД\_ЗАРП и ОТД\_РАЗМЕР в записи файла ОТДЕЛЫ, описывающей отдел этого сотрудника.

Вообще, согласованность данных является ключевым понятием баз данных. На самом деле, если информационная система (даже такая простая, как в нашем примере) поддерживает согласованное хранение информации в нескольких файлах, можно говорить о

том, что она поддерживает базу данных. Если же некоторая вспомогательная система управления данными позволяет работать с несколькими файлами, обеспечивая их согласованность, можно назвать ее системой управления базами данных. Уже только требование поддержания согласованности данных в нескольких файлах не позволяет обойтись библиотекой функций: такая система должна обладать некоторыми собственными данными (мета-данными) и даже знаниями, определяющими целостность данных.

Но это еще не все, что обычно требуют от СУБД. Во-первых, даже в нашем примере неудобно реализовывать такие запросы, как «выдать общую численность отдела, в котором работает Иван Арутюнович Шепитько». Было бы гораздо проще, если бы СУБД позволяла сформулировать такой запрос на близком пользователям языке. Такие языки называются языками запросов к базам данных. Например, на языке SQL наш запрос можно было выразить в форме

```
SELECT ОТД_РАЗМЕР
FROM СОТРУДНИКИ, ОТДЕЛЫ
WHERE СОТР_ИМЯ = "ИВАН АРУТЮНОВИЧ ШЕПИТЬКО"
AND СОТР_ОТД_НОМЕР = ОТД_НОМЕР
```

При формулировании запроса СУБД позволит не задумываться о том, каким образом будет выполняться этот запрос. Среди ее метаданных будет содержаться информация о том, что поле СОТР\_ИМЯ является ключевым для файла СОТРУДНИКИ, а ОТД\_НОМЕР - для файла ОТДЕЛЫ, и система сама воспользуется этим. Если же возникнет потребность в получении списка сотрудников, не соответствующих занимаемой должности, то достаточно предъявить системе запрос

```
SELECT СОТР_ИМЯ, СОТР_НОМЕР
FROM СОТРУДНИКИ
WHERE СОТР_СТАТ = "НЕТ",
```

и система сама выполнит необходимый полный просмотр файла СОТРУДНИКИ, поскольку поле СОТР\_СТАТ не является ключевым.

Далее, представим себе, что в первоначальной реализации информационной системы, основанной на использовании библиотек расширенных методов доступа к файлам, обрабатывается операция регистрации нового сотрудника. Следуя требованиям согласованного изменения файлов, информационная система вставила новую запись в файл СОТРУДНИКИ и собралась модифицировать запись файла ОТДЕЛЫ, но именно в этот момент произошло аварийное выключение электрического питания. Очевидно, что после перезапуска системы ее база данных будет находиться в рассогласованном состоянии. Потребуется выяснить это (а для этого нужно явно проверить соответствие информации с файлах СОТРУДНИКИ и ОТДЕЛЫ) и привести информацию в согласованное состояние. Настоящие СУБД берут такую работу на себя. Прикладная система обязана знать, какое состояние данных является корректным, но всю техническую работу принимает на себя СУБД.

Наконец, представим себе, что мы хотим обеспечить параллельную (например, многотерминальную) работу с базой данных сотрудников. Если опираться только на использование файлов, то для обеспечения корректности изменений на все время модификации любого из двух файлов доступ других пользователей к этому файлу будет заблокирован (вспомните возможности файловых систем для синхронизации параллельного доступа). Таким образом, зачисление на работу Петра Ивановича Сидорова существенно затормозит получение информации о сотруднике Иване Сидоровиче Петрове, даже если они будут работать в разных отделах. Настоящие СУБД обеспечивают гораздо более тонкую синхронизацию параллельного доступа к данным.

Таким образом, СУБД решают множество проблем, которые затруднительно или вообще невозможно решить при использовании файловых систем. При этом существуют приложения, для которых вполне достаточно файлов, приложения, для которых необходимо решать, какой уровень работы с данными во внешней памяти для них требуется, и приложения, для которых безусловно нужны базы данных.

Современные системы управления файлами и управления базами данных представляют собой весьма совершенные инструменты, каждый из которых может быть очень успешно

применен в соответствующей области деятельности. Но всегда необходимо помнить, что каждый инструмент приносит максимальную пользу именно в той области, для которой он создан.

## **Тема 6.5. СУБД и представление данных в базах данных**

Как было показано выше, традиционных возможностей файловых систем оказывается недостаточно для построения даже простых информационных систем. Мы выявили несколько потребностей, которые не покрываются возможностями систем управления файлами: поддержание логически согласованного набора файлов; обеспечение языка манипулирования данными; восстановление информации после разного рода сбоев; реально параллельная работа нескольких пользователей. Можно считать, что если прикладная информационная система опирается на некоторую систему управления данными, обладающую этими свойствами, то эта данными является системой управления базами данных (СУБД). Тем не менее, на наш взгляд, полезно сформулировать эти (и другие) важные функции отдельно.

### **Основные функции СУБД**

К числу функций СУБД принято относить следующее:

#### **Непосредственное управление данными во внешней памяти**

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения непосредственных данных, входящих в БД, так и для служебных целей, например, для ускорения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователи в любом случае не обязаны знать, использует ли СУБД файловую систему, а если использует, то как организованы файлы. В частности, СУБД поддерживает собственную систему именования объектов БД (это очень важно, поскольку имена объектов базы данных соответствуют именам объектов предметной области).

Существует множество различных способов организации внешней памяти баз данных. Как и все решения, принимаемые при организации баз данных, конкретные методы организации внешней памяти необходимо выбирать в тесной связи со всеми остальными решениями.

#### **Управление буферами оперативной памяти**

СУБД обычно работают с БД значительного размера; по крайней мере этот размер обычно существенно превышает доступный объем оперативной памяти. Понятно, если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Единственным же способом реального увеличения этой скорости является буферизация данных в оперативной памяти. И даже если операционная система производит общесистемную буферизацию (как в случае ОС UNIX), этого недостаточно для целей СУБД, которая располагает гораздо большей информацией о полезности буферизации той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов. При управлении буферами основной памяти приходится разрабатывать и применять согласованные алгоритмы буферизации, журнализации и синхронизации. Заметим, что существует отдельное направление СУБД, которые ориентированы на постоянное присутствие в оперативной памяти всей БД. Это направление основывается на предположении, что в предвидимом будущем объем оперативной памяти компьютеров сможет быть настолько велик, что позволит не беспокоиться о буферизации. Пока эти работы находятся в стадии исследований.

#### **Управление транзакциями**

Транзакция - это последовательность операций над БД, рассматриваемых СУБД как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует (COMMIT) изменения БД, произведенные ею, во внешней памяти, либо ни одно из этих изменений

никак не отражается в состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД. Если вспомнить наш пример информационной системы отдела кадров с файлами СОТРУДНИКИ и ОТДЕЛЫ, то единственным способом не нарушить целостность БД при выполнении операции приема на работу нового сотрудника будет объединение элементарных операций над файлами СОТРУДНИКИ и ОТДЕЛЫ в одну транзакцию. Таким образом, поддержание механизма транзакций - обязательное условие даже однопользовательских СУБД (если, конечно, такая система заслуживает названия СУБД). Но понятие транзакции гораздо существеннее во многопользовательских СУБД. То свойство, что каждая транзакция начинается при целостном состоянии БД и оставляет это состояние целостным после своего завершения, делает очень удобным использование понятия транзакции как единицы активности пользователя по отношению к БД. При соответствующем управлении параллельно выполняющимися транзакциями со стороны СУБД каждый пользователь может в принципе ощущать себя единственным пользователем СУБД (на самом деле, это несколько идеализированное представление, поскольку пользователи многопользовательских СУБД порой могут ощутить присутствие своих коллег).

С управлением транзакциями в многопользовательской СУБД связаны важные понятия сериализации транзакций и сериального плана выполнения смеси транзакций. Под сериализацией параллельно выполняющихся транзакций понимается такой порядок планирования их работы, при котором суммарный эффект смеси транзакций эквивалентен эффекту их некоторого последовательного выполнения. Сериальный план выполнения смеси транзакций - это такой способ их совместного выполнения, который приводит к сериализации транзакций. Понятно, что если удастся добиться действительно сериального выполнения смеси транзакций, то для каждого пользователя, по инициативе которого образована транзакция, присутствие других транзакций будет незаметно (если не считать некоторого замедления работы для каждого пользователя по сравнению с однопользовательским режимом).

Существует несколько базовых алгоритмов сериализации транзакций. В централизованных СУБД наиболее распространены алгоритмы, основанные на синхронизационных захватах объектов БД. При использовании любого алгоритма сериализации возможны ситуации конфликтов между двумя или более транзакциями по доступу к объектам БД. В этом случае для поддержания сериализации необходимо выполнить откат (ликвидировать все изменения, произведенные в БД) одной или более транзакций. Это один из случаев, когда пользователь многопользовательской СУБД может реально (и достаточно неприятно) ощутить присутствие в системе транзакций других пользователей.

### **Журнализация**

Одно из основных требований к СУБД - надежное хранение данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев: так называемые мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания), и жесткие сбои, характеризующиеся потерей информации на носителях внешней памяти. Примерами программных сбоев могут быть аварийное завершение работы СУБД (из-за ошибки в программе или некоторого аппаратного сбоя) или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается незавершенной. Первую ситуацию можно рассматривать как особый вид мягкого аппаратного сбоя; при возникновении последней требуется ликвидировать последствия только одной транзакции.

Но в любом случае для восстановления БД нужно располагать некоторой дополнительной информацией. Другими словами, поддержание надежного хранения данных в БД требует избыточности хранения данных, причем та их часть, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенный метод поддержания такой избыточной информации - ведение журнала изменений БД.



Журнал - это особая часть БД, недоступная пользователям СУБД и поддерживаемая особо тщательно (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. В разных СУБД изменения БД журналируются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), а порой запись соответствует минимальной внутренней операции модификации страницы внешней памяти. В некоторых системах одновременно используются оба подхода.

Во всех случаях придерживаются стратегии «упреждающей» записи в журнал (так называемого протокола Write Ahead Log - WAL). Грубо говоря, эта стратегия заключается в том, что запись об изменении любого объекта БД должна попасть во внешнюю память журнала раньше, чем измененный объект попадет во внешнюю память основной части БД. Известно, если в СУБД корректно соблюдается протокол WAL, то с помощью журнала можно решить все проблемы восстановления БД после любого сбоя.

Самая простая ситуация восстановления - индивидуальный откат транзакции. Строго говоря, для этого не требуется общесистемный журнал изменений БД. Достаточно для каждой транзакции поддерживать локальный журнал операций модификации БД, выполненных в этой транзакции, и производить откат транзакции выполнением обратных операций, следуя от конца локального журнала. В некоторых СУБД так и делают, но в большинстве систем локальные журналы не поддерживают, а индивидуальный откат транзакции выполняют по общесистемному журналу, для чего все записи от одной транзакции связывают обратным списком (от конца к началу).

При мягком сбое во внешней памяти основной части БД могут находиться объекты, модифицированные транзакциями, не закончившимися к моменту сбоя, и могут отсутствовать объекты, модифицированные транзакциями, которые к моменту сбоя успешно завершились (по причине использования буферов оперативной памяти, содержимое которых при мягком сбое пропадает). При соблюдении протокола WAL во внешней памяти журнала должны гарантированно находиться записи, относящиеся к операциям модификации обоих видов объектов. Целью процесса восстановления после мягкого сбоя является состояние внешней памяти основной части БД, которое возникло бы при фиксации во внешней памяти изменений всех завершившихся транзакций и которое не содержало бы никаких следов незаконченных транзакций. Чтобы этого добиться, сначала производят откат незавершенных транзакций (undo), а потом повторно воспроизводят (redo) те операции завершенных транзакций, результаты которых не отображены во внешней памяти. Этот процесс содержит много тонкостей, связанных с общей организацией управления буферами и журналом. Более подробно мы рассмотрим это в соответствующей лекции.

Для восстановления БД после жесткого сбоя используют журнал и архивную копию БД. Грубо говоря, архивная копия - это полная копия БД к моменту начала заполнения журнала (имеется много вариантов более гибкой трактовки смысла архивной копии). Конечно, для нормального восстановления БД после жесткого сбоя необходимо, чтобы журнал не пропал. Как уже отмечалось, к сохранности журнала во внешней памяти в СУБД предъявляются особо повышенные требования. Тогда восстановление БД состоит в том, что исходя из архивной копии по журналу воспроизводится работа всех транзакций, которые закончились к моменту сбоя. В принципе можно даже воспроизвести работу незавершенных транзакций и продолжить их работу после конца восстановления. Однако в реальных системах это обычно не делается, поскольку процесс восстановления после жесткого сбоя является достаточно длительным.

## **Раздел 7. ДОРЕЛЯЦИОННЫЕ СТРУКТУРЫ И МОДЕЛИ ДАННЫХ**

### **Тема 7.1. Различные представления о данных в базах данных**

Создание базы данных предполагает интеграцию данных, предназначенных для решения нескольких прикладных задач разных пользователей. Соответственно, при интеграции

данных должны учитываться требования к данным каждого пользователя, основанные на его представлении о данных и связях между ними. Далее эти требования должны обобщаться в единое представление, которое и будет служить основой для построения единой базы данных (рис. 7.1).

Обобщение представлений всех пользователей о данных называется **концептуальной моделью** (схемой) БД. **Концептуальная модель** представляет информационное описание предметной области с учетом логических взаимосвязей, поэтому её еще называют инфологической (информационно-логической) моделью. В модели отсутствуют какие-либо понятия, связанные с ЭВМ, памятью ЭВМ, способами размещения данных в памяти ЭВМ, и, по сути, это модель только предметной области.

