

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«Тверской государственный технический университет»**  
(ТвГТУ)

**Б.В. Палюх, А.Л. Борисов**

## **ОСНОВЫ ПОСТРОЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ**

Учебное пособие

Тверь 2019

УДК 681.3  
ББК 32.973

Рецензенты: декан факультета прикладной математики и кибернетики Тверского государственного университета, доктор физико-математических наук, профессор Язенин А.В.; профессор кафедры электронно-вычислительных машин Тверского государственного технического университета, доктор технических наук, профессор Матвеев Ю.Н.

Палюх, Б.В. Основы построения информационных систем: учебное пособие / Б.В. Палюх, А.Л. Борисов. Тверь: Тверской государственный технический университет, 2018. 136 с.

Рассматриваются теоретические и практические вопросы построения автоматизированных информационных систем на промышленных предприятиях и в административных учреждениях. Содержит основные понятия и общее математическое описание информационных систем, основные свойства единиц информации, модели данных и операции над отношениями, организацию данных в памяти ЭВМ, способы и методы разработки информационных систем на этапах предпроектного обследования объектов и проектирования функциональных и обеспечивающих подсистем.

Предназначено для студентов и магистрантов, изучающих вопросы разработки, построения и создания автоматизированных информационных систем в промышленности и административном управлении.

**Борис Васильевич Палюх**  
**Андрей Львович Борисов**

## **ОСНОВЫ ПОСТРОЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ**

Учебное пособие

Редактор Ю.А. Якушева  
Корректор  
Технический редактор Ю.Ф. Воробьева

---

Подписано в печать  
Формат 60 x 84/16  
Физ. печ. л.  
Тираж 100 экз.

Усл. печ. л.  
Заказ №

Бумага писчая  
Уч.-изд. л.  
С –

---

Редакционно-издательский центр  
Тверского государственного технического университета  
170026, г. Тверь, наб. А. Никитина, 22

ISBN

© Тверской государственный  
технический университет, 2019  
© Палюх Б.В., Борисов А.Л., 2019

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>СПИСОК СОКРАЩЕНИЙ.....</b>	<b>4</b>
<b>ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ, КЛАССИФИКАЦИЯ И ОБЩЕЕ ОПИСАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ.....</b>	<b>5</b>
1.1. Предмет и задачи курса .....	5
1.2. Основные определения и понятия, используемые при построении информационных систем.....	8
Классификация и особенности подсистем автоматизированных информационных систем .....	8
1.3. Основные принципы построения информационных систем.....	13
1.4. Общая задача управления современным промышленным предприятием.....	14
1.5. Синтез структуры автоматизированной информационной системы и ее основные характеристики .....	18
1.6. Синтез функциональной структуры информационной системы. Матричный метод декомпозиции на основе матрицы связанности документов.....	19
<b>ГЛАВА 2. СТРУКТУРА ИНФОРМАЦИОННОЙ БАЗЫ И ОСНОВНЫЕ СВОЙСТВА ЕДИНИЦ ИНФОРМАЦИИ.....</b>	<b>22</b>
2.1. Информационное обеспечение автоматизированных информационных систем.....	22
2.2. Организация информационной базы.....	24
2.3. Классификация и кодирование технико-экономической информации.....	26
2.4. Методы кодирования технико-экономической информации.....	27
2.5. Методы контроля ошибок в кодовом обозначении.....	31
Метод контроля ошибок по модулю 11 .....	31
2.6. Классификация и основные свойства единиц информации.....	32
Определение производственно-экономического показателя .....	34
2.7. Основные операции, производимые с составной единицей информации.....	35
Задания для самостоятельной работы при изучении главы 2.....	38
<b>ГЛАВА 3. МОДЕЛИ ДАННЫХ И ОПЕРАЦИИ НАД ОТНОШЕНИЯМИ.....</b>	<b>48</b>
3.1. Реляционная модель данных.....	48
3.2. Процедурное описание процессов обработки данных.....	50
3.3. Нормализация отношений.....	53
3.3.1. Функциональная зависимость и ключи.....	54
3.3.2. Вторая и третья нормальные формы отношения.....	57
3.4. Сетевая модель данных, примеры.....	58
3.4.1. Двухуровневые сетевые модели данных.....	59
3.4.2. Способ поиска информации в сетевой модели данных.....	61

3.5. Иерархическая модель данных.....	62
3.6. Преимущества и недостатки различных моделей данных.....	61
3.7. База знаний, модели знаний.....	64
Задания для самостоятельной работы при изучении главы 3.....	70
<b>ГЛАВА 4. ОРГАНИЗАЦИЯ ДАННЫХ</b>	
<b>В ПАМЯТИ ЭВМ.....</b>	<b>82</b>
4.1. Линейная организация данных в памяти ЭВМ.....	82
Сортировка данных.....	83
4.2. Методы поиска и корректировки при линейной организации данных.....	83
4.2.1. Методы поиска: определение, виды, сравнение.....	83
4.2.2. Методы корректировки данных.....	84
4.3. Списковая (цепная) организация данных.....	85
4.4. Древовидная организация данных.....	88
4.5. Методы ускорения доступа к данным.....	91
Вопросы для самопроверки к главе 4.....	95
Задания для самостоятельной работы при изучении главы 4.....	96
<b>ГЛАВА 5. СПОСОБЫ И МЕТОДЫ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ НА ЭТАПЕ ПРОЕКТИРОВАНИЯ.....</b>	<b>98</b>
5.1. Последовательность разработки информационных систем.....	98
5.2. Обследование существующей информационной системы на предприятии.....	100
5.3. Параметризация информационных систем.....	105
5.4. Параметризация потоков данных и запросов.....	106
5.5. Параметризация вычислительной системы.....	109
Определение количества компьютеров или каналов связи.....	109
5.6. Защита данных в информационных системах.....	110
5.6.1. Сохранение целостности данных.....	110
5.6.2. Методы управления доступом.....	111
<b>ГЛАВА 6. ПРИМЕРЫ МАТЕМАТИЧЕСКОГО И АЛГОРИТМИЧЕСКОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ ДЛЯ МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ.....</b>	<b>113</b>
6.1. Математическое обеспечение задачи «Прогнозирование».....	113
6.2. Математическое обеспечение задачи «Управление запасами».....	118
6.3. Математическое обеспечение задачи «Выбор порядка обработки деталей на двух станках.....	120
6.4. Пример подсистемы оперативного управления основным производством машиностроительного предприятия.....	121
<b>ПРИЛОЖЕНИЕ.....</b>	<b>125</b>
<b>БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....</b>	<b>135</b>

## ВВЕДЕНИЕ

Для современных промышленных предприятий и административных учреждений необходимо обеспечить эффективную информационную поддержку процессов принятия решений путем применения автоматизированных информационных систем.

Дисциплина «Основы построения информационных систем» является одним из основополагающих курсов по направлениям подготовки «Информационные системы и технологии» и «Прикладная информатика», соответствующие профессиональному стандарту «Системный аналитик». Этот курс дает студентам необходимые знания, которые способствуют успешному изучению дисциплин, относящихся к организации компьютерной обработки производственно-экономических данных, проектированию функциональных и обеспечивающих подсистем автоматизированных информационных систем, внедрению информационных технологий на промышленных производствах и в административных системах.

В рамках курса рассматриваются:

структуры производственно-экономической информации и функционального состава задач автоматизированной информационной системы;

построение моделей данных и способы организация данных в памяти ЭВМ;

моделирование процессов взаимосвязи основных компонент функциональных и обеспечивающих подсистем информационных систем;

средства и методы информационных технологий для решения задач принятия решений при управлении производственно-экономическим объектом.

Учебное пособие состоит из шести глав, приложения и библиографического списка.

В первой главе проводится классификация производственно-экономической информации, даются основные определения и понятия, принципы построения информационных систем, изучаются методы синтеза структуры автоматизированной информационной системы и показатели, которые ее характеризуют.

Вторая глава посвящена информационному обеспечению автоматизированных информационных систем. На примерах изучаются вопросы классификации и кодирования технико-экономической информации. Представлены базовые знания по данной дисциплине на основе классификации и основных свойств единиц информации.

В третьей главе описаны модели данных, которые используются в базах данных, характеристики реляционной, сетевой и иерархической моделей данных. Разобраны преимущества и недостатки различных моделей данных.

В четвертой главе приведены методы организации данных в памяти ЭВМ, методы ускорения доступа к данным. Проведено сравнение различных способов организации данных.

Пятая глава посвящена вопросам последовательности разработки информационных систем. Изучены задачи параметризации информационных систем, потоков данных и запросов, вычислительной системы.

В шестой главе рассматриваются стадии создания автоматизированных информационных систем, вопросы обследования существующей системы управления предприятием, построение схем документооборота, способы параметризации отдельных компонент информационной системы [10], приведены примеры математического и алгоритмического обеспечения информационных систем для машиностроительного предприятия. Дан пример структуры интегрированного пакета прикладных программ для машиностроительного предприятия.

Целью изучения дисциплины «Основы построения производственно-экономических и административных информационных систем» является повышение компетентности по вопросам приобретения знаний в области теоретических и прикладных основ построения информационных систем автоматизированного управления административными и производственно-экономическими объектами, состава производственно-экономической информации, структурной организации данных в памяти ЭВМ, методов и средств описания информационной системы и ее элементов, методов формализованного преобразования данных с использованием соответствующих моделей данных.

Для успешного изучения курса обучающийся должен владеть знаниями из области специальных разделов высшей математики (о теории множеств, математической логике, теории графов и др.), вычислительных машин, комплексов и систем передачи данных, алгоритмических языков и программирования, математических методов оптимизации. В результате изучения дисциплины студент приобретает знания, необходимые для освоения специальных дисциплин по анализу, разработке и проектированию автоматизированных информационных систем, и навыки анализа и разработки этих систем, выполнения дипломного проекта.