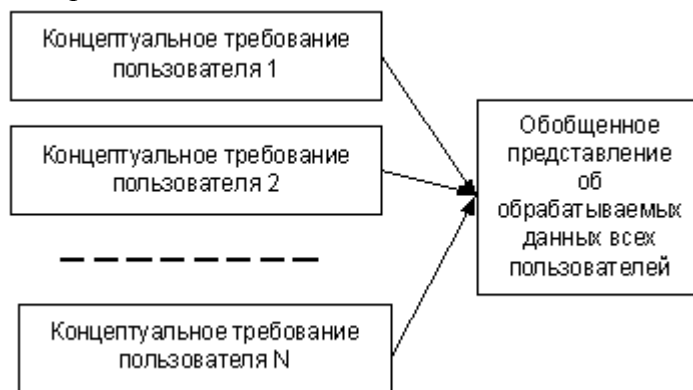


Рис. 7.1. Обобщение представления пользователей о данных

Как уже отмечалось, для создания базы данных и работы с ней используется система управления базами данных. Каждая конкретная СУБД поддерживает определенный вид данных (форматов записей и отношений), называемый **моделью данных СУБД**.

Следующий этап разработки базы данных предполагает выбор представления **концептуальной модели** с помощью модели данных конкретной СУБД. **Полученное таким образом представление концептуальной модели называется логической моделью БД. Или другими словами, логическая модель – это концептуальная схема, специфицированная в языке конкретной СУБД.** Логическая модель представляет данные и элементы данных вне зависимости от их содержания и среды хранения. Далее разработчик системы средствами СУБД отображает полученную логическую модель БД в память ЭВМ и определяет методы доступа. Полученное представление данных в памяти ЭВМ называется внутренним представлением или структурой хранения. Прикладные программы работают с логической моделью, причем каждому пользователю представляется подмножество этой логической модели (подсхема), отражающее его представление о предметной области. Каждая прикладная программа "видит" и обрабатывает только те данные, которые необходимы именно ей.

**Соответствующее «видение» данных прикладными программами (пользователями) представляет собой внешние представления.** Взаимосвязь вышеуказанных моделей изображена на рис. 7.2.

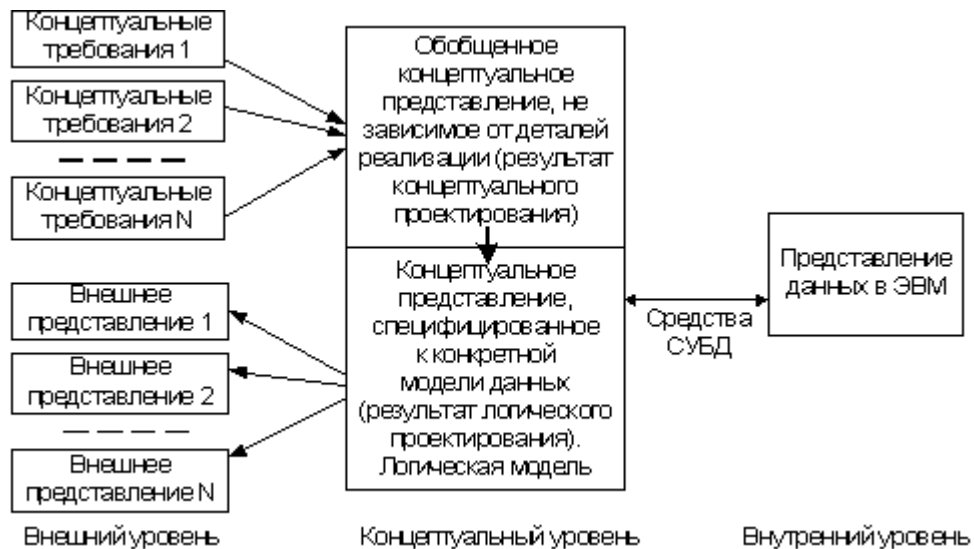


Рис. 7.2. Различные представления о данных в БД

На данной схеме выделены три различных уровня описания данных (внешний, концептуальный, внутренний). **Эти уровни формируют так называемую трехуровневую архитектуру ANSI/SPARC**, предложенную в 1975 г. Комитетом планирования стандартов и норм SPARC (Standards Planning and Requirements Committee) Национального института стандартизации США (American National Standards Institute – ANSI). Основная цель этой архитектуры состоит в отделении пользовательского представления о данных в базе данных от их физического представления. Использование таких представлений о данных позволяет обеспечить выполнение основного требования к БД – независимости программ и данных. При изменении прикладных программ может измениться соответствующее *внешнее представление*, но *логическая модель* данных не изменяется и, соответственно, не будут изменяться другие прикладные программы. При изменении внутреннего представления (структур хранения) *логическая модель* не изменяется, соответственно, не изменяются прикладные программы.

Использование соответствующих представлений также позволяет четко разграничить полномочия различных лиц, работающих с базой данных.

Соответствующие представления позволяют описать «видение» базы данных разными лицами, работающими с ней:

- *внешнее представление* – представление специалиста предметной области (пользователя);
- *внешнее представление* и *логическая модель* – представление прикладного программиста, разрабатывающего конкретное приложение для пользователя;
- *логическая модель* и *внутреннее представление* – представление системного программиста, администрирующего базу данных.

## Тема 7.2. Основные этапы проектирования базы данных

*Проектирование данных (базы данных)* представляет собой процесс последовательного отображения исследуемых явлений реального мира в виде данных в памяти ЭВМ (рис. 7.3).

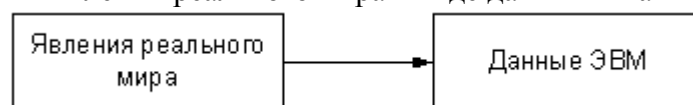


Рис. 7.3. Общая схема проектирования

**Конкретные явления реального мира, представляющие интерес для проводимого исследования, будем называть предметной областью.**

Проектирование (моделирование) базы данных представляет собой многоэтапный процесс. Основные этапы этого процесса приведены на рис. 7.4.).

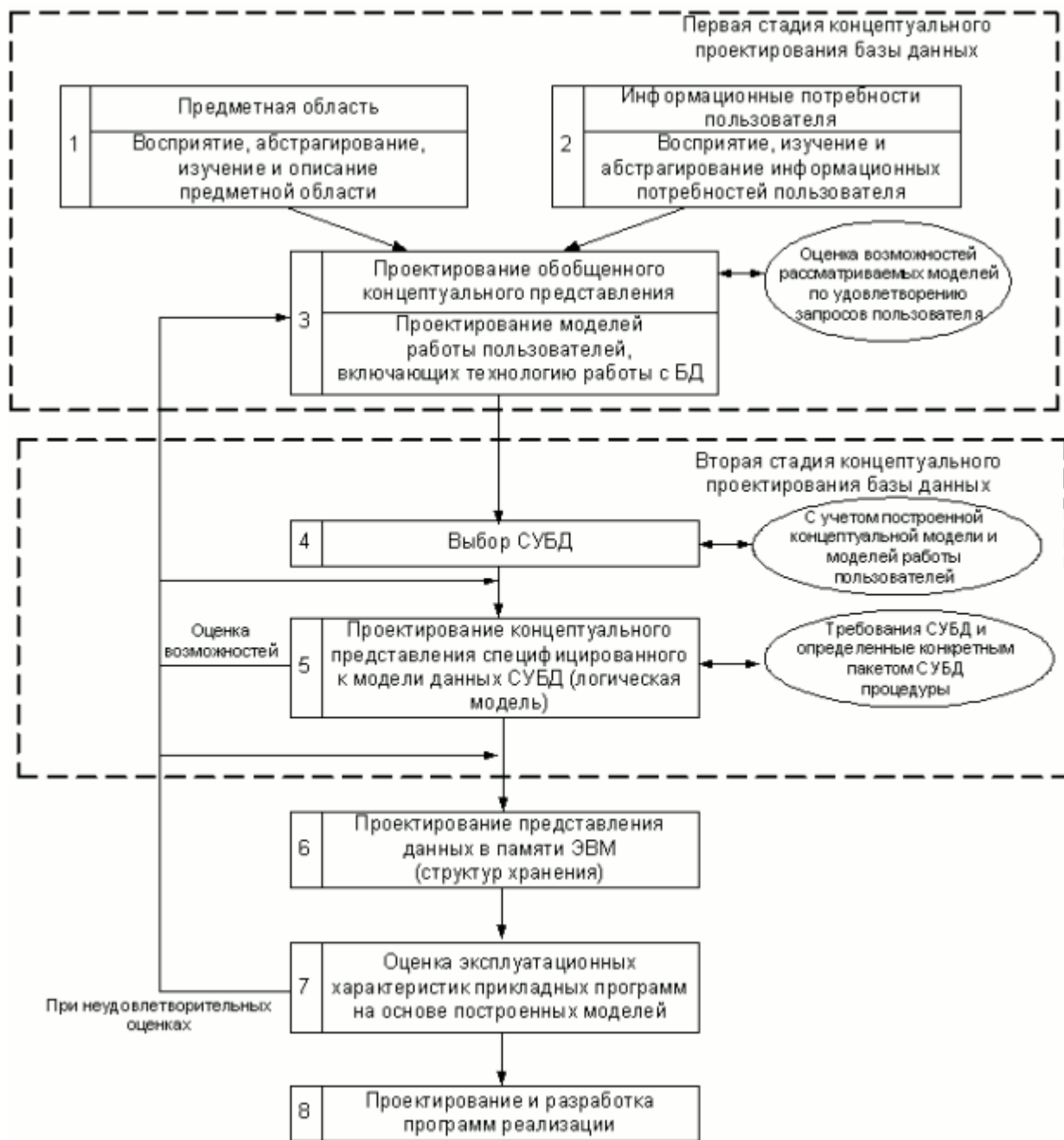


Рис. 7.4. Этапы проектирования базы данных

Подробно действия, отраженные на приведенном рисунке, будут рассмотрены в следующих лекциях. Здесь дадим лишь краткие комментарии к соответствующим блокам.

В блоках 1,2 необходимо особое внимание обратить на слово «абстрагирование». Имеется в виду, что проектирование базы данных нужно вести не под конкретный документ, обрабатываемый пользователем, и не под конкретные действия пользователя с этим документом, а под обобщенный (абстрактный) образ документов и обобщенные (абстрактные) действия пользователей. Например, рассматривать документ не с конкретными числами строк и столбцов, а с абстрактными числами  $n$  и  $m$ ; вместо требуемого пользователем поиска по конкретному полю (например, фамилии) рассматривать поиск по любому полю и т.д. Это очень важно, так как конкретные формы документов и действия пользователей при работе с ними достаточно часто изменяются. В этом случае при проектировании базы данных под конкретные формы документов и конкретные действия

придется перепроектировать базу данных, что связано с существенными временными и стоимостными затратами.

Очень важным является выбор СУБД (блок 4), от которого в значительной степени зависит работоспособность построенной базы данных. Проблема выбора СУБД уже обсуждалась в лекции 3. Заметим здесь, что выбор СУБД зависит от количества форм документов, от сложности связей между данными, от объема обрабатываемых данных, от количества пользователей, работающих с БД и т.д.

Ранее отмечалось, что отображение логической модели базы данных в структуру хранения (представление данных в памяти компьютера) осуществляется системой управления базой данных. Тем не менее, во многих СУБД для повышения эффективности функционирования базы данных представляется возможность выбора ряда параметров, управляющих представлением данных в памяти компьютера. Выбор таких параметров и подразумевается в блоке 6.

Заметим, что очень важно при *проектировании базы данных* делать оценки ее возможной работоспособности. Так, по завершении проектирования обобщенного концептуального представления нужно попытаться оценить необходимое число производимых операций с элементами моделей при реализации возможных запросов пользователей. При невозможности в рамках построенной модели ответить на какой-то запрос пользователя или при значительном числе производимых при этом операций (что приведет к невозможности реализации соответствующего запроса в реальном масштабе времени) необходим возврат по схеме рис. 7.4. на шаг назад (построение более эффективного обобщенного концептуального представления). Аналогичные оценки необходимо делать и при завершении других этапов проектирования (блоки 5, 7). При этом возможен возврат назад на один или несколько шагов. Так, например, при проектировании логической модели (блок 5) не удастся достичь адекватного представления *концептуальной модели* средствами *модели данных СУБД*. В этом случае необходимо либо вернуться на шаг назад и выбрать другую СУБД, либо вернуться к блоку 3 и изменить вид *концептуальной модели*. Так же, если полученные при реализации блока 7 оценки эксплуатационных характеристик не отвечают требованиям пользователя, возможны пересмотры всех ранее полученных решений (блоки 7, 6, 5, 4, 3). Кроме этого, необходим возврат на проектирование обобщенного концептуального представления при изменении внешних требований пользователей, а также при выявленных ошибках проектирования.

## **Раздел 8. РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ**

### **Тема 8.1. Модели данных**

Концептуальная модель предметной области достаточно полно и точно отражает систему состояний ПО. Однако она не может быть использована как динамическая модель базы данных ФИПС по нескольким причинам, основными из которых являются:

- концептуальная модель отображает только взаимосвязи типов сущностей, а не элементов;
- концептуальная модель ПО не отображает динамику изменений ПО;
- концептуальная модель ПО не достаточно формализована для ее восприятия компьютером.

От всех этих и многих других проблем избавляет переход на следующий уровень представления данных - логическую модель БД. Однако логическое моделирование предполагает жесткую структуризацию данных и операций их преобразования. Это достигается с помощью модели данных, как средства формализованного представления данных и операций их обработки с учетом семантики предметной области. Модель данных включает:

- структуру данных;
- операции над данными;

- ограничение целостности.

Одним из основных способов структуризации данных является использование абстракций. Абстракция позволяет облегчить и улучшить процесс понимания сложных предметов путем сосредоточения внимания только на их существенных свойствах и связях. При структуризации данных применяется два типа абстракции: обобщение и агрегация. Обобщение позволяет трактовать класс различных *подобных* объектов как один поименованный обобщенный тип объекта. В обобщении подчеркивается общая природа объектов. В случае многоуровневой иерархии обобщений структура обобщения образует родовую иерархию, что приводит к появлению понятий рядовой и видовой сущности. Иерархию обобщений иллюстрирует рис. 8.1.

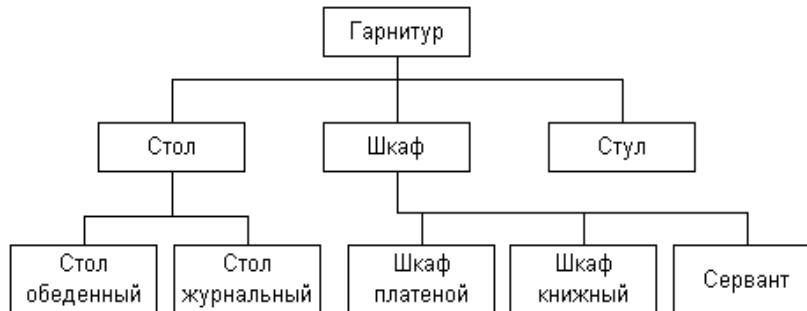
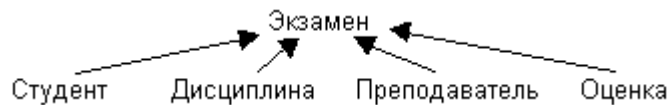


Рис. 8.1.

Агрегация позволяет рассмотреть связь между элементами данных как новый элемент более высокого уровня. Например, связь между сущностями СТУДЕНТ, ДИСЦИПЛИНА, ПРЕПОДАВАТЕЛЬ, ОЦЕНКА, имеющая смысловое описание «студент по фамилии Иванов получил на экзамене по дисциплине «Высшая математика» у преподавателя по фамилии Петров оценку хорошо», может быть представлена агрегированным элементом – ЭКЗАМЕН:



На основе обобщения и агрегации можно из одних категорий строить другие более общие, что используется в различных моделях данных (МД). Определение *структуры данных* некоторого типа на формализованном языке называют ее схемой. Схема задает совокупность свойств соответствующих данному типу структуры.

Правила композиции схемы и правила генерации реализаций схемы регламентируются языком описания данных (ЯОД), на котором и задаются схемы структур данных.

Современные ФИПС используют несколько разных типов структур данных, каждый из которых ориентирован на формализованное представление данных определенных предметных областей. На практике используют структуры данных трех наиболее распространенных типов. Самой простой и наиболее распространенной в использовании является плоская структура данных, которая включает следующие составляющие (рис. 8.2):

- поле - наименьшая поименованная единица данных;
- запись - поименованная совокупность полей;
- файл - поименованная совокупность экземпляров записей;
- набор-файлов - поименованная совокупность файлов, обрабатываемых системой.



Рис. 8.2. Плоская структура данных

Другой, широко распространенной структурой, является структура данных КОДАСИЛ. КОДАСИЛ - это ассоциация по языкам систем обработки данных. Структура данных этой модели представлена на рис. 8.3.

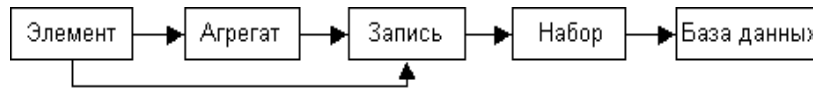


Рис. 8.3. Структура данных КОДАСИЛ

Исходной составляющей этой структуры является элемент данных (ЭД), представляющий собой наименьшую поименованную единицу данных, к которой можно адресоваться непосредственно и с помощью которой выполняется построение всех остальных составляющих структуры. Имя элемента данных используется для его идентификации в схеме составляющих структуры данных более высокого уровня. Значение ЭД может быть числового и нечислового типа. Множество значений, которые может представлять ЭД, называют областью определения.

Под агрегатом данных (АД) понимают поименованную совокупность ЭД. Агрегат данных может быть простым, если состоит только из элементов данных, или составным, если включает в свой состав другие агрегаты.

Запись – поименованная совокупность элементов данных и(или) агрегатов. Набор – это поименованная совокупность записей, образующих двухуровневую иерархическую структуру. Понятие набора позволяет образовывать структуры, представляющие связь одной записи со многими, а порядок внутри набора обеспечивает последовательную связь. Пример набора приведен на рис. 8.4.



Рис. 8.4. Пример набора

База данных представляет собой поименованную совокупность записей и наборов. Описание базы данных задается ее схемой.

Рассмотренные *типы* структур данных носят абстрактный характер и не имеют какой-либо определенной формы представления. Однако, в моделировании данных форма представления данных является весьма важной, поскольку необходимо, во-первых, хранить данные в памяти ЭВМ, а во-вторых, представлять их пользователю в форме, облегчающей интерпретацию и понимание.

Простейшей формой представления данных являются таблицы. При машинной обработке в качестве аналогов таблиц и их элементов могут выступать файлы, записи, поля. Другим способом представления данных является графовая форма.

Операции над данными – это допустимые действия над некоторой реализацией базы данных для перевода ее из одного состояния в другое. Динамические свойства модели данных определяются множеством операций над данными. Любая операция над данными включает в себя две компоненты: селекцию и действие. Если селекция определяет критерий отбора данных, над которыми должно быть произведено действие, то действие определяет характер операции. Действие в модели данных обычно выражается одной из следующих операций:

- идентификация данного и нахождение его в позиции БД;
- выборка данных из БД;
- добавление новых данных в БД;
- удаление данных из БД.

Указанные процедуры выполняются по схеме «селекция – действие». Однако существуют и другие операции, выполняемые по другим схемам и реализующие более сложные функции, например, функцию поддержания целостности. Такие обобщенные операции над базой данных называют процедурами БД. Процедуру рассматривают как

единую макрооперацию. Использование процедур позволяет существенно расширить динамические свойства модели данных.

Логические ограничения, накладываемые на данные, называют ограничениями целостности. Они вместе со схемой структур данных отражают свойства данных. Ограничения вводятся в модель данных в целях повышения ее семантической и связанной с этим адекватностью отражения реального мира в модели, а также для обеспечения непротиворечивости данных заданным ограничениям при переводе БД из одного состояния в другое.

СУБД проверяет непротиворечивость системы ограничений и при своем функционировании обеспечивает целостность данных в базе данных по отношению к заданным ограничениям.

## Тема 8.2. Типы моделей данных

Все многообразие существующих моделей данных подразделяют на несколько типов, наиболее известными из которых являются реляционная, сетевая и иерархическая. Если сетевая и иерархическая модели данных применяются с начала 60-х годов, то реляционная модель данных была предложена в начале 70-х годов. Эти три модели различаются в основном способами представления взаимосвязей между объектами.

### Иерархическая модель данных

При построении иерархической модели данных считают, что вся предметная область (представляющая некоторый класс объектов) разбивается на подклассы, подклассы на подподклассы и т.д. Таким образом, предметная область представляет собой иерархию (дерево) частей предметной области с одной вершиной. Структуру иерархического дерева называют иерархической схемой данных. Совокупность данных о предметной области, представленных в соответствии с иерархическим деревом, называют иерархической базой данных.

Иерархическая МД организует данные в виде иерархической древовидной структуры, которая состоит из узлов и ветвей. Наивысший узел называют корнем; он находится на первом уровне, зависимые узлы располагаются на более низких уровнях (рис. 8.5).

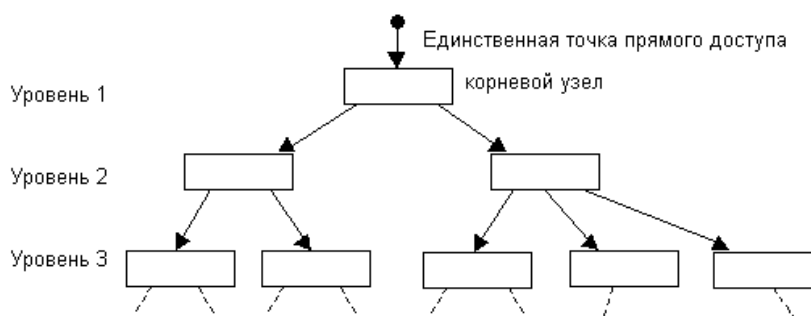


Рис. 8.5. Схема структуры данных иерархической базы данных

Иерархическая древовидная структура всегда удовлетворяет следующим условиям:

- иерархия начинается с корневого узла;
- на первом уровне ( $i=1$ ) может находиться только один узел – корневой;
- на нижних уровнях ( $i=2,3,\dots,n$ ) находятся порожденные (зависимые) узлы;
- каждый порожденный узел, находящийся на  $i$ -м уровне, связан только с одним непосредственно исходным узлом, находящимся на более высоком уровне ( $i-1$ ) иерархии дерева;
- каждый исходный узел может иметь один или несколько непосредственных узлов, которые называют подобными;
- доступ к каждому порожденному узлу выполняется через его непосредственно исходный узел;



- существует единственный иерархический путь доступа к любому узлу, начиная с корня дерева.

Основными структурными единицами модели данных являются: элемент данных, агрегат данных и запись. В агрегатах допускается множественный элемент, который содержит несколько значений элемента в одном экземпляре агрегата. Запись в общем случае является составным агрегатом, который не входит в состав других агрегатов. Таким образом, структура записи в данных моделях может быть иерархической и линейной. Запись конкретного типа является объектом в модели данных. Пример иерархической записи «Договор», где представлена вся структура информации о договорах с заказчиками на поставку изделий, представлен на рис. 8.6.

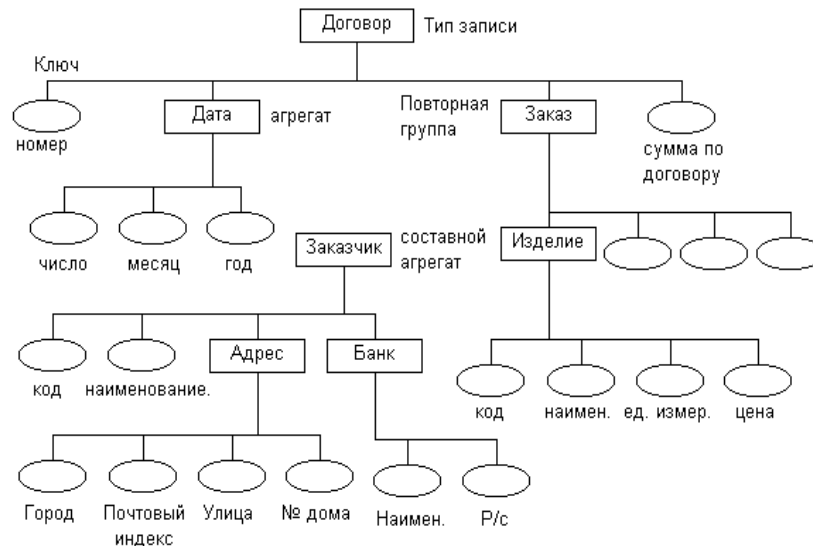


Рис. 8.6. Пример записи «Договор»

Следует отметить, что в этой записи элементарные данные, имеющие значения, представлены лишь в конечных вершинах записи, а остальные типы структур данных, в том числе агрегаты, являются лишь именованной совокупностью данных.

К основным ограничениям относятся следующие:

- все типы связей должны быть вида 1:1, 1:n, n:1; представление связей типа m:n непосредственно невозможно;
- структура связи должна быть *древовидной*.

Ограничения, которые иерархическая модель данных накладывает на связь между записями, порождают определенные особенности выполнения операций манипулирования данными. Так, например, при добавлении данных необходимо учитывать, что экземпляр порожденного узла не может существовать в отсутствии исходного узла, а при удалении экземпляра исходного узла также удаляются и все экземпляры порожденных узлов.

Недостатки использования иерархических МД:

- взаимосвязи типа m:n могут быть реализованы только искусственно, но структура становится громоздкой, а также требуется избыточное хранение данных;
- из-за строгой иерархической упорядоченности объектов модели значительно усложняются операции добавления и удаления;
- особенности иерархических структур обуславливают процедурность манипулирования данными;
- так как корневой тип узла является главным, то доступ к любому порожденному узлу возможен только через исходный.

Достоинства иерархической модели следующие:

- простота; хотя модель использует три информационные конструкции, иерархический принцип является естественным для многих экономических задач (например, организация статистической отчетности);
- независимость данных;
- минимальный расход памяти.

### Сетевая модель данных

Применение сетевых МД базируется на гипотезе о сетевой структуре моделируемой ПО, когда вся область рассматривается как совокупность частей, связанных между собой бинарными связями различных типов. В свою очередь части поддаются аналогичной декомпозиции. Структуру данных, представленную сетью, называют сетевой схемой. Совокупность данных, представленных в соответствии с сетевой схемой данных, называют сетевой БД. Сетевые модели данных базируются на использовании графовой формы представления данных. Вершины графа используются для интерпретации типов сущностей, а дуги – типов связи (рис. 8.7).