## СПИСОК СОКРАЩЕНИЙ

- АРМ – автоматизированные рабочие места  
БД – база данных  
БЗ – база знаний  
ГИ – готовые изделия  
ГП – готовая продукция  
КУ – комплектующие узлы.  
ИБ – информационная база.  
ИО – информационное обеспечение  
МО – математическое обеспечение  
ОЗ – открытые заказы, запущенные в производство и обеспеченные всеми ресурсами  
ОТ и З – отдел труда и заработной платы  
ПДО – планово-договорной отдел  
ПЗ – плановые заказы  
ПМ – планирование мощности  
ПО – программное обеспечение  
ПП – планирование потребностей  
ПТН – массив пооперационных трудовых нормативов (время обработки деталей на каждом станке по операциям)  
ПЭО – планово-экономический отдел  
РМ – массив рабочих мест  
СЕИ – составная единица информации  
СИ – массив состава изделий, в котором указываются входящие в изделие детали  
СПЗ – массив скорректированных плановых заказов с учетом даты обработки  
СУБД – система управления базами данных  
ТЗ – техническое задание  
ТО – техническое обеспечение  
ТЭО – технико-экономическое обоснование  
УЗ – управление запасами  
УЦ – управление цехом  
Ч/п – чистые потребности  
ЭС – экспертная система

# **ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ, КЛАССИФИКАЦИЯ И ОБЩЕЕ ОПИСАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

Общее понятие информации можно охарактеризовать как осознанные сведения об окружающем мире, которые являются объектом хранения, преобразования, передачи и использования. Элементарные свойства информации, принципы построения информационных систем и методы синтеза их структур представлены в данной главе.

## **1.1. Предмет и задача курса**

Основы построения производственно-экономических и административных информационных систем – это дисциплина, которая изучает состав, структуру и основные фундаментальные свойства автоматизированных производственно-экономических и административных информационных систем и элементов, из которых они состоят.

Предметом (объектом) изучения является административная, экономическая и производственная информация, которая необходима при планировании, учете и анализе финансово-экономической и производственной деятельности предприятия или административного учреждения.

Актуальность применения автоматизированных ИС определяется задачами, стоящими перед современным постиндустриальным информационным обществом, например цифровизацией во всех областях деятельности общества.

Можно выделить ряд особенностей, характерных для современного научно-технического прогресса на промышленных предприятиях:

- увеличивается сложность объектов управления;
- возрастают темпы роста объема выпускаемой продукции;
- быстро изменяется номенклатура выпускаемой продукции;
- с течением времени возрастают объемы потоков производственно-экономической информации;
- уменьшается допустимое время на обработку информации и принятие решений.

Соответственно, изменилось представление о повышении эффективности функционирования предприятия. Увеличить эффективность функционирования можно, не только усовершенствовав технологию производства, но и создав обратную связь. Информация тогда будет поступать с выхода объекта (ГП) на его вход (С, Э, Т), что не предусматривалось ранее (рис. 1.1).



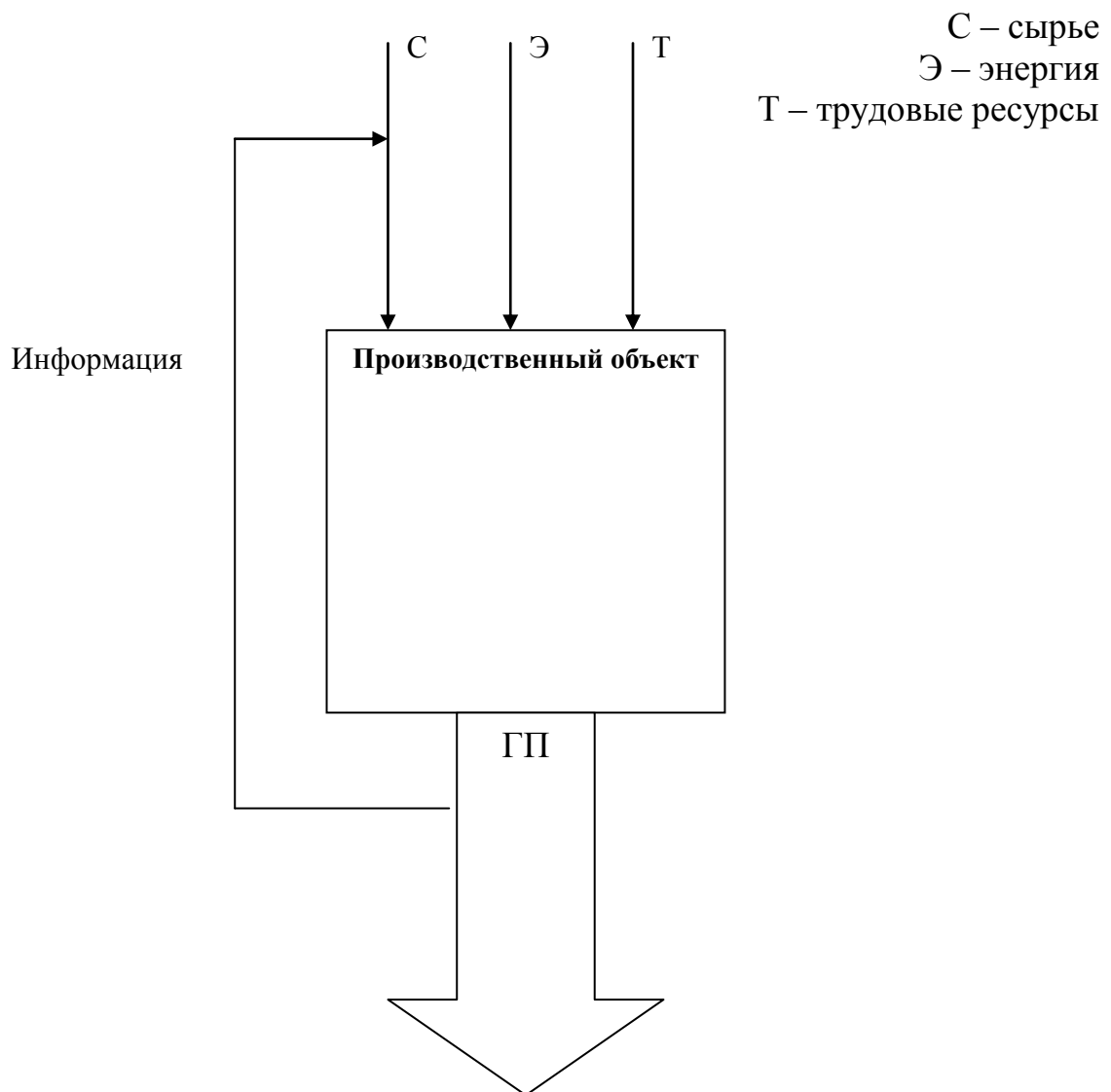


Рис. 1.1

Усложнение объекта приводит к увеличению количества обрабатываемой информации в квадрате по отношению к приросту количества элементов объекта. Приращение информации пропорционально не только приращению числа объектов  $\Delta k$ , но и общему количеству объектов  $k$ .

Прирост объема информации, возникающий при увеличении количества элементов в производстве, высчитывается по формуле  $\Delta V = \mu \cdot k \cdot \Delta k$ , где  $k$  – количество элементов, из которых состоит объект;  $\Delta k$  – прирост количества элементов объекта;  $\mu$  – коэффициент пропорциональности.

$$\int dV = \int \mu \cdot k dk \quad V = \mu \cdot k^2$$

Классификация производственно-экономической информации и функции управления предприятием представлены на рис. 1.2.