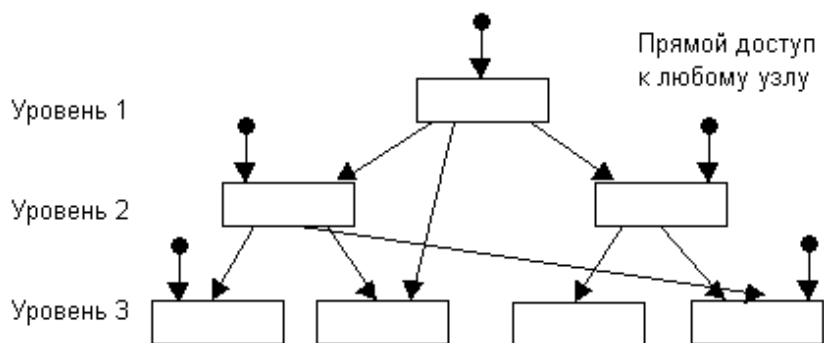


Рис. 8.7. Схема структуры данных сетевой БД

В этой модели данных используются две основные структуры данных: типы записей и наборы. Типы записей определяют обычным образом как совокупности логически связанных записей. В сетевых моделях данных непосредственный доступ может обеспечиваться к любому объекту независимо от уровня, на котором он находится в модели. Возможен также доступ по связям от любой точки доступа. Ограничением модели является функциональность связей, что делает невозможным непосредственное представление связей типа  $m:n$ . Такой тип связи может быть представлен вспомогательным типом записи и двумя функциональными связями типа  $1:m$  и  $n:1$  (рис. 8.8).

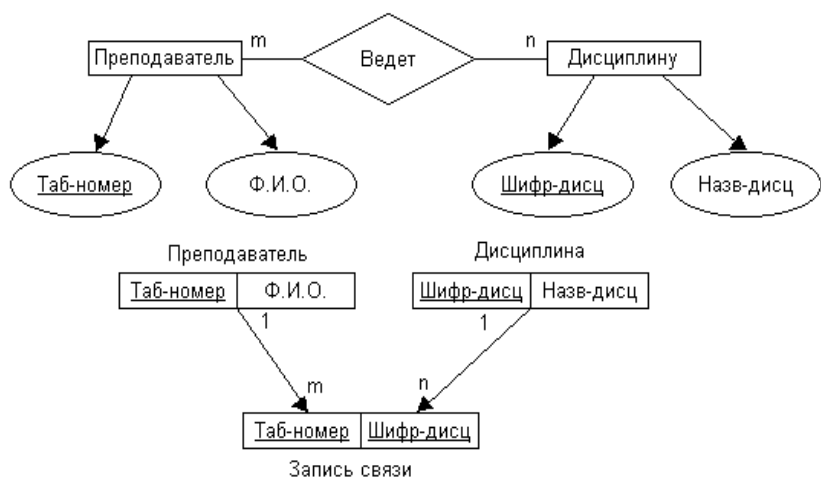


Рис. 8.8. Представление связи типа « $m:n$ »

Наряду с ограничением на функциональные связи в сетевой МД имеются и особенности манипулирования данными, а именно СУБД должна обеспечивать:

- установление позиции данного в БД безотносительно к связи между записями;
- навигацию по связям, соединяющим записи.

Следует отметить, что язык манипулирования данными современных СУБД позволяет достаточно гибко управлять выполнением этих операций.

Недостатками сетевой модели являются:

- возможность потери независимости данных при реорганизации БД;
- используемое прикладное программное обеспечение сложнее, чем в иерархической модели;
- допустимость только навигационного принципа доступа к данным.

Сетевая модель на примере БД, содержащей: справочные данные о цехах; плановые данные выпуска изделий цехами; учетных данных о сдаче выпущенных цехами изделий на склад, приведена на рис. 8.8.

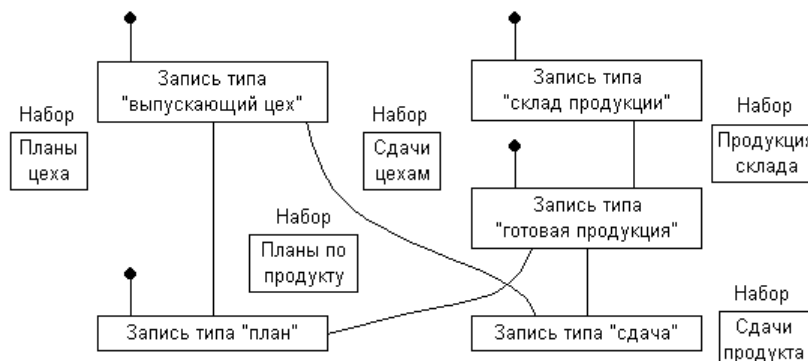


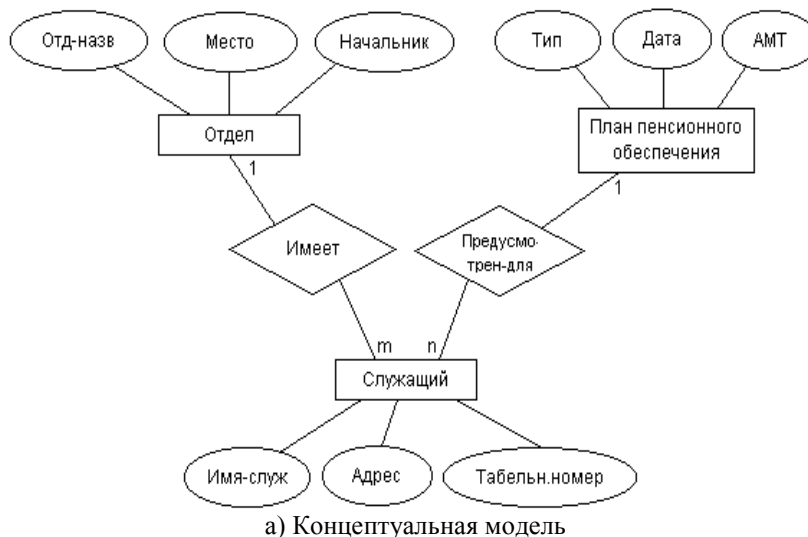
Рис. 8.8. Пример БД, реализованной сетевой моделью

Сетевые МД по сравнению с иерархическими является более универсальным средством отображения структуры информации для разных ПО. Следует отметить, что взаимосвязи данных большинства ПО имеют сетевой характер, что ограничивает использование СУБД с иерархической моделью данных.

Сетевые модели позволяют обрабатывать также иерархические взаимосвязи данных. Достоинством сетевых моделей являются отсутствие дублирования данных в различных элементах модели. Кроме того, технология работы с сетевыми моделями является удобной для пользователя, так как доступ к данным практически не имеет ограничений и возможен непосредственно к объекту любого уровня, а также допустимы всевозможные запросы. Примерный набор операций может быть следующим:

- найти конкретную запись в наборе однотипных записей (инженера Сидирова);
- создать новую запись;
- уничтожить запись;
- модифицировать запись;
- включить в связь;
- исключить из связи;
- переставить в другую связь и т.п.

Пример сетевого и иерархического представления концептуальной модели данных, в которой определены связи типа 1:n, приведена на рис. 8.9.



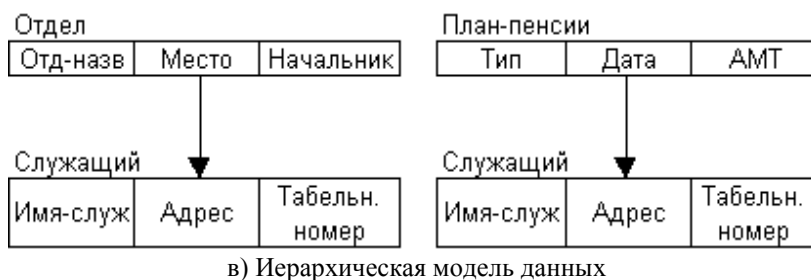
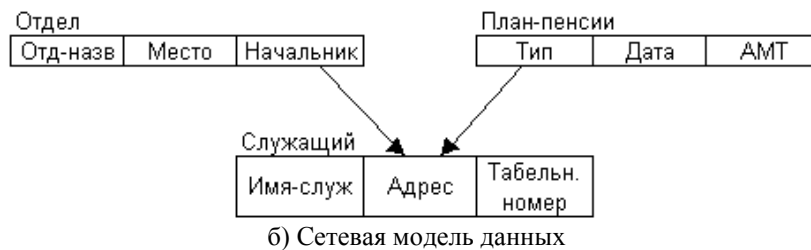


Рис. 8.9. Концептуальная модель и ее сетевое и иерархическое представления

Из рис. 8.9. следует, что:

1. Принципиальная разница между сетевой и иерархической моделями состоит в том, что в сетевой модели потомок может принадлежать нескольким предкам, тогда как в иерархической модели – только одному. (Предок – это точка в «хвосте» стрелки в иерархической структуре данных. Потомок – это точка на острие стрелки в иерархической модели.);

2. В иерархической модели данных присутствует избыточность, т.к. каждое вхождение сегмента Служащий записывается дважды (сегмент аналогичен типу записей сетевой модели). Следует отметить, что при записи избыточности не возникает, т.к. первое вхождение сегмента (в нашем примере, Служащий) будет запоминаться обычным образом, а последующие вхождения будут запоминаться не как действительные сегменты, а как указатели, т.е. физические адреса на диске, обозначающие место хранения данного сегмента. Однако, это ведет к дополнительному объему памяти для указателей.

### Реляционная модель данных

Реляционные модели данных отличаются от рассмотренных выше сетевых и иерархических простой структурой данных, удобным для пользователей табличным представлением и доступом к данным.

Таблица является основным типом структуры данных (объектом) реляционной модели. Структура таблицы определяется совокупностью столбцов. В каждой строке таблицы содержится по одному значению в соответствующем столбце. В ней не может быть двух одинаковых строк, при этом общее число строк не ограничено.

Столбцы уникально именуется в рамках отношений и в языке данных используются только эти имена, называемые атрибутами, а не номера столбцов. Каждому столбцу отношения соответствует определенное свойство описываемых отношением объектов. Как правило, это свойство отражается в его наименовании.

Таким образом, атрибут – это информационное отображение отдельного свойства некоторого объекта, процесса или явления. Один или несколько атрибутов, значения которых однозначно идентифицируют строки таблицы, являются ключом таблицы. Столбец таблицы со значениями соответствующего атрибута называют доменом, а строку со значениями разных атрибутов – кортежем.

Кортежи не должны повторяться внутри таблицы-отношения и, соответственно, они должны иметь уникальный идентификатор – первичный ключ. Первичный ключ называют простым, когда он состоит из одного атрибута, или составным, когда он состоит из нескольких атрибутов. Первичный ключ не должен иметь дополнительных атрибутов – это значит, что если единичный произвольный атрибут исключить из первичного ключа, то оставшихся атрибутов будет недостаточно для однозначной идентификации отдельных

кортежей. Кроме первичного ключа, в отношении могут существовать и вторичные ключи. Вторичный ключ – это такой ключ, значения которого могут повторяться в разных строках – кортежах. По ним может отыскиваться группа строк с одинаковым значением вторичного ключа. Двумерные таблицы, называемые отношениями, являются универсальным объектом реляционных моделей.

Строка отношения описывает свойства определенного объекта предметной области и может рассматриваться как формальная запись знаний о свойствах этого объекта. Однородность объектов, описываемых одним отношением, определяется тем, что все они имеют одинаковый набор свойств, представленный в отношениях атрибутами. Совокупность реляционных таблиц образуют реляционную базу данных (РБД). Отношения в РБД имеют уникальные имена, как правило, имя отношения связано с некоторым классом объектов, описываемой предметной области. Совокупность имени отношения и его атрибутов называется схемой отношения, которую можно рассматривать как формальную запись знания о том, какой класс объектов предметной области представлен в базе данных и какие свойства объектов этого класса описываются данным отношением.

Совокупность схем всех отношений, входящих в базу, называют схемой базы данных. Она содержит достаточно полную систему знаний о том, какие знания и о каких объектах предметной области содержатся в базе данных. На рис. 8.10 приведен пример реляционной базы данных, которая содержит три типа информации о строительной компании:

1. Информация о поставщиках, поставляющих детали предприятию. Сюда относятся номер поставщика (Пном – предполагается уникальн), его фамилия (Пфам) и город проживания (не является уникальным). Данная информации содержится в отношении ПРОИЗВОДИТЕЛЬ;

2. Информация о деталях используемых на предприятии. Сюда относятся номер детали (Дном – является уникальным), название (Дназ), цвечи вес (не уникальные). Эта информация содержится в отношении ДЕТАЛЬ;

3. Информация о номерах (Пном) и количестве деталей от каждого поставщика. Эта информация содержится в отношении ПД.

Схема реляционной БД включают следующие компоненты:

$S(\text{rel}) = \langle A, R, \text{Dom}, \text{Real}, V(s) \rangle$ ,

где  $A$  – множество имен атрибутов;  $R$  – множество имен отношений,  $\text{Dom}$  – вхождение атрибутов в домены;  $\text{Rel}$  – вхождение атрибутов в отношения (описание структуры отношений);  $V(s)$  – множество ограничений в базе данных (в том числе функциональных зависимостей).



Рис. 8.10. Пример реляционной базы данных

Достоинствами реляционной модели являются:

- простота, поскольку в модели используется всего лишь одна информационная конструкция, которая формализует табличное представление данных, привычное для пользователей-экономистов;
- теоретическое обоснование, что позволяет получить базы данных с заранее заданными свойствами;
- независимость данных, что приводит к минимальным изменениям в прикладных программах при изменении структуры РБД;
- точность, связность, секретность и др.

К недостаткам такой модели относят:

- низкое быстродействие при реализации операции соединения;
- большой расход памяти для ее представления.

Данная модель является наиболее перспективной с точки зрения развития, а также использования в интеллектуальных системах.

Число ограничений в реляционной МД весьма невелико, что обеспечивает достаточную свободу в выборе представления типов связей и сущностей.

Основное ограничение связано с тем, что в табличном представлении отношения дублирование строк не допускается, т.е. невозможно представление в отношении кортежей-дубликатов. Это ограничение удовлетворяется введением идентификатора кортежа или первичного ключа отношения.

Ограничения, накладываемые на атрибуты, могут быть весьма разнообразными. Например, можно специфицировать область значения атрибута, стандартные единицы измерения и др. Современные языки описания данных реляционных СУБД имеют довольно развитые средства для описания явных ограничений.

Следует отметить, что большинство явных ограничений, встречающихся на практике – это ограничения на зависимости между атрибутами. Поскольку в РМД ограничения задаются явно, то можно проводить самостоятельное исследование зависимостей между атрибутами. Данное исследование позволяет грамотно проектировать схемы БД; получать схемы, обладающие необходимыми свойствами.

Для реляционной модели языка манипулирования данными (ЯМД) представлены двумя классами языков: алгебраическими и языками исчислений.

Языки первого типа относятся к классу процедурных. Запрос описывается средствами реляционной алгебры.

Множества операций над отношениями, реализуемых алгебраическими языками, разбивают на два подмножества:

- стандартные теоретико-множественные операции: объединение, пересечение, разность, декартово произведение;

- специальные операции: проекция, соединение, селекция, частное.

Различают унарные операции (операции над одним отношением) и бинарные операции (операции над двумя отношениями).

Основные операции реляционной алгебры приведены в табл. 8.1.

Таблица 8.1 – Основные операции реляционной алгебры

Название операции	Обозначение операции	Определение операции	Условия реализации	Пример реализации
Объединение	$R=R_1 \cup R_2$	Множество кортежей, принадлежащих $R_1$ и $R_2$ , либо им обоим	Одинаковая арность $R_1$ и $R_2$ ; отношение $R$ той же арности	рис. 8.11 (а,б)
Разность (вычитание)	$R=R_1 \setminus R_2$	Множество кортежей, принадлежащих $R_1$ , но не принадлежащих $R_2$		рис. 8.11 (а,в)
Пересечение	$R=R_1 \cap R_2$	Множество кортежей, принадлежащих и $R_1$ , и $R_2$		рис. 8.11 (а,г)
Декартово произведение	$R=R_1 R_2$	$R$ –множество кортежей арности $(k_1+k_2)$ , причем первые $k_1$ элементов образуют кортеж из $R_1$ , а последние $k_2$ –из $R_2$	при $k_1$ –арность $R_1$ , $k_2$ –арность $R_2$	рис. 8.11 (а,д)
Проекция	$R=\Pi_{i_1, \dots, i_k}(R_1)$ , где $i_1, i_2, \dots, i_k$ –номера столбцов $R_1$	Выборка из отношения указанных столбцов и компоновка их в указанном порядке		рис. 29(а,е)
Соединение	$R=R_1 \bowtie R_2$	Комбинация двух отношений по общим атрибутам	$R_1 \cap R_2 \neq \emptyset$	рис. 31(а)
Эквисоединение	$R=R_1 \ltimes R_2$	Расширение операции соединения для сравнения столбцов с различными именами атрибутов		рис. 31(б)
Селекция (выборка)	$R=\sigma_f(R_1)$ */	Подмножество кортежей $R_1$ , для которых значения выделенных атрибутов определены $f$		рис. 31(в)
Частное (деление)	$R=R_1 \div R_2$	Подмножество кортежей $R$ арности $(k_1-k_2)$ , таких, что для всех кортежей $p$ арности $k_2$ , принадлежащих $R_2$ ,	При $k_1$ –арность $R_1$ , $k_2$ –арность $R_2$ , $k_1 > k_2$ , $k_2 > 0$	рис. 30

		кортеж принадлежит $R_1$	$P_n$		
--	--	-----------------------------	-------	--	--

\*/ где  $f$  – формула, образованная:

- операндами, являющимися номерами столбцов;
- логическими операторами  $\square$  (и),  $\square$  (или),  $\square$  (не);
- арифметическими операторами сравнения  $<$ ,  $=$ ,  $>$ ,  $\square$ ,  $\square$ ,  $\square$ .

<p>а) <math>R_1</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>а</td><td>б</td><td>в</td></tr> <tr><td>г</td><td>д</td><td>е</td></tr> <tr><td>ж</td><td>з</td><td>и</td></tr> </table>	а	б	в	г	д	е	ж	з	и	<p>б) <math>R=R_1 \square R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>а</td><td>б</td><td>в</td></tr> <tr><td>г</td><td>д</td><td>е</td></tr> <tr><td>ж</td><td>з</td><td>и</td></tr> <tr><td>к</td><td>л</td><td>м</td></tr> </table>	а	б	в	г	д	е	ж	з	и	к	л	м	<p>в) <math>R=R_1 \setminus R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>а</td><td>б</td><td>в</td></tr> <tr><td>г</td><td>д</td><td>е</td></tr> </table>	а	б	в	г	д	е															
а	б	в																																										
г	д	е																																										
ж	з	и																																										
а	б	в																																										
г	д	е																																										
ж	з	и																																										
к	л	м																																										
а	б	в																																										
г	д	е																																										
<p><math>R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>ж</td><td>з</td><td>и</td></tr> <tr><td>к</td><td>л</td><td>м</td></tr> </table>	ж	з	и	к	л	м	<p>г) <math>R=R_1 \square R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>ж</td><td>з</td><td>и</td></tr> </table>	ж	з	и																																		
ж	з	и																																										
к	л	м																																										
ж	з	и																																										
<p>д) <math>R=R_1 R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>а</td><td>б</td><td>в</td><td>ж</td><td>з</td><td>и</td></tr> <tr><td>г</td><td>д</td><td>е</td><td>ж</td><td>з</td><td>и</td></tr> <tr><td>ж</td><td>з</td><td>и</td><td>ж</td><td>з</td><td>и</td></tr> <tr><td>а</td><td>б</td><td>в</td><td>к</td><td>л</td><td>м</td></tr> <tr><td>г</td><td>д</td><td>е</td><td>к</td><td>л</td><td>м</td></tr> <tr><td>ж</td><td>з</td><td>и</td><td>к</td><td>л</td><td>м</td></tr> </table>	а	б	в	ж	з	и	г	д	е	ж	з	и	ж	з	и	ж	з	и	а	б	в	к	л	м	г	д	е	к	л	м	ж	з	и	к	л	м	<p><math>R=\Pi_{3,1}(R_1)</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>в</td><td>а</td></tr> <tr><td>е</td><td>г</td></tr> <tr><td>и</td><td>ж</td></tr> </table>	в	а	е	г	и	ж	
а	б	в	ж	з	и																																							
г	д	е	ж	з	и																																							
ж	з	и	ж	з	и																																							
а	б	в	к	л	м																																							
г	д	е	к	л	м																																							
ж	з	и	к	л	м																																							
в	а																																											
е	г																																											
и	ж																																											

Рис. 8.11

<p><math>R_1</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>а</td><td>б</td><td>в</td><td>г</td></tr> <tr><td>а</td><td>б</td><td>д</td><td>е</td></tr> <tr><td>к</td><td>л</td><td>д</td><td>е</td></tr> <tr><td>м</td><td>н</td><td>в</td><td>г</td></tr> <tr><td>м</td><td>н</td><td>д</td><td>е</td></tr> <tr><td>а</td><td>б</td><td>д</td><td>е</td></tr> </table>	а	б	в	г	а	б	д	е	к	л	д	е	м	н	в	г	м	н	д	е	а	б	д	е	<p><math>R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>в</td><td>г</td></tr> <tr><td>д</td><td>е</td></tr> </table>	в	г	д	е	<p><math>R_1 \square R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>а</td><td>б</td></tr> <tr><td>м</td><td>н</td></tr> </table>	а	б	м	н
а	б	в	г																															
а	б	д	е																															
к	л	д	е																															
м	н	в	г																															
м	н	д	е																															
а	б	д	е																															
в	г																																	
д	е																																	
а	б																																	
м	н																																	

Рис. 8.12

<p>а) <math>R_1</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>А</td><td>В</td><td>С</td></tr> <tr><td>а</td><td>б</td><td>в</td></tr> <tr><td>г</td><td>д</td><td>е</td></tr> <tr><td>ж</td><td>з</td><td>и</td></tr> </table>	А	В	С	а	б	в	г	д	е	ж	з	и	<p><math>R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>С</td><td>Д</td></tr> <tr><td>в</td><td>м</td></tr> <tr><td>е</td><td>п</td></tr> </table>	С	Д	в	м	е	п	<p><math>R=R_1  &gt;&lt;  R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>А</td><td>В</td><td>С</td><td>С</td><td>Д</td></tr> <tr><td>а</td><td>б</td><td>в</td><td>в</td><td>м</td></tr> <tr><td>г</td><td>д</td><td>е</td><td>е</td><td>п</td></tr> </table>	А	В	С	С	Д	а	б	в	в	м	г	д	е	е	п					
А	В	С																																						
а	б	в																																						
г	д	е																																						
ж	з	и																																						
С	Д																																							
в	м																																							
е	п																																							
А	В	С	С	Д																																				
а	б	в	в	м																																				
г	д	е	е	п																																				
<p>б) <math>R_1</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>А</td><td>В</td><td>С</td></tr> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> </table>	А	В	С	1	2	3	4	5	6	7	8	9	<p><math>R_2</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>Д</td><td>Е</td></tr> <tr><td>10</td><td>15</td></tr> <tr><td>7</td><td>12</td></tr> </table>	Д	Е	10	15	7	12	<p><math>R=R_1  &gt;&lt;  R_2</math> <math>C &lt; D</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>А</td><td>В</td><td>С</td><td>Д</td><td>Е</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>10</td><td>15</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>7</td><td>12</td></tr> <tr><td>4</td><td>5</td><td>6</td><td>7</td><td>12</td></tr> </table>	А	В	С	Д	Е	1	2	3	10	15	1	2	3	7	12	4	5	6	7	12
А	В	С																																						
1	2	3																																						
4	5	6																																						
7	8	9																																						
Д	Е																																							
10	15																																							
7	12																																							
А	В	С	Д	Е																																				
1	2	3	10	15																																				
1	2	3	7	12																																				
4	5	6	7	12																																				
<p>в) <math>R_1</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>А</td><td>В</td><td>С</td></tr> <tr><td>а</td><td>б</td><td>в</td></tr> <tr><td>г</td><td>д</td><td>е</td></tr> <tr><td>ж</td><td>з</td><td>и</td></tr> </table>	А	В	С	а	б	в	г	д	е	ж	з	и	<p><math>R=\square_{A=a}(R_1)</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>А</td><td>В</td><td>С</td></tr> <tr><td>а</td><td>б</td><td>в</td></tr> </table>	А	В	С	а	б	в	<p><math>R=\square_{A=a \text{ или } A=g}(R_1)</math></p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>А</td><td>В</td><td>С</td></tr> <tr><td>а</td><td>б</td><td>в</td></tr> <tr><td>г</td><td>д</td><td>е</td></tr> </table>	А	В	С	а	б	в	г	д	е											
А	В	С																																						
а	б	в																																						
г	д	е																																						
ж	з	и																																						
А	В	С																																						
а	б	в																																						
А	В	С																																						
а	б	в																																						
г	д	е																																						

Рис. 8.13

Языки манипулирования данными, основанные на исчислении предикатов, относятся к классу декларативных языков. При декларативном подходе запрос к БД соответствует формуле некоторой формально-логической теории, а ответом является множество объектов из области интерпретации, на котором истинна формула, соответствующая запросу.



### 4.3. Лабораторные работы

Учебным планом не предусмотрено.

### 4.4. Практические занятия

<i>№ п/п</i>	<i>Номер раздела дисциплины</i>	<i>Наименование тем практических занятий</i>	<i>Объем (час.)</i>	<i>Вид занятия в интерактивной, активной, инновационной формах, (час.)</i>
1	2.	Разработка локальной базы данных под управлением СУБД Access	4	-
2	3.	Разработка приложения базы данных в интегрированной среде Builder С++ с использованием интерфейса BDE	4	-
3	4.	Разработка приложения базы данных в интегрированной среде Builder С++ с использованием стандартов ODBC	4	-
4	5.	Разработка приложения базы данных в интегрированной среде Builder С++ с использованием стандартов ADO	4	-
5	7.	Разработка многопользовательской автоматизированной информационной системы под управлением СУБД SQL Server	4	Разбор конкретных ситуаций (4 час.)
6	8.	Базовые технологии доступа к БД в Borland С++ Builder	4	-
<b>ИТОГО</b>			<b>24</b>	<b>4</b>

### 4.5. Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат.

Учебным планом не предусмотрены.