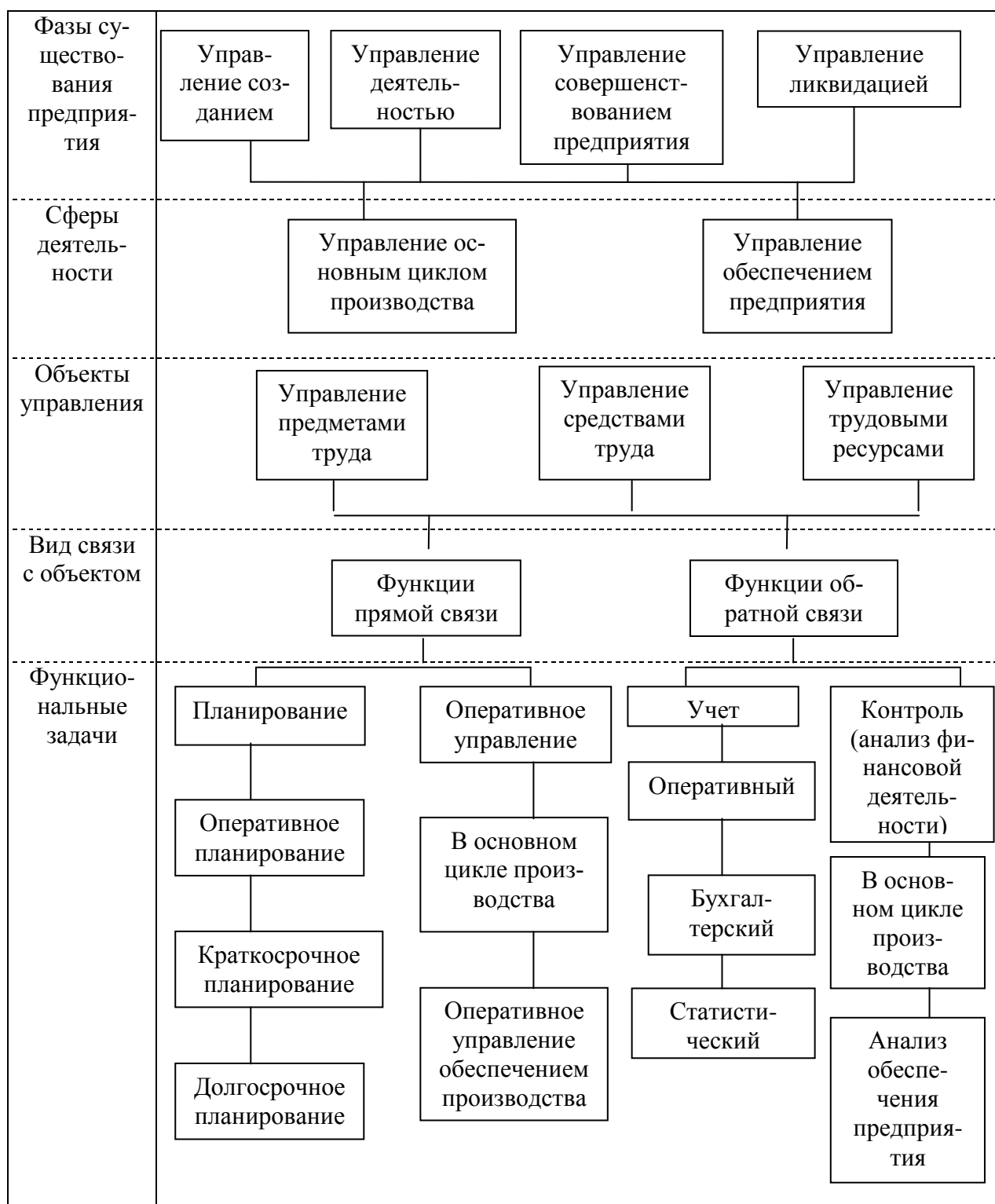


Рис. 1.2

## **1.2. Основные определения и понятия, используемые при построении информационных систем**

Информационные системы относятся к классу автоматизированных систем управления.

Автоматизированные системы управления – это системы, в которых управление осуществляется с помощью технических средств управления и человека-оператора, который является звеном в автоматизированной информационной системе. Автоматизированная система управления отличается от автоматической тем, что при автоматическом управлении человек не принимает прямого участия в оперативном управлении, объект управления менее масштабный, поэтому автоматическая система управления в основном используется для локальных технологических процессов и агрегатов.

Информационная система – это автоматизированная человеко-машинная система, предназначенная для сбора, отображения, обработки и хранения производственно-экономической информации, предназначенной для принятия управленческих решений в различных сферах человеческой деятельности [1].

Производственно-экономическая информация – новые актуальные сведения о производственно-экономическом объекте; классифицируется следующим образом:

нормативно-справочная информация (нормы, стандарты, ценники), долгосрочного хранения (изменяется значительно реже прочих видов);

плановая (является ориентиром для производственно-экономической деятельности предприятия и может быть краткосрочной и долгосрочной);

управляющая (об оперативных и периодических управляющих воздействиях на объект);

учетная (оперативная, бухгалтерская и статистическая).

Информационная система сложна, поэтому при проектировании ее разбивают на отдельные части – подсистемы.

### ***Классификация и особенности подсистем автоматизированных информационных систем***

Подсистема – это часть информационной системы, которая выделена по структурному или функциональному признаку и отвечает конкретным целям и задачам. Примеры структурного признака – бухгалтерия, отдел кадров, отдел сбыта и т. д. Функциональный признак – это функции, которые выполняются на производственно-экономическом или административном объекте.

Информационная система (ИС) подразделяется на две части: функциональную и обеспечивающую (рис. 1.3).

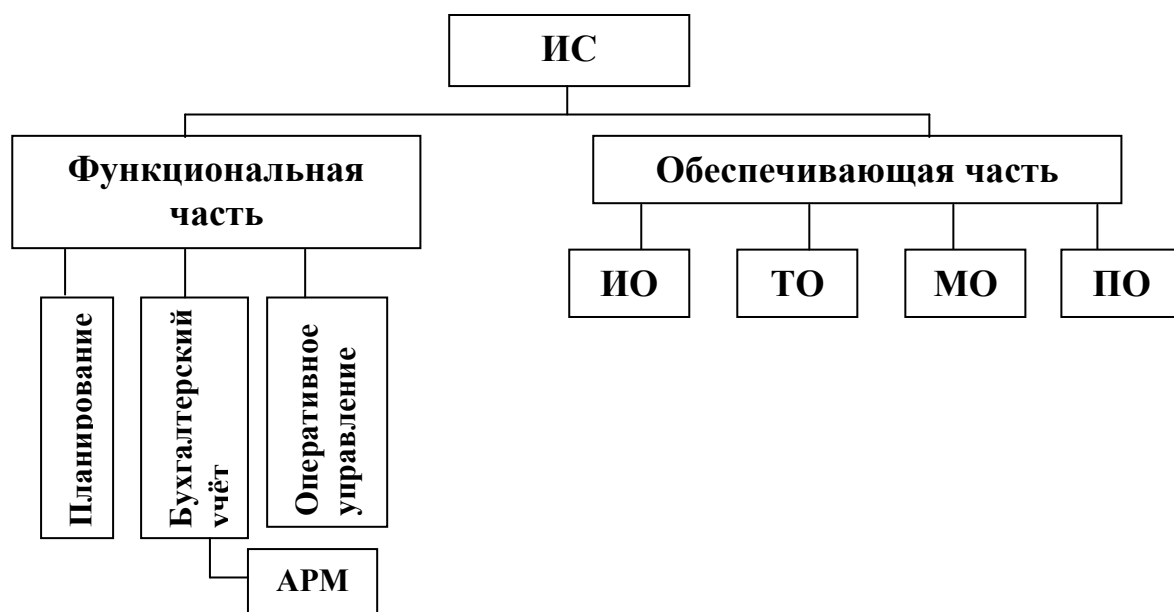


Рис. 1.3

Функциональной частью называется комплекс административных, организационных и экономико-математических методов, предназначенных для решения задач планирования, учета и анализа производственно-экономических показателей для принятия управленческих решений. Подсистемы, входящие в функциональную часть информационной системы, называются функциональными. Примеры этих подсистем – подсистемы планирования, бухгалтерского учета, оперативного управления.

Обеспечивающая часть – это комплекс ресурсов, которые обеспечивают решение производственно-экономических задач функциональных подсистем. Каждый ресурс оформляют в виде функциональной подсистемы. К обеспечивающим подсистемам относятся ТО, МО, ИО, ПО и др.

Информационное обеспечение – это комплекс единой системы классификации и кодирования технико-экономической информации, унифицированных систем документооборота, баз данных и знаний, которые располагаются в памяти ЭВМ.

Техническое обеспечение – это комплекс технических средств, необходимых для решения производственно-экономических задач (компьютеры, сетевое оборудование, принтеры и т. д.).

Математическое обеспечение – это совокупность математических моделей, методов оптимизации производственно-экономической деятельности и управления, алгоритмов решения производственно-экономических задач из функциональных подсистем.

Программное обеспечение – это комплекс программных средств, необходимых для функционирования информационной системы, включая базовое, сетевое, прикладное программное обеспечения. Прикладное программное обеспечение разрабатывается для каждой производственно-экономической задачи из соответствующей функциональной подсистемы.

Каждая функциональная подсистема включает в себя комплекс АРМ (руководителя, специалиста, секретаря, технического работника).

### **1.3. Основные принципы построения информационных систем**

Существует ряд основных принципов построения информационной системы, использование которых приводит к созданию на этапе проектирования эффективной информационной системы.

1. Принцип новых задач. При обследовании существующей системы управления предприятием необходимо выявлять узкие места в управлении, ликвидация которых позволит повысить эффективность всего производства. К узким местам на предприятии можно отнести сверхнормативные или недостаточные запасы материалов, нерациональную загрузку оборудования и др. Их ликвидация позволит повысить производительность и улучшить качество продукции; может быть осуществлена с помощью реализации новых производственно-экономических задач, в которых задействованы современные методы и подходы.

2. Системный подход к проектированию. Он предполагает системный анализ производственного объекта и окружающей среды на стадии предпроектного обследования. Системный анализ – это всестороннее основанное на определенном наборе правил исследование объекта в целом совместно с окружающей средой. По результатам исследования строят математическую модель, которая адекватно описывает функционирование этого объекта.

Этапы системного анализа:

постановка задачи в широком смысле; должна включать цель функционирования производственного объекта и возможные пути и средства достижения этой цели;

структуризация объекта и системы. Под ней понимается определение границ этого объекта и окружающей среды. Осуществляется разбиение объекта на отдельные части или подсистемы, для которых можно построить математическую модель функционирования объекта и его составных частей;

построение модели. В ней должны быть учтены наиболее важные характеристики объекта и окружающей среды;

исследование модели. Изменяют входные переменные модели и рассчитывают по модели выходные. Затем выходные переменные измеряют в реальных условиях и потом сравнивают с результатами моделирования путем вычисления значений заданного критерия адекватности. Если вычисленное значение критерия удовлетворяет поставленным условиям, то принимают решение об адекватности модели объекту.

3. Принцип первого руководителя. Сущность его сводится к тому, что все работы по проектированию и созданию информационных систем должны вестись под руководством первого руководителя этого объекта, так как только он знает цели, которые хочет реализовать, и те способы, с помощью которых предполагает их достичь. Поэтому все работы по проектированию информационных систем подразделяются на два вида:

выполняемые заказчиком;  
разработчиком.

Работы, выполняемые заказчиком:

определение цели управления объектом и показателей эффективности, которым должны удовлетворять каждая из разрабатываемых функциональных подсистем, разработка документа «Технико-экономическое обоснование проектных решений»;

установление приоритетности и очередности ввода функциональных подсистем и отдельных задач;

разработка информационного обеспечения (в первую очередь документооборота на предприятии). Заказчик проводит обучение персонала работе с новым информационным обеспечением.