**5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

<i>№, наименование разделов дисциплины</i>	<i>Компетенции</i>	<i>Кол-во часов</i>	<i>Компетенции</i>		$\Sigma$ <i>комп.</i>	$t_{cp}$ , час	<i>Вид учебных занятий</i>	<i>Оценка результатов</i>
			<i>ПК</i>					
			<b>6</b>					
<b>1</b>	<b>2</b>	<b>3</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>		
<b>1.</b> Введение. Предмет и задачи курса		12	+		1	12	Лк, СРС	Зачёт
<b>2.</b> Управление информацией и системы баз данных		16	+		1	16	Лк, ПЗ, СРС	Зачёт
<b>3.</b> Жизненный цикл информационных систем		15	+		1	15	Лк, ПЗ, СРС	Зачёт
<b>4.</b> Технология и методология проектирования автоматизированных информационных систем		15	+		1	15	Лк, ПЗ, СРС	Зачёт
<b>5.</b> Требования к организации диалога и представлению данных		15	+		1	15	Лк, ПЗ, СРС	Зачёт
<b>6.</b> Принципы построения и организации баз данных		9	+		1	9	Лк, СРС	Зачёт
<b>7.</b> Дореляционные структуры и модели данных		13	+		1	13	Лк, ПЗ, СРС	Зачёт
<b>8.</b> Реляционная модель данных		13	+		1	13	Лк, ПЗ, СРС	Зачёт
<b>всего часов</b>		<b>108</b>		<b>108</b>	<b>1</b>	<b>108</b>		

## 6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Волкова, В. Н. Теория информационных процессов и систем [текст] : учебник и практикум для академического бакалавриата / В.Н. Волкова. - М. : Юрайт, 2014. - 502 с. - (бакалавр. академический курс). Страницы для СР: 57 - 89, 101 – 112, 137 – 156, 311 – 378, 389 – 415.

## 7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№	Наименование издания (автор, заглавие, выходные данные)	Вид занятия	Количество экземпляров в библиотеке, шт.	Обеспеченность, (экз./ чел.)
1	2	3	4	5
<b>Основная литература</b>				
1.	Белов, В.В. Проектирование информационных систем : учебник / В.В. Белов, В.И. Чистякова. - Москва : Академия, 2013. - 352 с. - (Бакалавриат)	ПЗ	15	1,0
2.	Золотов, С.Ю. Проектирование информационных систем: учебное пособие / Томск: Эль Контент, 2013. – 88 с. <a href="http://biblioclub.ru/index.php?page=book_red&amp;id=208706&amp;sr=1">http://biblioclub.ru/index.php?page=book_red&amp;id=208706&amp;sr=1</a>	Лк	ЭР	1,0
3.	Зубатов А.Ю. Информационное обеспечение процессов управления на предприятии / М.: Лаборатория книги, 2012. – 105 с. <a href="http://biblioclub.ru/index.php?page=book_red&amp;id=140252&amp;sr=1">http://biblioclub.ru/index.php?page=book_red&amp;id=140252&amp;sr=1</a>	Лк	ЭР	1,0
4.	Кузнецов, С. Д. Базы данных : учебник / С.Д. Кузнецов. - Москва : Академия, 2012. - 496 с. - (Прикладная математика и информатика)	ПЗ	15	1,0
5.	Рябов И. В. Автоматизированные информационно-управляющие системы: учебное пособие / Йошкар-Ола: ПГТУ, 2015. – 200 с. <a href="http://biblioclub.ru/index.php?page=book_red&amp;id=439330&amp;sr=1">http://biblioclub.ru/index.php?page=book_red&amp;id=439330&amp;sr=1</a>	Лк	ЭР	1,0
6.	Стружкин, Н. П. Базы данных: проектирование. Практикум [Текст] : учебное пособие для академического бакалавриата / Н.П. Стружкин, В.В. Годин. - Москва : Юрайт, 2016. - 291 с. - (Бакалавр. Академический курс).	ПЗ	10	0,7
7.	Щелоков С. А., Чернопрудова Е. Проектирование распределенных информационных систем : курс лекций по дисциплине «Проектирование распределенных информационных систем»: учебное пособие / Оренбург: Оренбургский государственный университет, 2012. 195 с. <a href="http://biblioclub.ru/index.php?page=book_red&amp;id=260753&amp;sr=1">http://biblioclub.ru/index.php?page=book_red&amp;id=260753&amp;sr=1</a>	Лк	ЭР	1,0
<b>Дополнительная литература</b>				
8.	Архангельский, А. Я. Приемы программирования в С++ Builder 6 и 2006 : учебное пособие / А.Я. Архангельский. - Москва : БИНОМ, 2006. - 992 с.	ПЗ	5	0,3
9.	Ганьшин, Д. А. Информационное обеспечение систем управления [Электронный ресурс] : конспект лекций / Д.А. Ганьшин, М.А. Крупская . - Минск : БГУИР, 2005. - 61 с.	Лк	ЭР	1,0
10.	Голенищев, Э. П. Информационное обеспечение систем	Лк	ЭР	1,0

	управления [Электронный ресурс] / Э. П. Голенищев, И.В. Клименко. - Ростов-на-Дону : Феникс, 2003. - 157 с. - (Учебники и учебные пособия) <a href="http://ecat.brstu.ru/catalog/Ресурсы%20свободного%20до%20стуга/Голенищев%20Э.П.Информационное%20обеспечение%20систем%20управления.Учеб.пособие.2003.pdf">http://ecat.brstu.ru/catalog/Ресурсы%20свободного%20до%20стуга/Голенищев%20Э.П.Информационное%20обеспечение%20систем%20управления.Учеб.пособие.2003.pdf</a>			
11.	Гудков, А. Д. Проектирование информационных систем : методические указания / А.Д. Гудков, С.В. Шаров. - Братск : БрГТУ, 2004. - 76 с.	ПЗ	87	1,0
12.	Гудков, А. Д. Проектирование информационных систем. Ч.1 : конспект лекций / А. Д. Гудков. - Братск : БрГТУ, 2001. - 62 с.	Лк	41	1,0
13.	Гудков, А. Д. Проектирование информационных систем. Ч.2 : конспект лекций / А. Д. Гудков. - Братск : БрГТУ, 2002. - 59 с.	Лк	21	1,0
14.	Заботина, Н. Н. Проектирование информационных систем. В 2 ч. Ч. 1-2 / Н.Н. Заботина. - Братск : БрГУ, 2005. Ч.1 : Методология функционального моделирования. - 146 с.	Лк	158	1,0
15.	Заботина, Н. Н. Проектирование информационных систем. В 2 ч. Ч.1-2 / Н. Н. Заботина. - Братск : БрГУ, 2005. Ч.2 : Концептуальное моделирование базы данных. - 146 с.	Лк	149	1,0
16.	Иванов, М. Ю. Информационные системы и технологии (часть 2) : методические указания к выполнению практического занятия / М. Ю. Иванов. - Братск : БрГУ, 2013. - 63 с.	ПЗ	25	1,0
17.	Иванов, М. Ю. Информационная система и организация (часть 1) : методические указания к выполнению практического занятия и контрольной работы / М. Ю. Иванов. - Братск : БрГУ, 2013. - 29 с.	ПЗ	25	1,0
18.	Корпоративные информационные системы управления : учебное пособие / Под ред. Н.М. Абдикеева. - М. : ИНФРА-М, 2011. - 464 с. + 1 эл. опт. диск (CD-ROM). - (Учебники для программы MBA).	Лк	5	0,3
19.	Петров, В. Н. Информационные системы : учебное пособие для вузов / В. Н. Петров. - Санкт-Петербург : Питер, 2003. - 688 с.	Лк	14	0,9
20.	Хомоненко, А. Д. Базы данных : учебник для вузов / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев. - 5-е изд., доп. - Москва : Бинوم-Пресс, 2006. - 736 с.	ПЗ	15	1,0

## 8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ  
[http://irbis.brstu.ru/CGI/irbis64r\\_15/cgiirbis\\_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=](http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=).
2. Электронная библиотека БрГУ  
<http://ecat.brstu.ru/catalog>.
3. Электронно-библиотечная система «Университетская библиотека online»  
<http://biblioclub.ru>.
4. Электронно-библиотечная система «Издательство «Лань»  
<http://e.lanbook.com>.
5. Информационная система "Единое окно доступа к образовательным ресурсам"  
<http://window.edu.ru>.
6. Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru>.

7. Университетская информационная система РОССИЯ (УИС РОССИЯ)  
<https://uisrussia.msu.ru/> .

8. Национальная электронная библиотека НЭБ  
<http://xn--90ax2c.xn--p1ai/how-to-search/> .

## **9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

### **9.1. Методические указания для обучающихся по выполнению практических занятий**

#### **Практическое занятие №1**

#### **Разработка локальной базы данных под управлением СУБД Access**

##### Цель работы:

Научиться создавать базы данных с помощью СУБД Access.

##### Задание:

1. Создать таблицы и определить связи между ними.

В таблицы базы данных впоследствии будет вводиться информация. В дальнейшем данные в таблице могут дополняться новыми данными, редактироваться или исключаться из таблицы. Можно просматривать данные в таблицах или упорядочивать их по некоторым признакам. Информация, содержащаяся в таблицах, может быть использована для составления отчетов. Кроме того, можно дать графическую интерпретацию информации, содержащейся в базе данных.

2. Определить необходимые в таблице поля.

Каждая таблица должна содержать информацию на отдельную тему, а каждое поле в таблице – содержать отдельные сведения по теме таблицы. При разработке полей для каждой таблицы необходимо помнить:

- а) Каждое поле должно быть связано с темой таблицы;
- б) Не рекомендуется включать в таблицу данные, которые являются результатом выражения;
- в) В таблице должна присутствовать вся необходимая информация;
- г) Информацию следует разбивать на наименьшие логические единицы.

3. Задать индивидуальные значения каждому полю.

С тем чтобы Microsoft Access мог связать данные из разных таблиц, каждая таблица должна содержать поле или набор полей, которые будут задавать индивидуальное значение каждой записи в таблице. Такое поле или набор полей называют основным ключом.

4. Определить связи между таблицами.

После распределения данных по таблицам и определения ключевых полей необходимо выбрать схему для связи данных в разных таблицах. Для этого нужно определить связи между таблицами. Желательно изучить связи между таблицами в уже существующей базе данных. Таблицы создать при помощи конструктора.

5. Разработать запросы.

6. Разработать запрос в режиме конструктора.

Для создания запроса необходимо в окне БД выбрать строку «Запросы» и выбрать на панели иконку «Создать». В открывшемся окне «Новый запрос» выбрать строку «Конструктор». Далее в открывшемся окне «Добавление таблицы» нужно выбрать используемые в запросе таблицы и нажать кнопку «Добавить». Для закрытия окна нажать кнопку «Закрыть».

7. Разработать формы с помощью конструктора.

8. Разработать формы с помощью мастера форм.

Чтобы начать создание формы, в окне БД нужно выбрать строку «Формы». Для получения доступа к списку всех вариантов создания новой формы необходимо на панели

инструментов окна БД нажать кнопку «Создать». Для вызова Мастера форм в окне «Новая форма» выбрать соответствующую строку.

В режиме формы, построенной по описанным выше правилам, выполнить поиск, сортировку, фильтрацию и корректировку данных. С помощью кнопок в нижней части формы переходить от одной записи к другой. Открыв форму конструктором, добавить кнопки поиска, удаления, добавления, сохранения.

9. Разработать отчеты.

Средства Access по разработке отчетов предназначены для конструирования макета отчета, по которому может быть осуществлен вывод данных в виде выходного печатного документа. Перед началом конструирования отчета нужно спроектировать его макет, т.е. определить состав и содержание разделов отчета, размещение в нем значений, выводимых из полей таблиц (запросов) БД, и вычисляемых реквизитов, поля, по которым необходимо группировать данные. Для каждого уровня группировки определить заголовки и примечания, вычисляемые итоговые значения. Кроме того, оформить заголовки и подписи реквизитов отчета и определить порядок вывода данных. Отчет можно создать с помощью мастера или в режиме конструктора отчетов.

10. Определить вид макета: табличный; ориентацию: книжная и поставить галочку в окне «Настроить ширину полей» для размещения на одной странице.

Порядок выполнения:

Соответствует пунктам 1 – 10 задания.

Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Цель работы;
2. Задание;
3. Поэтапное выполнение всех заданий практического занятия;
4. Выводы.

Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины.

Основная литература

1. Хомоненко, А. Д. Базы данных : учебник для вузов / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев. - 5-е изд., доп. - Москва : Бином-Пресс, 2006. - 736 с.

Дополнительная литература

2. Иванов, М. Ю. Информационная система и организация (часть 1) : методические указания к выполнению практического занятия и контрольной работы / М. Ю. Иванов. - Братск : БрГУ, 2013. - 29 с.

Контрольные вопросы для самопроверки

1. Дайте определение понятия «основной ключ».
2. Перечислите типы запросов в Access.

## **Практическое занятие №2**

**Разработка приложения базы данных в интегрированной среде Builder C++ с использованием интерфейса BDE**

Цель работы:

Изучить принципы разработки приложений для работы с БД в Borland C++ Builder. Получить навыки работы с Borland Database Engine..