Работы, выполняемые разработчиком:

постановка производственно-экономических задач и реализация алгоритма их решения;

разработка ПО по каждой производственно-экономической задаче;

внедрение информационной системы: установка и отладка оборудования, ИО и ПО, сдача в эксплуатацию информационной системы согласно акту внедрения.

4. Принцип типизации проектных решений. Сущность его состоит в том, что существуют уже апробированные типовые элементы информационных систем, которые необходимо тиражировать и применять при разработке новых информационных систем с целью сокращения сроков и затрат на проектирование и внедрение этих систем. Этот принцип не только сокращает затраты на разработку, но и повышает качество системы за счет применения наиболее эффективных способов построения комплекса технических средств, информационного обеспечения, пакетов прикладных программ.

5. Принцип непрерывного развития системы. При проектировании информационной системы разработчики должны учитывать возможность расширения функциональной и обеспечивающей частей системы в будущем в соответствии с требованиями научно-технического прогресса и рыночной ситуации. Все обеспечивающие подсистемы должны быть разработаны таким образом, чтобы можно было модифицировать уже существующие и наращивать число новых производственно-экономических задач, которые решаются в рамках информационной системы.

#### **1.4. Общая задача управления современным промышленным предприятием**

Информационная система разрабатывается и внедряется с определенной целью (повышение эффективности деятельности предприятия). Эта цель достигается путем моделирования различных вариантов функционирования предприятия в рамках общей задачи управления современ-

ным промышленным предприятием. В ней реализуются принципы системного подхода.

Решение общей задачи управления имеет две особенности:

общая задача управления – задача большой размерности, которая насчитывает большое число входных и выходных переменных;

на предприятии действует большое количество внешних возмущений, каждое из которых учесть невозможно, поэтому общая задача решается с помощью методов декомпозиции (т. е. разделения объекта на отдельные части); декомпозиция производится до такого уровня, когда можно эффективно решить общую задачу управления.

Общая задача управления сводится к обеспечению наиболее эффективного ведения технико-экономического процесса производства и реализации продукции в заданной номенклатуре и объеме.

Методы декомпозиции общей задачи управления:

1) структурная декомпозиция объекта на отдельные подпроцессы с целью более эффективного управления;

2) декомпозиция системы управления объектом на функциональные подсистемы;

3) декомпозиция во времени общей задачи управления, т. е. декомпозиция по периодичности решения.

### ***Общее описание систем в общей теории управления***

Системы управления подразделяются на децентрализованные и централизованные. Децентрализованные системы управления предполагают автономное управление отдельными подсистемами производственно-экономического объекта и применяются обычно для небольших объектов. В крупных объектах используются централизованные системы управления, поэтому в качестве примера рассмотрим двухуровневую централизованную систему управления (рис. 1.4).

Информация от производственного процесса  $P$  передается на подсистемы нижнего уровня и обрабатывается. Информация от подсистем нижнего уровня поступает на вышестоящую координирующую подсистему. Введем обозначения:

$C_0$  – вышестоящая координирующая подсистема;

$C_i$  –  $i$ -я подсистема нижнего уровня;

$m_i, \gamma_i$  – командные сигналы, поступающие от вышестоящей к нижестоящей подсистеме;

$z_i, \dot{\omega}_i$  – информационные сигналы обратной связи;

$w \in W$  – множество возмущений, действующих на объект;

$y \in Y$  – множество выходов производственного процесса  $P$ ;

$m \in M$  – множество управляющих воздействий, которые управляют производственным процессом  $P$ ;

$Y \in \Psi$  – множество координирующих командных сигналов от вышестоящей подсистемы к нижестоящим;

$z \in Z$  – множество информационных сигналов обратной связи о функционировании производственного процесса  $P$ ;

$\omega \in \Omega$  – информационные сигналы обратной связи (от нижестоящих подсистем к вышестоящей).

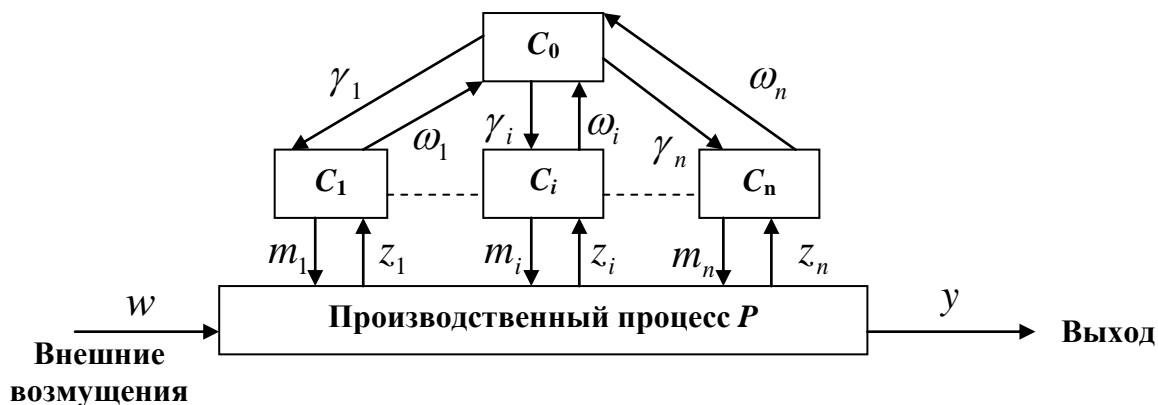
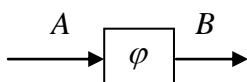


Рис. 1.4

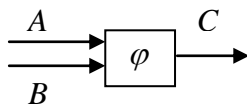
Таким образом, мы описали структуру централизованной двухуровневой системы управления.

Общее описание системы осуществляется с помощью теории отображения с применением входных и выходных множеств переменных для характеристики отдельных элементов или подсистем. Рассмотрим основные понятия теории отображения:



Отображение множества  $A$  во множество  $B$ , которое называется  $\varphi$  выглядит так:  $\varphi : A \rightarrow B$ . Каждому элементу  $a$  соответствует единственный элемент  $b$ :  $a \rightarrow b$ .

Отображение – функция или оператор, который может представлять из себя закон соответствия, позволяющий произвести сопоставление каждому элементу из множества  $A$  некоторый (единственный, определенный) элемент множества  $B$ :



$\varphi : A \cdot B \rightarrow C$  – декартово произведение множеств  $A$  и  $B$ .

Каждой паре входных элементов ставится в соответствие некоторый единственный элемент  $c$ :  $(a, b) \rightarrow c$ .

Математическая модель централизованной системы управления выглядит так:



$$\begin{aligned}
P &: M \times W \rightarrow Y \quad ; \\
C_i &: \Psi_i \times Z_i \rightarrow M_i \quad ; \\
M &= M_1 \times M_2 \times \dots \times M_n \quad ; \\
C_0 &: \Omega \rightarrow \Psi \quad ; \\
f_{pi} &: W \times M \times Y \rightarrow Z_i \quad (f_{pi} - \text{отдельная подсистема} \\
&\quad \text{производственного процесса } P); \\
f_{ci} &: \Psi_i \times M_i \times Z_i \rightarrow \Omega_i \quad .
\end{aligned}$$

Недостатки модели:

подсистемы нижнего уровня не обмениваются информацией друг с другом, поэтому командные сигналы от  $C_0$  должны нести информацию о функционировании этих подсистем;

нет прямых управляемых воздействий от вышестоящих подсистем  $C_0$  к управляемому объекту ( $P$ ).

Каждая из подсистем может подвергаться декомпозиции на отдельные процедурные модули. Декомпозиция производственного процесса  $P$ : производственный процесс можно разбить на отдельные подпроцессы  $P_i$  по числу управляющих подсистем (рис. 1.5).  $u_i$  – множество сигналов, с помощью которых подпроцесс  $P_i$  связывается с другими подпроцессами.  $H_i$  – связывающие функции подпроцессов;  $P_i: m_i \cdot w \cdot u_i \rightarrow y_i$ ,  $H_i: m_i \cdot y_i \rightarrow u_i$ .

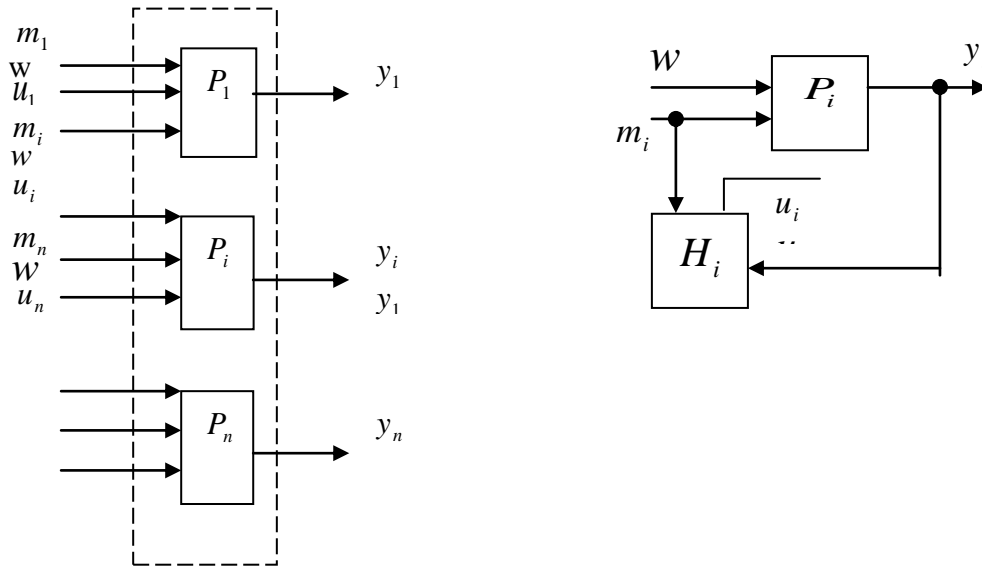


Рис. 1.5

Декомпозиция  $c_0$  показана на рис. 1.6.

Функции  $C_0$  – это модель принятия решения ( $d_0$ ), модель выработки координирующих управляющих сигналов ( $c'$ ).

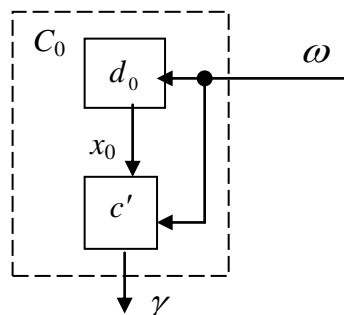


Рис. 1.6

$d_0 : \Omega \rightarrow x_0$  – найденное множество решений.

$c' : \Omega \cdot x_0 \rightarrow \Psi$  – выработанные координирующие воздействия на  $c_i$ .

Декомпозиция подсистемы  $C_i$  проводится аналогично, но с использованием четырех типов информационных сигналов.

### 1.5. Синтез структуры автоматизированной информационной системы и ее основные характеристики

В процессе проектирования информационной системы необходимо разрабатывать различные виды структуры этой системы. К ним относятся функциональная структура, структура информационных данных, структура экономических показателей, структура комплекса технических средств и другие виды структур, которые отображают одну из обеспечивающих или функциональных подсистем. Эти структуры должны удовлетворять выбранным критериям. Существуют два класса критериев:

критерии, которые базируются на показателях централизации информационной системы. К показателям централизации относятся число уровней иерархии управляющей системы, число подсистем на каждом уровне иерархии, степень централизации системы, норма управляемости, мера равномерности вертикальных связей;

критерии, связанные с такими показателями качества функционирования системы, как эффективность системы, надежность функционирования системы, быстродействие системы, достоверность обработки информации. Показатели качества связаны со сложными моделями и изучаются в специальных дисциплинах.

Показатели централизации:

1)  $\alpha_n$  – степень централизации между двумя смежными уровнями иерархии:

$$\alpha_n = W_n / W_{n-1},$$

где  $W_n$  – число подсистем на  $n$ -м уровне иерархии.

2)  $\alpha$  – степень централизации всей системы:

$$\alpha = \sum_{n=2}^N \beta_n \cdot \alpha_n$$

где  $\beta_n$  – весовой коэффициент, который задает важность задач для каждого уровня иерархии;  $N$  – количество уровней иерархии информационной системы.

3) норма управляемости – число подсистем, которые эффективно могут управляться одной вышестоящей подсистемой;

4) мера равномерности вертикальных связей:  
на  $n$ -м уровне системы:

$$R_n = \frac{\prod_{i=1}^K a_i}{(a_{cp})^K}$$

где  $K$  – количество подсистем на  $n$ -м уровне информационной системы.

$$a_{cp} = \frac{\sum_{i=2}^K a_i}{K}$$

Показатель  $a_{cp}$ , где  $a_i$  – число связей  $i$ -й подсистемы  $n$ -го уровня с подсистемами  $n - 1$  уровня;

для всей многоуровневой информационной системы:

$$R = \frac{\sum_{n=2}^N R_n}{N - 1}$$

## **1.6. Синтез функциональной структуры информационной системы. Матричный метод декомпозиции на основе матрицы связанности документов**

Синтез функциональной структуры информационной системы – это распределение функций управления объектом между отдельными функциональными подсистемами.

Одним из критериев распределения функций управления между функциональными подсистемами является критерий близости задачи, решаемой в одной подсистеме. Близость задачи можно идентифицировать, если использовать одни и те же производственно-экономические показатели, которые применяются в различных производственно-экономических задачах.

Матричный метод декомпозиции на основные матрицы связанности документов (рис. 1.7), в которых  $d \in D$  – множество документов, которые

отображают процесс функционирования и управления производственным процессом  $P$ , СУ – система управления производственным процессом  $P$ .

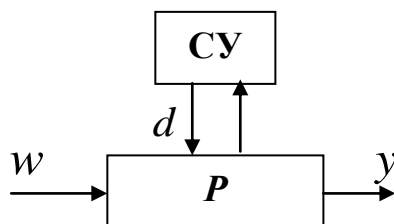


Рис. 1.7

В качестве критерия близости производственно-экономических задач будут выступать общие производственно-экономические показатели. Чтобы решить эти задачи необходимо выявить все документы, составить перечень документов и показателей в ней, сформировать матрицу связности документов (рис. 1.8).

	1	2	3	...	$j$	...	$n$
1	10	0	1				
2	0	6	2				
3	1	2	8				
...							
$i$					$a_{ij}$		
...							
$n$							

Рис. 1.8

Элементы матрицы по диагонали – это количество производственно-экономических показателей в каждом документе. Пересечение  $i$ -й строки и  $j$ -го столбца – количество общих показателей в  $i$ -м и  $j$ -м документах.

Возникает задача нахождения наиболее тесно связанных по показателям групп документов. Метод заключается в следующем: путем перемещения столбцов и строк увеличивают суммарную весовую связь данной матрицы с использованием критерия

$$\max \left\{ \sum_{i=1}^n \sum_{j=1}^n a_{ij} [a_{i,j-1} + a_{i,j+1} + a_{i-1,j} + a_{i+1,j}] \right\}$$

Путем перемещения столбцов и строк ищется максимум приведенного выше выражения. Документы с максимальными связями по показателям будут группироваться вокруг диагонали. Необходимо выделить блоки документов, тесно связанных по показателям, которые будут входить в ту или иную функциональную подсистему.

Алгоритм нахождения максимума указанного выше выражения:

выбираем произвольный столбец ( $j = 1$ ) и закрепляем его в качестве крайнего левого столбца матрицы;

каждый из оставшихся  $n-j$  столбцов располагаем последовательно справа от закрепленного столбца и подсчитываем вклад этих двух столбцов по формуле, приведенной ниже рис. 1.8;

столбец, который имеет наибольший вклад согласно выражению выше, закрепляем справа от  $j$ -го столбца, присваиваем значение  $j = j + 1$ ;

оставшиеся столбцы последовательно закрепляем справа от нового закрепленного столбца и рассчитываем вклад этих столбцов по вышеприведенной формуле. Тот столбец, который имеет наибольший вклад согласно этой формуле, закрепляем в качестве третьего столбца справа и т. д., пока не будут расположены все  $n$  столбцов.

После того, как закончатся перестановки столбцов, тот же алгоритм применяем для перестановки строк. В результате работы алгоритма образуются блоки наиболее тесно связанных по показателям документов. Рассмотрим применение алгоритма для матрицы из четырех документов:  $S_{1-2} = 10 \cdot (0 + 0) + 0 \cdot (10 + 7 + 1) + 1 \cdot (0 + 0 + 2) \cdot (1 + 1) = 6$ ;  $S_{1-3} = 10 \cdot (1 + 0) + 0 + 1 \cdot (3 + 2) + 2 \cdot (1 + 0) = 17$ ;  $S_{1-4} = 10 \cdot (2 + 0) + 0 + 1 \cdot (0 + 0 + 2) + 2 \cdot (1 + 4) = 32$  (рис. 1.9).

	1	2	3	4
1	10	0	1	2
2	0	7	0	1
3	1	0	3	0
4	2	1	0	4

Рис. 1.9

Наибольший вклад дает четвертый столбец:  $S_{4-2} = 2 \cdot (10 + 0 + 1) + 1 \cdot (2 + 7) + 0 + 4 \cdot (2 + 1) = 43$ ;  $S_{4-3} = 2 \cdot (10 + 1 + 1) + 1 \cdot (2 + 0) + 0 + 4 \cdot (2 + 0) = 34$  (рис. 1.10).

	1	4	2	3
1	10	2	0	1
2	0	1	7	0
3	1	0	0	3
4	2	4	1	0

Рис. 1.10

Блоки, расположенные вдоль основной диагонали, представляют собой основные группировки документов. Эти блоки содержат перечень документов, которые можно включить в конкретную функциональную подсистему. Блоки вне основной диагонали показывают на межгрупповые отношения и могут служить вспомогательной координирующей подсистемой более высокого уровня.

## **ГЛАВА 2. СТРУКТУРА ИНФОРМАЦИОННОЙ БАЗЫ И ОСНОВНЫЕ СВОЙСТВА ЕДИНИЦ ИНФОРМАЦИИ**

Определение и основные свойства единиц информации являются базовыми понятиями, через которые определяется структура информационной базы информационной системы.

### **2.1. Информационное обеспечение автоматизированных информационных систем**

Информационное обеспечение – совокупность информационных данных на машинных и немашинных носителях информации, системы классификации и кодирования технико-экономической информации, программных средств описания информационных данных и организационных принципов поддержки информационной системы (рис. 2.1).

Информационные данные могут быть размещены в производственно-экономических документах на немашинных носителях информации, на машинных носителях в виде записей в памяти ЭВМ, на различного типа машинных носителях информации.

Под записью понимается элемент данных, к которому обращается пользователь программного обеспечения для решения производственно-экономических задач. Записи можно интегрировать в электронные таблицы или элементы сетевой модели данных, элементы иерархической модели данных или в элементы последовательного линейного файла в зависимости от принятой модели данных, которую использует производственно-экономическая информационная система [2; 3].

Запись может быть разделена на мелкие и большие единицы информации, т. е. на атрибуты и составные единицы информации. В памяти ЭВМ атрибут – это ячейка, т. е. наименьшая единица информации. Запись интегрируется в модель данных, заполненная модель данных представляет собой базу данных, в которой хранятся детерминированные производственно-экономические показатели и другие массивы данных.

Информационные данные размещаются в базе знаний в соответствии с заданной моделью знаний. Если используется база знаний, то информационная система относится к классу интеллектуальных систем.