Задание:

1. Повторить теоретические сведения о настольной СУБД Microsoft Access, изложенные в указаниях к выполнению практического занятия;
2. Построить схему данных (ER-диаграмму) для информационной системы «Телефонный справочник»;
3. *Телефонный справочник* содержит записи о номерах телефонов и их владельцах. Каждый человек проживает в определенном городе, по которому определяется префикс (код города) его телефонного номера. У каждого человека может быть несколько телефонных номеров. Тип номера (мобильный, рабочий, домашний) указывается в телефонном справочнике. При отображении номера в справочнике для городских телефонных номеров должен автоматически отображаться префикс;
4. По созданной схеме данных спроектировать базу данных в Microsoft Access и заполнить ее данными (несколько записей в каждую таблицу);
5. Следуя указаниям к практическому занятию, создать BDE-псевдоним для базы данных телефонного справочника. Указание. Обычно ярлык программы BDE Administrator находится по адресу: **Пуск | Настройка | Панель Управления | BDE Administrator**;
6. В новом проекте приложения Borland C++ Builder поместить на главную форму компонент TDatabase и настроить его для работы с БД телефонного справочника. Обратить внимание на свойство компонента **LoginPrompt**. Проверить, что соединение устанавливается успешно (установив в инспекторе объектов свойство **Active = true**);
7. При невозможности выполнения задания 5:
  - а. Указать в отчете к практическому занятию суть ошибки и *попытаться объяснить причину ее возникновения*;
  - б. Выполнить установку *объектов доступа к данным* (Microsoft Data Access Objects, DAO) версии 3.5:
    - Из архива **dao35.zip**, входящего в комплект практического занятия, извлечь файл **dao35.dll** и поместить его в папку “X:\Program Files\Common Files\Microsoft Shared\DAO”. Здесь и далее вместо переменной X необходимо указывать букву диска, на котором установлена рабочая операционная система;
    - Извлечь все файлы из папки **Jet** указанного архива в системную папку **System32** рабочей операционной системы;
    - С помощью служебной программы ОС Windows **regsvr32.exe** зарегистрировать библиотеку **dao35.dll**, выполнив команду:

```
regsvr32.exe "X:\Program Files\Common Files\Microsoft Shared\DAO\dao35.dll"
```
  - в. Повторить выполнение заданий 4 и 5. Если по-прежнему не удается установить соединение, то привести в отчете к практическому занятию описание возникающей ошибки и *попытаться объяснить причину ее появления*;
  - г. Повторить выполнение заданий 4 и 5 для файла db97.mdb, входящего в комплект практического занятия;
  - д. Если после выполнения всех коррекционных действий, указанных в пунктах «а»-«г», соединение с БД установить не удалось, то убедиться, что ход занятия отражен в отчете в полном объеме (включая описание всех ошибок с попыткой объяснения причин их возникновения) и сделать выводы по проделанной части работы. Задания 7-11 выполняются только в том случае, если соединение с БД установлено успешно;
7. Поместить на главную форму приложения компонент набора данных TTable и связать его с одной из таблиц телефонного справочника;
8. Поместить на главную форму компонент визуального отображения данных TDBGrid и настроить его таким образом, чтобы он занимал все пространство формы;
9. Обеспечить связь компонента TDBGrid с данными таблицы TTable;

10. Обеспечить, чтобы при запуске программы набор данных TTable находился в активном (открытом) состоянии;
11. Запустить приложение. Выполнить несколько операций перемещения курсора по записям и редактирования и добавления данных;
12. Отразить в отчете к практическому занятию полный ход ее выполнения и сформулировать выводы.

Порядок выполнения:

Соответствует пунктам 1 – 12 задания.

Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Цель работы;
2. Задание;
3. Поэтапное выполнение всех заданий практического занятия;
4. Выводы.

Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в третьем разделе данной дисциплины.

Основная литература

1. Кузнецов, С. Д. Базы данных : учебник / С.Д. Кузнецов. - Москва : Академия, 2012. - 496 с. - (Прикладная математика и информатика).

Дополнительная литература

2. Гудков, А. Д. Проектирование информационных систем : методические указания / А.Д. Гудков, С.В. Шаров. - Братск : БрГТУ, 2004. - 76 с.

Контрольные вопросы для самопроверки

1. Понятие фактографической информационной системы.
2. Назовите достоинства и недостатки технологии BDE.

**Практическое занятие №3**

**Разработка приложения базы данных в интегрированной среде Builder C++ с использованием стандартов ODBC**

Цель работы:

Изучить принципы разработки приложений для работы с БД в Borland C++ Builder. Получить навыки работы с технологией ODBC.

Задание:

1. Создать поименованный источник данных ODBC для базы данных Microsoft Access из практического занятия № 2;
2. Изучить назначение компонентов для визуального отображения данных (вкладка «Data Controls» палитры компонентов). В отчете к практическому занятию указать назначение и основные свойства для компонентов TDBCtrlGrid и TDBLookupComboBox;
3. С использованием указаний к практическому занятию написать программу в среде Borland C++ Builder, реализующую следующие функции информационной системы «Телефонный справочник»:
  - а) Отображение справочника телефонных номеров в виде таблицы или в виде набора карточек (по желанию пользователя);



- б) В справочнике указывается ФИО владельца, город проживания и телефонный номер с указанием префикса;
  - в) Реализовать выравнивание названия и содержимого столбца с именем владельца по центру. Все наименования столбцов должны быть на русском языке;
  - г) Обеспечить навигацию по записям справочника.
4. Выяснить назначение следующих свойства объектов класса **TField**: **DisplayLabel**, **DisplayWidth**, **Visible**. Изучить влияние указанных свойств на внешний вид сетки **TDBGrid**, связанной с набором данных поля. Рассмотреть 2 случая:
- д) редактор столбцов сетки используется;
  - е) редактор столбцов сетки не используется. Сделать выводы.
5. Преобразовать базу данных db97.mdb в формат Microsoft Access 2002-2003 (для этого запустить Microsoft Access и вызвать команду главного меню **Сервис | Служебные команды | Преобразовать базу данных | в формат Access 2002-2003...**). Настроить ранее созданный источник данных ODBC для работы с БД из полученного файла. Проверить работоспособность приложения. Сделать выводы;
6. Отразить в отчете к практическому занятию ход ее выполнения и сделанные выводы.

#### Порядок выполнения:

Соответствует пунктам 1 – 6 задания.

#### Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Цель работы;
2. Задание;
3. Поэтапное выполнение всех заданий практического занятия;
4. Выводы.

#### Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

#### Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в четвёртом разделе данной дисциплины.

#### Основная литература

1. Стружкин, Н. П. Базы данных: проектирование. Практикум [Текст] : учебное пособие для академического бакалавриата / Н.П. Стружкин, В.В. Годин. - Москва : Юрайт, 2016. - 291 с. - (Бакалавр. Академический курс).

#### Дополнительная литература

2. Иванов, М. Ю. Информационные системы и технологии (часть 2) : методические указания к выполнению практического занятия / М. Ю. Иванов. - Братск : БрГУ, 2013. - 63 с.

#### Контрольные вопросы для самопроверки

1. Понятие и архитектура ODBC.
2. Объясните принцип технологии «BDE+ODBC».

### **Практическое занятие №4**

**Разработка приложения базы данных в интегрированной среде Builder C++ с использованием стандартов ADO**

#### Цель работы:

Изучить принципы разработки приложений для работы с БД в Borland C++ Builder. Получить навыки работы с Borland Database Engine.

### Задание:

1. Создать новую базу данных в Microsoft Access, состоящую из единственной таблицы DUMMY с двумя полями: uid (счетчик) и Value (текст);
2. Создать поименованный источник данных ODBC для работы с новой базой данных;
3. Поместить на форму нового приложения C++ Builder пару компонентов TDatabase и TTable. Настроить параметры компонентов таким образом, чтобы через набор данных TTable можно было работать с таблицей DUMMY;
4. Поместить на форму пару компонентов TADOConnection и TADOTable. Открыть диалоговое окно настройки параметров соединения. Убедиться в наличии поставщика данных Microsoft OLE DB Provider for ODBC Drivers. Проверить наличие поставщика данных Microsoft Jet OLE DB Provide;
5. Используя указания к практическому занятию, сформировать для компонента TADOConnection строку соединения ADO с созданной ранее БД через поставщика данных Microsoft OLE DB Provider for ODBC Drivers;
6. Настроить компоненту TADOTable для работы с таблицей DUMMY;
7. Если поставщик данных Microsoft Jet OLE DB Provider имеется в списке доступных поставщиков, то поместить на форму еще одну пару компонентов TADOConnection и TADOTable и повторить шаги 6 и 7 для их настройки. Использовать для соединения с БД поставщик Microsoft Jet OLE DB Provider;
8. Измерить скорость обращения к данным БД с использованием настроенных компонентов. Для этого с каждым набором данных формы выполнить следующие операции:
  - а) Циклическое добавление в *пустую* таблицу большого количества записей (несколько десятков тысяч). Значение поля Value генерировать как строку из 100-200 символов случайным образом (дублирование строк не допускается);
  - б) Открывание заполненной таблицы и выполнение фильтрации набора данных по определенному критерию;
  - в) Открывание заполненной таблицы с упорядочением записей по возрастанию значений *неиндексированного* поля Value и выполнение фильтрации набора данных по критерию, выбранному в предыдущем пункте;
  - г) Открывание заполненной таблицы с упорядочением записей по возрастанию значений *индексированного* поля Value и выполнение фильтрации набора данных по критерию, выбранному в пункте **б**.

Замерить время выполнения каждой операции для каждого набора данных.

### Порядок выполнения:

Соответствует пунктам 1 – 8 задания.

### Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Цель работы;
2. Задание;
3. Краткое описание хода выполнения занятия.
4. Рисунок окна настройки соединения ADO со списком доступных поставщиков данных OLE DB;
5. Результаты измерений (в табличном виде) скорости выполнения операций доступа к БД с использованием различных технологий (BDE+ODBC, ADO+OLE DB+ODBC, ADO+OLE DB+JET);
6. Выводы по проделанной работе.

### Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

### Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в пятом разделе данной дисциплины.

#### Основная литература

1. Белов, В.В. Проектирование информационных систем : учебник / В.В. Белов, В.И. Чистякова. - Москва : Академия, 2013. - 352 с. - (Бакалавриат).

#### Дополнительная литература

2. Архангельский, А. Я. Приемы программирования в С++ Builder 6 и 2006 : учебное пособие / А.Я. Архангельский. - Москва : БИНОМ, 2006. - 992 с.

#### Контрольные вопросы для самопроверки

1. Опишите технологию OLE DB и её отличия от ODBC.
2. Каким образом производится настройка компонентов ADO в среде Borland C++ Builder.

#### **Практическое занятие №5**

#### **Разработка многопользовательской автоматизированной информационной системы под управлением СУБД SQL Server**

#### Цель работы:

Научиться создавать базы данных в среде SQL Server.

#### Задание:

1. Создать базу данных и ее таблицы в соответствии с вариантом. Выполнить задание с помощью стандартных команд языка T-SQL;
2. Создать диаграмму БД средствами среды SQL Server Management Studio.

#### Порядок выполнения:

Соответствует пунктам 1 – 2 задания.

#### Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Цель работы;
2. Задание;
3. Поэтапное выполнение всех заданий практического занятия;
4. Выводы.

#### Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

#### Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в седьмом разделе данной дисциплины.

#### Основная литература

1. Хомоненко, А. Д. Базы данных : учебник для вузов / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев. - 5-е изд., доп. - Москва : Бином-Пресс, 2006. - 736 с.

#### Дополнительная литература

2. Иванов, М. Ю. Информационная система и организация (часть 1) : методические указания к выполнению практического занятия и контрольной работы / М. Ю. Иванов. - Братск : БрГУ, 2013. - 29 с.

#### Контрольные вопросы для самопроверки

1. Приведите синтаксис SQL-команды создания нового файла.
2. Какие действия можно выполнить в языке запросов T-SQL с БД?

## **Практическое занятие №6** **Базовые технологии доступа к БД в Borland C++ Builder**

### Цель работы:

Изучить основные способы работы с наборами данных в среде Borland C++ Builder. Получить навыки проектирования несложных фактографических систем.

### Задание:

1. В соответствии с вариантом задания спроектировать и реализовать простейшую фактографическую информационную систему;
2. При проектировании БД рекомендуется использовать результаты выполнения практических занятий 4 и 5;
3. При реализации программного приложения для работы с БД выполнить следующие требования:
  - а) Разработку программного приложения начать с проектирования пользовательского интерфейса и подготовки прототипа приложения;
  - б) Развитие проекта программного приложения осуществлять в соответствии с принципами итеративной разработки;
  - в) Обеспечить возможность добавления, редактирования и удаления записей во все основные таблицы БД;
  - г) Обязательно использовать компонент TDBLookupComboBox для редактирования и/или определения условий поиска информации;
  - д) Обязательно использовать поля синхронного просмотра и вычисляемые поля;
  - е) Обязательно использовать один из способов упорядочения набора данных;
  - ж) Обязательно использовать один из способов поиска информации в наборе данных.

### Порядок выполнения:

Соответствует пунктам 1 – 3 задания.

### Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Цель работы;
2. Задание;
3. Поэтапное выполнение всех заданий практического занятия;
4. Модель данных (ER-диаграмма) ИС;
5. Перечень и содержание выполненных итераций по разработке программного приложения;
6. Результаты тестирования программы;
7. Выводы по проделанной работе.

### Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

### Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в восьмом разделе данной дисциплины.

#### Основная литература

1. Стружкин, Н. П. Базы данных: проектирование. Практикум [Текст] : учебное пособие для академического бакалавриата / Н.П. Стружкин, В.В. Годин. - Москва : Юрайт, 2016. - 291 с. - (Бакалавр. Академический курс).

#### Дополнительная литература

2. Иванов, М. Ю. Информационные системы и технологии (часть 2) : методические указания к выполнению практического занятия / М. Ю. Иванов. - Братск : БрГУ, 2013. - 63 с.

### Контрольные вопросы для самопроверки

1. Опишите класс **TDataSet** и производные компоненты наборов данных.
2. Каким образом осуществляется навигация по набору данных.

### **10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

1. ОС Windows 7 Professional.
2. Microsoft Office 2007 Russian Academic OPEN No Level.
3. Антивирусное программное обеспечение Kaspersky Security.