К ИО относится также система классификации и кодирования технико-экономической информации. Система классификации применяется для упорядочивания множества данных, когда они разбиваются на подмножества по заданным классификационным признакам. После классификации данные кодируют в соответствии с принятым методом кодирования технико-экономической информации.

К организационным принципам поддержки и ведения ИО относятся должностные инструкции по пользованию ИО. Они гарантируют системность управления доступом к информации, санкционированность доступа к информационным данным, сохранение целостности информационных данных.

### Структура информационного обеспечения ИС

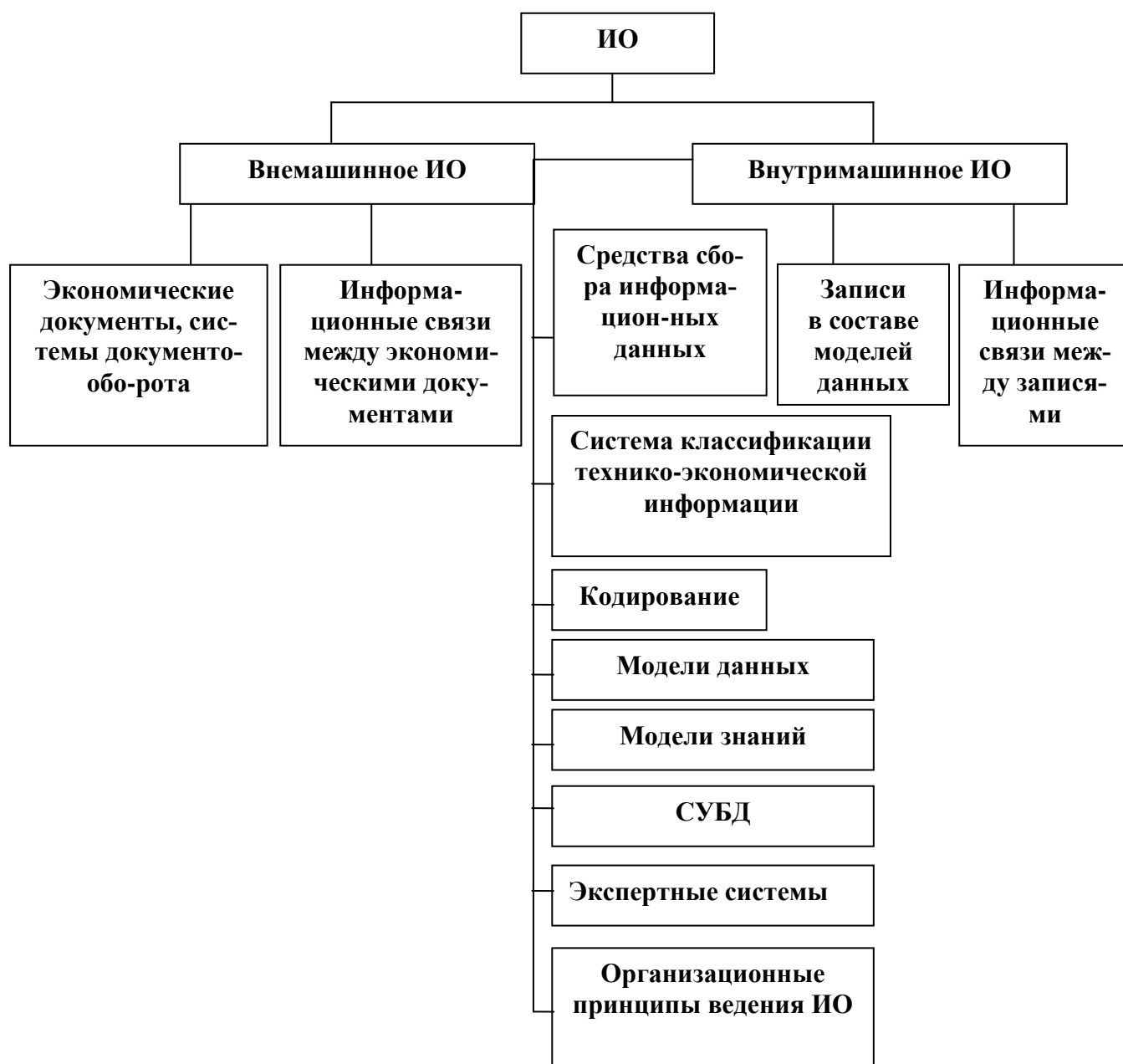


Рис. 2.1

## 2.2. Организация информационной базы

Информация, которая хранится на немашиных и машинных носителях, представляет собой информационную базу (ИБ) автоматизированной информационной системы. Схема информационной базы представлена на рис. 2.2, в которой:

- Ф – функциональная подсистема;
- З – производственно-экономическая задача;
- Д – производственно-экономические документы;
- П – производственно-экономические показатели;
- Р – реквизиты.

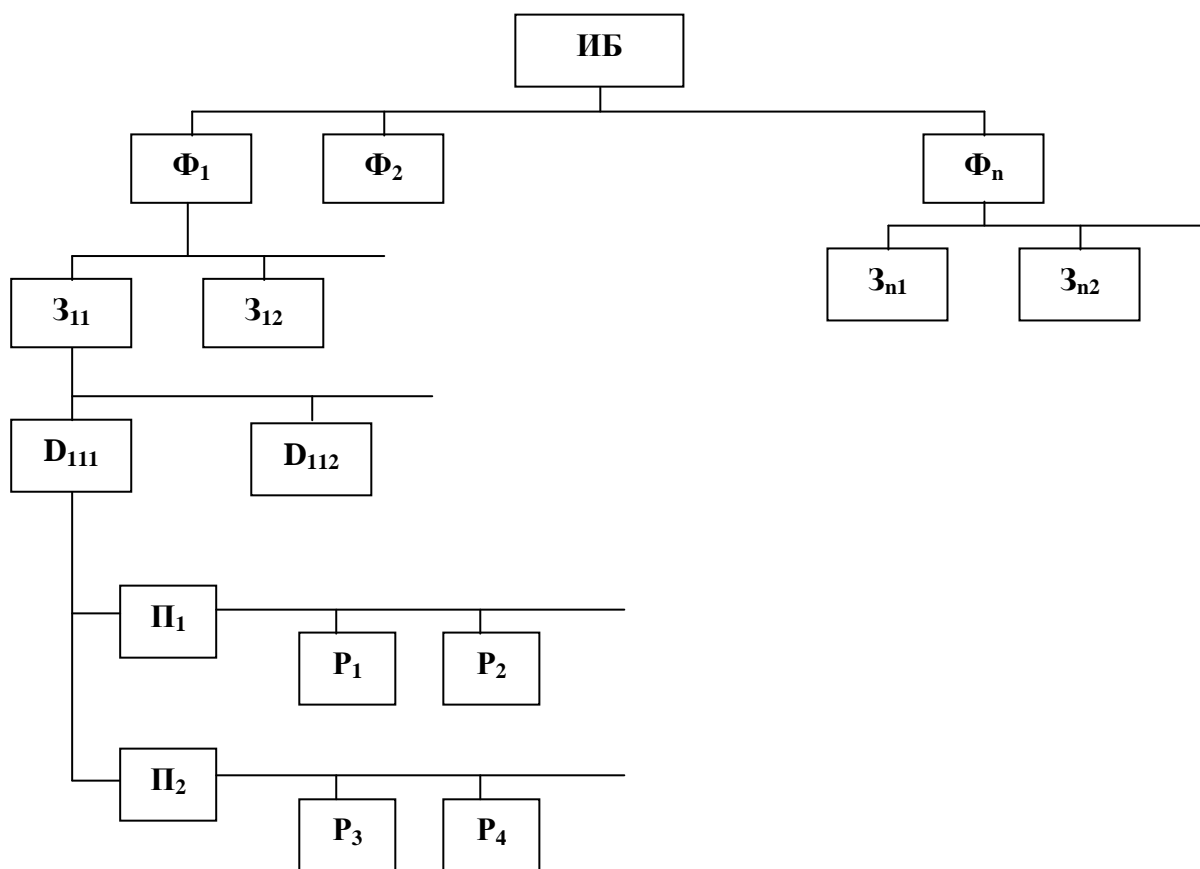


Рис. 2.2

Производственно-экономический документ – носитель упорядоченных данных, которые имеют юридическую силу в свете распоряжений и законов, которые действуют на данном предприятии и в данном подразделении.

Производственно-экономический документ служит для выполнения двух функций:

- является носителем упорядоченных сведений о производственно-экономическом объекте;

- служит юридическим основанием для принятия управленческих решений.



Производственно-экономический документ имеет типовую структуру построения и состоит из трех частей:

заголовочной;

основного содержания (приводятся значения производственно-экономических показателей);

оформительской (в ней указываются должность специалиста, который подготовил документ, должность руководителя, который подтверждает документ).

Требования к производственно-экономическому документу:

1. Наличие документа должно быть обосновано.

2. По возможности не должны дублироваться данные в производственно-экономических документах.

3. Производственно-экономические показатели, которые составляют основное содержание документа, должны располагаться в логической последовательности (в порядке вычисления или др.).

4. Информация, которую переносят на другой документ или на машинный носитель, должна группироваться в одном месте документа.

Реквизит (атрибут) – наименьшая единица производственно-экономической информации, которая отображает отдельный факт или событие. Реквизит определяется тремя характеристиками: именем; структурой, т. е. типом (текстовой, числовой) и длиной реквизита; значением.

Существует два вида реквизитов:

реквизиты-признаки – отражают качественные признаки объекта (предметы, продукты, время и т. п.) и служат для выполнения логических операций с показателями (например, сортировки данных);

реквизиты-основания – отражают количественные характеристики объекта (количество предметов, стоимость продуктов, временные сроки и т. п.) и необходимы для выполнения количественных расчетов с производственно-экономическими показателями.

Производственно-экономический показатель – совокупность одного реквизита-основания, который задает количественное значение показателя, и ряда реквизитов-признаков, логически связанных с реквизитом-основанием и друг с другом и задающих условия существования реквизита-основания.

Пример производственно-экономического показателя представлен в таблице.

Выпуск деталей по цеху 1

№ участка	Наименование детали	Шифр детали	Количество деталей
02	Болт	01137	55

## 2.3. Классификация и кодирование технико-экономической информации

Для однозначного описания данных, анализа, поиска и идентификации блоков производственно-экономической информации, упорядочивания информации о производственно-экономическом объекте, применяется система классификации технико-экономической информации. Система классификации – это набор правил, по которым множество объектов распределяется на подмножества в соответствии с классификационными признаками.

Признак классификации – это свойство объекта, по которому можно объекты отличать друг от друга или находить их сходства.

Процесс классификации – это результат деления заданного множества объектов на подмножества в соответствии с заданным методом классификации.

Существует два метода классификации:

фасетный;

иерархический (наиболее распространен).

Иерархический метод классификации: состоит в разбиении множества объектов  $M$  на подчиненные подмножества последовательно по нескольким уровням классификации (рис. 2.3). Степень классификации – этап деления заданного множества объектов на подмножества. Число степеней классификации называется глубиной классификации.

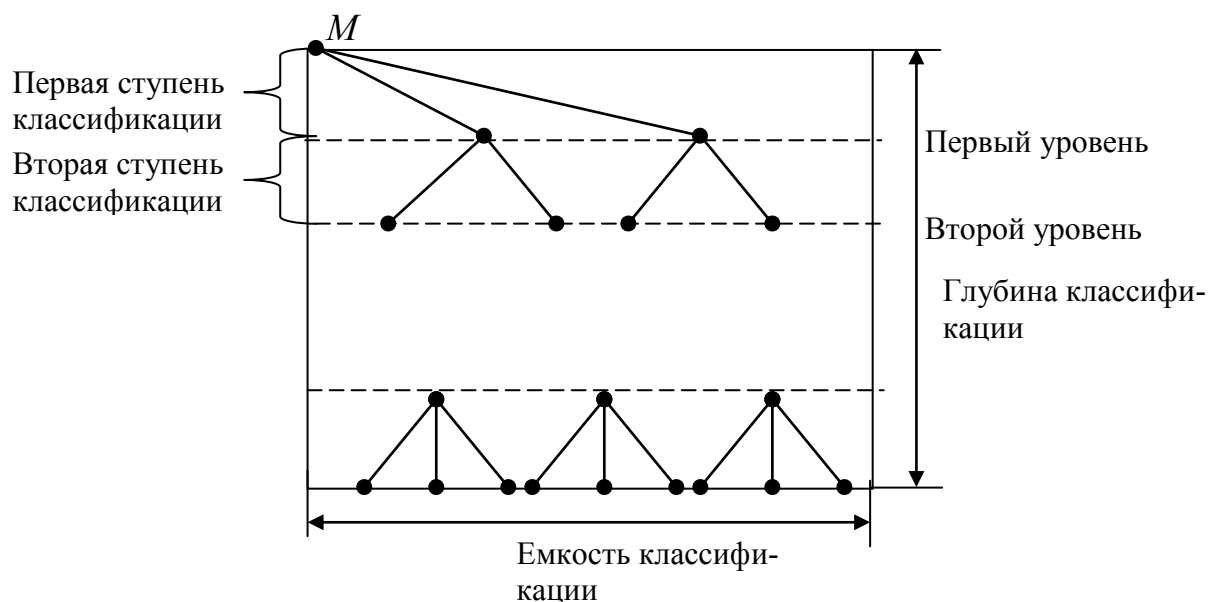


Рис. 2.3

Фасетный метод классификации отличается от иерархического тем, что заданное множество объектов  $M$  разбивается на группировки, не зависящие друг от друга, и группа объектов определяется путем простого перечисления фасет (признаков, которым она должна соответствовать). Ис-

ходное множество  $M$  может, к примеру, параллельно организовывать определенные группировки (рис. 2.4).

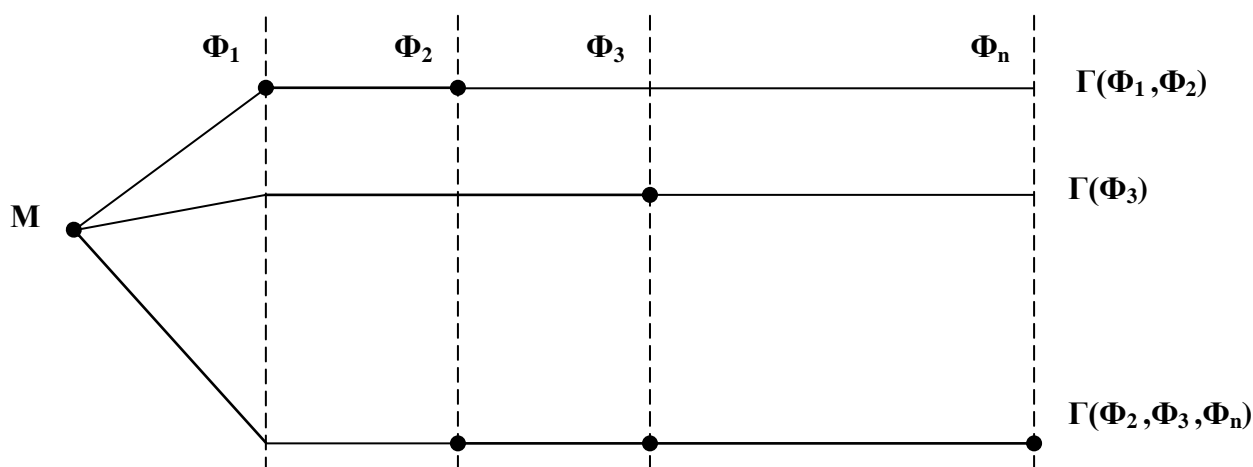


Рис. 2.4

После завершения классификации осуществляется кодирование технико-экономической информации, т. е. присвоение кодового обозначения (кода) объекту классификации, признаку классификации или подмножеству объектов. Условное кодовое обозначение называют сокращенно кодом. Длина кода – количество знаков в кодовом обозначении, например, А(5) – пять букв, 9(3) – три цифры, Х(6) – шесть буквенно-цифровых знаков.

Коды и классификаторы обязаны удовлетворять двум условиям:

они должны обеспечивать решение всех производственно-экономических задач;

кодовое обозначение должно иметь по возможности минимальную длину.

## 2.4. Методы кодирования технико-экономической информации

Существует три основных группы методов кодирования (рис. 2.5).

Метод порядкового кодирования: используется натуральный ряд чисел для кодирования. Пример – кодирование категорий персонала (рис. 2.6).

Порядковый метод с резервными номерами: при его применении используется натуральный ряд чисел для кодирования и выделяют резервные номера, чтобы можно было включить другие объекты кодирования. Пример – кодирование единиц измерения (рис. 2.7).



Рис. 2.5

Код	Рабочие
01	основные рабочие
02	вспомогательные рабочие
03	инженерно-технические работники

Рис. 2.6

Код	Единица измерения
1	мм
2	см
3	м
...	...
6	г
7	кг

Рис. 2.7

Серийно-порядковый метод: для заданного класса объектов выделяют диапазон значений чисел натурального ряда (рис. 2.8):

единицы длины: 01...10;  
 площади: 11...20;  
 объема: 21...30.

Код	Единица измерения
01	мм
02	см
11	мм <sup>2</sup>
12	см <sup>2</sup>

Рис. 2.8

Последовательный метод применяется в случае, если была использована иерархическая система классификации. Кодировать каждую ступень классификации (рис. 2.9).

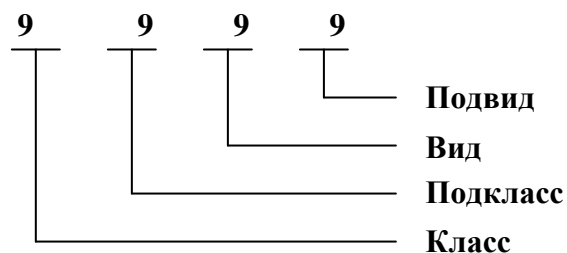


Рис. 2.9

Пример:

3000 – электронные приборы;  
 3500 – полупроводниковые приборы;  
 3540 – диоды полупроводниковые;  
 3541 – диоды германиевые.

Параллельный метод кодирования: применяется, если была использована фасетная система классификации. Разряды кода не зависят друг от друга. Пример: опора квадратная – 1; материал – сталь 5; размер – 40x40x62мм (рис. 2.10).

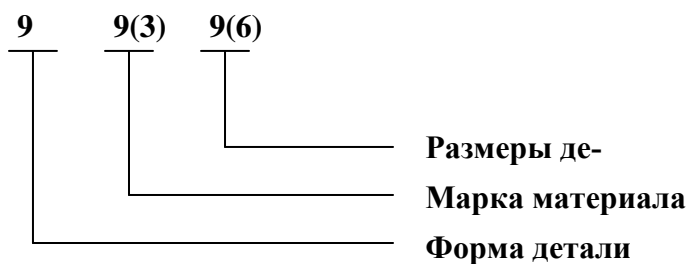


Рис. 2.10

Параллельно-последовательный метод: суть его – в применении комбинации последовательного и параллельного кодирования (рис. 2.11).

Часть выражения представляет собой последовательное кодирование (иерархический метод), другая часть – параллельное кодирование (фасетный метод).