### **11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

<i>Вид занятия</i>	<i>Наименование аудитории</i>	<i>Перечень основного оборудования</i>	<i>№ ПЗ</i>
<b>1</b>	<b>3</b>	<b>4</b>	<b>5</b>
ПЗ	Дисплейный класс	AMD Athlon 64 (5GHz/250Gb/2Gb/DD-RW), 2 ядра	ПЗ 1-6
СР	ЧЗ №3	Оборудование 15- CPU 5000/RAM 2Gb/HDD (Монитор TFT 19 LG 1953S-SF);принтер HP LaserJet P3005	-

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ  
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

**1. Описание фонда оценочных средств (паспорт)**

№ компетенции	Элемент компетенции	Раздел	Тема	ФОС
ПК-6	способность производить расчёты и проектирование отдельных блоков и устройств систем автоматизации и управления и выбирать стандартные средства автоматизации, измерительной и вычислительной техники для проектирования систем автоматизации и управления	1. Введение. Предмет и задачи курса	1.1. Введение	-
			1.2. Основные понятия курса	Вопросы к зачёту 1.1, 1.2
			1.3. Основные процессы преобразования информации и их характеристика	Вопрос к зачёту 1.3
			1.4. Системы информационного обмена	Вопросы к зачёту 1.4, 1.5
		2. Управление информацией и системы баз данных	2.1. Задачи и функции ИС	Вопрос к зачёту 2.1
			2.2. Типология ИС	Вопрос к зачёту 2.2
			2.3. Состав и структура ИС	Вопрос к зачёту 2.3
			2.4. АИПС, банки и базы данных	Вопрос к зачёту 2.4
			2.5. Информационные сети	Вопрос к зачёту 2.5
			2.6. Обеспечивающие подсистемы ИС и их характеристика	Вопросы к зачёту 2.6-2.8
		3. Жизненный цикл информационных систем	3.1. Модели жизненного цикла (ЖЦ) ИС	Вопрос к зачёту 3.1
			3.2. Стандарты на проектирование ИС	Вопрос к зачёту 3.2
	3.3. Процессы ЖЦ ПО		Вопрос к зачёту 3.3	
	3.4. Каноническое проектирование ИС		Вопрос к зачёту 3.4	
	3.5. Содержание технического задания на ИС		Вопрос к зачёту 3.5	
	3.6. Содержание технического проекта ИС		Вопрос к зачёту 3.6	
	4. Технология и методология проектирования автоматизированных информационных систем	4.1. Специфика информационных программных систем	Вопрос к зачёту 4.1	
		4.2. Трудности, встречающиеся при проектировании ИС	Вопрос к зачёту 4.2	
		4.3. Критерии	Вопрос к зачёту	

			создания ИС	4.3
		5. Требования к организации диалога и представлению данных	5.1. Стратегии построения ИС	Вопрос к зачёту 5.1
			5.2. Методологии и технологии проектирования ИС	Вопрос к зачёту 5.2
			5.3. Структурное представление ИС при их создании	Вопросы к зачёту 5.3-5.5
		6. Принципы построения и организации баз данных	6.1. Численные и информационные прикладные системы	Вопрос к зачёту 6.1
			6.2. Файловые системы	Вопросы к зачёту 6.2, 6.3
			6.3. Области применения файлов	Вопрос к зачёту 6.4
			6.4. Потребности информационных систем	Вопрос к зачёту 6.5
			6.5. СУБД и представление данных в базах данных	Вопросы к зачёту 6.6-6.9
		7. Дореляционные структуры и модели данных	7.1. Различные представления о данных в базах данных	Вопросы к зачёту 7.1, 7.2
			7.2. Основные этапы проектирования базы данных	Вопросы к зачёту 7.3, 7.4
		8. Реляционная модель данных	8.1. Модели данных	Вопрос к зачёту 8.1
			8.2. Типы моделей данных	Вопросы к зачёту 8.2-8.4

## 2. Вопросы к зачёту

№ п/п	Компетенции		ВОПРОСЫ К ЗАЧЁТУ	№ и наименование раздела
	Код	Определение		
1	2	3	4	5
1.	ПК-6	способность производить расчёты и проектирование отдельных блоков и устройств систем автоматизации и управления и выбирать стандартные средства автоматики, измерительной и вычислительной техники для проектирования систем автоматизации и управления	1. Основные понятия курса	1. Введение. Предмет и задачи курса
			2. Схема пересечения тезаурусов	
			3. Основные процессы преобразования информации и их характеристика	
			4. Системы информационного обмена	
			5. Информационная структура СИОБ	
			1. Задачи и функции ИС	2. Управление информацией и системы баз данных
			2. Типология ИС	
			3. Состав и структура ИС	
			4. АИПС, банки и базы данных	
			5. Информационные сети	
6. Информационное обеспечение ИС				
7. Техническое, математическое и программное обеспечение ИС				

			<b>8. Организационное и правовое обеспечение ИС</b>	
			<b>1. Модели жизненного цикла ИС</b>	<b>3. Жизненный цикл информационных систем</b>
			<b>2. Стандарты на проектирование ИС</b>	
			<b>3. Процессы жизненного цикла программного обеспечения</b>	
			<b>4. Каноническое проектирование ИС</b>	
			<b>5. Содержание технического задания на ИС</b>	
			<b>6. Содержание технического проекта ИС</b>	
			<b>1. Специфика информационных программных систем</b>	<b>4. Технология и методология проектирования автоматизированных информационных систем</b>
			<b>2. Трудности, встречающиеся при проектировании ИС</b>	
			<b>3. Критерии создания ИС</b>	
			<b>1. Стратегии построения ИС</b>	<b>5. Требования к организации диалога и представлению данных</b>
			<b>2. Методологии и технологии проектирования ИС</b>	
			<b>3. Структурный анализ</b>	
			<b>4. Структурное проектирование</b>	
			<b>5. Внедрение ИС</b>	
			<b>1. Численные и информационные прикладные системы</b>	<b>6. Принципы построения и организации баз данных</b>
			<b>2. Структуры файлов</b>	
			<b>3. Именованые файлов</b>	
			<b>4. Защита файлов</b>	
			<b>5. Режим многопользовательского доступа</b>	
			<b>6. Области применения файлов</b>	
			<b>7. Потребности ИС</b>	
			<b>8. Непосредственное управление данными во внешней памяти. Управление буферами оперативной памяти. Управление транзакциями</b>	
			<b>9. Журнализация</b>	
			<b>1. Обобщение представления пользователей о данных</b>	<b>7. Дореляционные структуры и модели данных</b>
			<b>2. Различные представления о данных в БД</b>	
			<b>3. Общая схема проектирования БД</b>	
			<b>4. Этапы проектирования БД</b>	
			<b>1. Модели данных</b>	<b>8. Реляционная модель данных</b>
			<b>2. Иерархическая модель данных</b>	
			<b>3. Сетевая модель данных</b>	
			<b>4. Реляционная модель данных</b>	



### 3. Описание показателей и критериев оценивания компетенций

Показатели	Оценка	Критерии
<p><b>Знать</b> (ПК-6): – принципы проектирования и разработки информационного обеспечения систем управления;</p> <p><b>Уметь</b> (ПК-6): - выбирать стандартные средства автоматизации, измерительной и вычислительной техники для проектирования систем автоматизации и управления;</p> <p><b>Владеть</b> (ПК-6): - навыками расчётов и проектирования отдельных блоков и устройств систем автоматизации и управления.</p>	<p><b>зачтёно</b></p>	<p>Обучающийся должен во время ответа показать знания: назначения и возможностей современных средств проектирования информационных систем, принципов проектирования и разработки информационного обеспечения систем управления, структур хранения данных и методов доступа к ним, основных терминов, используемых в научно-технической литературе по теории информационных систем. Обучающийся должен иметь навыки владения: проектирования баз данных и строящихся на их основе прикладных программных продуктов, автоматизирующих деятельность технического процесса, понимания материала и способности высказывания мыслей на научно-техническом языке. Обучающийся во время ответа должен продемонстрировать умения: осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий, пользоваться распространёнными средствами для проектирования информационных систем, разрабатывать многопользовательские приложения на современных СУБД.</p>
	<p><b>незачтено</b></p>	<p>На оба вопроса обучающийся отвечает неубедительно. На дополнительные вопросы преподавателя также не может ответить.</p>

### 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Дисциплина «Информационное обеспечение систем управления» направлена на ознакомление обучающихся с базовыми принципами создания прикладных программных средств и информационного обеспечения систем управления в целом, а также формирование практических навыков по проектированию базы данных как основы информационного обеспечения систем управления.

Изучение дисциплины предусматривает:

- лекции;
- практические занятия;
- самостоятельную работу;
- зачёт.

В ходе освоения раздела 1 «Введение. Предмет и задачи курса» обучающиеся должны

уяснить: понятия информации, информационной деятельности, процессы преобразования информации и системы информационного обмена.

В ходе освоения раздела 2 «Управление информацией и системы баз данных» обучающиеся должны знать: состав и структуру ИС, обеспечивающие подсистемы ИС.

В ходе освоения раздела 3 «Жизненный цикл информационных систем» обучающиеся должны уяснить: 3 модели жизненного цикла ИС, стандарты на проектирование ИС.

В ходе освоения раздела 4 «Технологии и методология проектирования автоматизированных информационных систем» обучающиеся должны знать: специфику информационных программных систем, трудности, встречающиеся при проектировании ИС, критерии создания ИС.

В ходе освоения раздела 5 «Требования к организации диалога и представлению данных» обучающиеся должны уяснить: стратегии построения ИС, методологии и технологии проектирования ИС, структурное проектирование и внедрение ИС.

В ходе освоения раздела 6 «Принципы построения и организации баз данных» обучающиеся должны знать: численные и информационные прикладные системы, структуры файлов, системы управления БД.

В ходе освоения раздела 7 «Дореляционные структуры и модели данных» обучающиеся должны обратить внимание на: различные представления данных в БД, общую схему и этапы проектирования БД.

В ходе освоения раздела 8 «Реляционная модель данных» обучающиеся должны знать: три типа моделей данных.

На практических занятиях обучающиеся приобретают навыки проектирования БД с использованием СУБД Access 2000 и интегрированной среды разработки Builder C++.

При подготовке к зачёту рекомендуется особое внимание уделить следующим вопросам: состав и структура ИС, три типа моделей жизненного цикла ИС, стандарты на проектирование ИС.

Работа с литературой является важнейшим элементом в получении знаний по дисциплине. Прежде всего, необходимо воспользоваться списком рекомендуемой литературы. Дополнительные сведения по изучаемым темам можно найти в периодической печати и Интернете.

Предусмотрено проведение аудиторных занятий в интерактивной форме (лекции с текущим контролем, практические занятия с разбором конкретных ситуаций) в сочетании с внеаудиторной работой.

**АННОТАЦИЯ**  
**рабочей программы дисциплины**  
**Информационное обеспечение систем управления**

**1. Цель и задачи дисциплины**

Целью изучения дисциплины является изложение базовых принципов создания прикладных программных средств и информационного обеспечения систем управления в целом, а также формирование практических навыков по проектированию базы данных как основы информационного обеспечения систем управления.

Задачей изучения дисциплины является подготовка обучающихся к самостоятельной работе по решению практических задач, связанных с построением и функционированием информационного обеспечения систем управления (ИОСУ), с использованием методов и средств создания проектов автоматизированных систем управления технологическими процессами с применением современных программных средств, с проектированием моделей ИОСУ реальных объектов регулирования.

**2. Структура дисциплины**

2.1 Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: Лк – 24 час.; ПЗ – 24 час.; СР – 60 час.

Общая трудоемкость дисциплины составляет 108 часов, 3 зачетные единицы.

2.2 Основные разделы дисциплины:

- 1 – Введение. Предмет и задачи курса;
- 2 – Управление информацией и системы баз данных;
- 3 – Жизненный цикл информационных систем;
- 4 – Технология и методология проектирования автоматизированных информационных систем;
- 5 – Требования к организации диалога и представлению данных;
- 6 – Принципы построения и организации баз данных;
- 7 – Дореляционные структуры и модели данных;
- 8 – Реляционная модель данных.

**3. Планируемые результаты обучения (перечень компетенций)**

Процесс изучения дисциплины направлен на формирование следующих компетенций:  
ПК-6 – способность производить расчёты и проектирование отдельных блоков и устройств систем автоматизации и управления и выбирать стандартные средства автоматизации, измерительной и вычислительной техники для проектирования систем автоматизации и управления.

**4. Вид промежуточной аттестации:** зачёт.

*Протокол о дополнениях и изменениях в рабочей программе  
на 201\_\_\_ - 201\_\_\_ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

---

---

2. В рабочую программу по дисциплине вносятся следующие изменения:

---

---

Протокол заседания кафедры № \_\_\_\_\_ от «\_\_\_» \_\_\_\_\_ 201\_\_\_ г.,  
*(разработчик)*

Заведующий кафедрой \_\_\_\_\_

*(подпись)*

\_\_\_\_\_

*(Ф.И.О.)*

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 27.03.04 Управление в технических системах от «20» октября 2015 г. № 1171

для набора 2018 года и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «12» марта 2018 г. № 130, заочной формы обучения от «12» марта 2018 г. № 130.

**Программу составил:**

Крумин О.К., доцент кафедры УТС, к.т.н. \_\_\_\_\_

Рабочая программа рассмотрена и утверждена на заседании кафедры УТС  
от 28 декабря 2018 г, протокол № 6

Заведующий кафедрой УТС \_\_\_\_\_

Игнатъев И.В.

**СОГЛАСОВАНО:**

Заведующий выпускающей кафедрой \_\_\_\_\_

Игнатъев И.В.

Директор библиотеки \_\_\_\_\_

Сотник Т.Ф.

Рабочая программа одобрена методической комиссией ФЭиА факультета  
от 28 декабря 2018 г, протокол № 5

Председатель методической комиссии факультета \_\_\_\_\_

Ульянов А.Д.

**СОГЛАСОВАНО:**

Начальник  
учебно-методического управления \_\_\_\_\_

Нежевец Г.П.

Регистрационный № \_\_\_\_\_