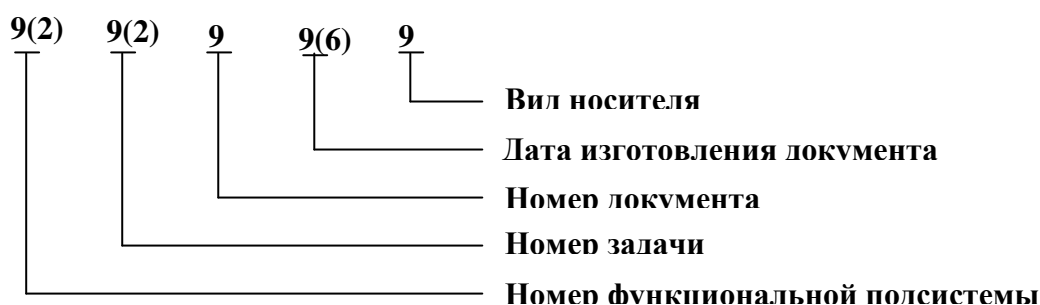


Рис. 2.11

Метод порядковой идентификации и отдельной классификации: его суть состоит в том, что идентификация элементов производится независимо от их классификации.

Код состоит из блоков (рис. 2.12):

- 1) идентификационного;
- 2) классификационных характеристик.

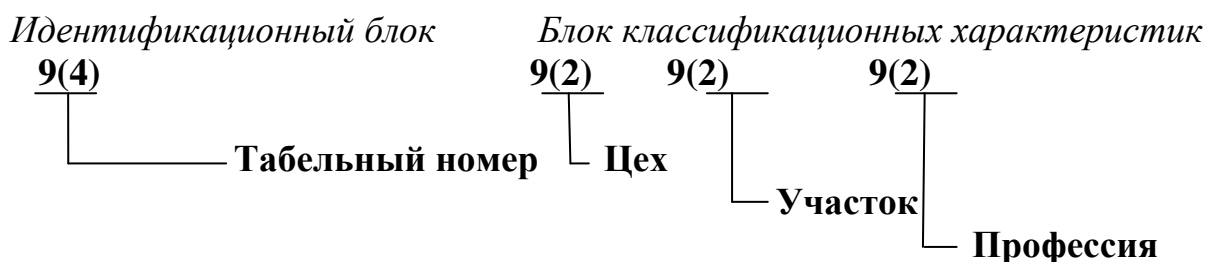


Рис. 2.12

Совмещенная идентификация и отдельная классификация: в блок идентификации вводят классифицирующие признаки для ускорения поиска и облегчения анализа информации (рис. 2.13).

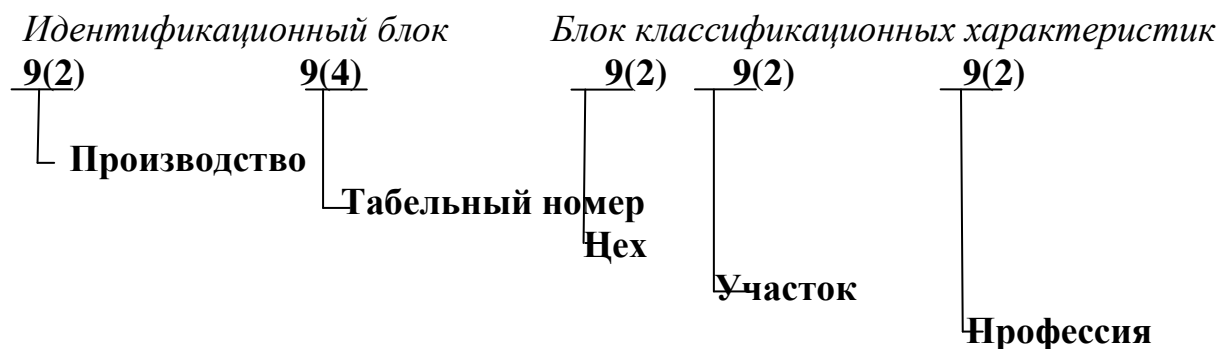


Рис. 2.13

## 2.5. Методы контроля ошибок в кодовом обозначении

При передаче кодового обозначения на значительные расстояния могут возникнуть искажения и какие-то разряды кода могут принять ошибочные значения, поэтому применяют методы помехоустойчивого кодирования и методы, позволяющие обнаружить и исправить ошибки в кодовом обозначении.

Существуют сложные методы помехоустойчивого кодирования – коды Хемминга, циклическое кодирование. Эти методы могут автоматически устранять ошибки.

### *Метод контроля ошибок по модулю 11*

Рассмотрим простейший метод контроля ошибок по модулю 11. Его сущность: каждому разряду кода от старшего к младшему присваивается соответствующий весовой коэффициент. Например, можно использовать натуральный ряд чисел, начиная с 1 при выборе последовательности весов, потом рассчитывается определенная сумма произведений и делится на 11, остаток – это контрольное число.  $d_i$  – разряды кода,  $w_i$  – вес  $i$ -го разряда:

$d_i$	$d_1$	$d_2$	$\dots$	$d_n$
$w_i$	1	2	$\dots$	$n$

Определяется контрольное число по алгоритму:

$$1) \sum_{i=1}^n w_i \cdot d_i ;$$

$$2) \sum_{i=1}^n w_i \cdot d_i - \left\lfloor \frac{\sum_{i=1}^n w_i \cdot d_i}{11} \right\rfloor = K, \text{ где } \lfloor \rfloor - \text{целая часть числа от деления}$$

на 11;  $K$  – контрольное число.

По каналу связи передается код вместе с контрольным числом:

$d_1$	$d_2$	$\dots$	$d_n$	$K$
-------	-------	---------	-------	-----

Пример расчета по данному методу:

$d_i$	2	4	0	3
$w_i$	1	2	3	4

$$2 \cdot 1 + 4 \cdot 2 + 0 \cdot 3 + 3 \cdot 4 = 22; 22/11 = 2 \Rightarrow K = 0 \text{ (остаток).}$$

## 2.6. Классификация и основные свойства единиц экономической информации

Основная неделимая единица информации в документах – реквизит. Он обычно носит название «атрибут» при внутримашинном представлении информации. Атрибут соответствует понятию переменной в языках программирования [9].

Итак, наименьшая неделимая единица информации в памяти ЭВМ – это атрибут. Атрибуты имеют имя, структуру, значение и бывают двух видов:

атрибут-признак (качественное свойство);

атрибут-основание (количественная характеристика).

Атрибут с именем  $x$  представляет собой пару  $(x, z)$ , где  $x$  – имя,  $z$  – значение атрибута  $x$  в заданный момент времени.

$z \in Z$  – множество возможных значений атрибута  $x$ , домен значений атрибута. Все допустимые значения атрибута образуют множество, называемое доменом этого атрибута.

Домен определяется списком значений, например, задается перечислением: день 1, ..., 31; месяц 1, ..., 12, но часто перечислить все элементы домена нельзя. В этом случае указывается тип и длина значения. Типы: текстовые –  $A$ , числовые –  $9$ , логические –  $1$ , смешанные –  $X$ . Пример – домен фамилий  $FAM$  (рис. 2.14).

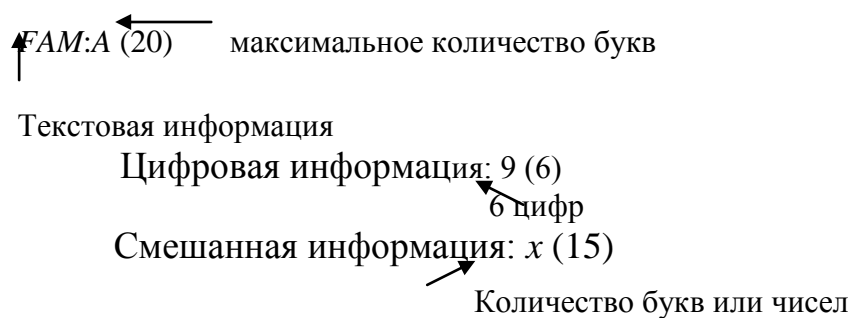


Рис. 2.14

Если в домене необходимо перечислить объекты из некоторого класса, то разрабатывается классификатор, содержащий коды объектов. Имеются также атрибуты-идентификаторы, например табельный номер работающего.

К основным единицам информации относится СЕИ. Составная единица информации – множество атрибутов и возможных других составных единиц. Составной единицей информации можно описать экономический документ. Множество атрибутов объединяются в составную единицу информации по принципам:

1) все атрибуты должны описывать один и тот же производственно-экономический процесс,



2) значения всех атрибутов, входящих в СЕИ, должны возникать одновременно и быть связанными друг с другом логическими или математическими отношениями.

Составная единица информации характеризуется также тремя характеристиками: именем, структурой (вхождением одних единиц информации в другие), значением. Составные единицы информации необходимы для описания структуры производственно-экономических документов. Общее описание СЕИ: после имени СЕИ указывают в скобках размерность (количество таких СЕИ), потом ставят точку и после нее в круглых скобках указывают список входящих в нее атрибутов и других СЕИ. Шаблон условного обозначения СЕИ:

<имя>(размерность).(имена атрибутов и других СЕИ).

Пример СЕИ показан на рис. 2.15.

ПРИХ

Дата	Код поставщика материала	Склад
01.10.01	311	02

Шифр материала	Кол-во материала, поступившего по накладной	Кол-во материала, поступившего на склад	Цена	Сумма
2613	10	10	400	4000
7911	7	6	500	3000

ПРИХ

Дата	Код поставщика материала	Склад
02.10.01	312	02

Шифр материала	Кол-во материала, поступившего по накладной	Кол-во материала, поступившего на склад	Цена	Сумма
7911	8	7	500	3500
5001	5	5	900	4500
2613	9	9	400	3600

ПРИХ(2).(ДАТА, КПМ, СКЛАД, ТАБЛ(3). (ШМ, КМПН КМПС, ЦЕНА, СУММА))

Рис. 2.15

Собранием СЕИ называется множество ее значений в составе СЕИ более высокого уровня. Количество значений в собрании СЕИ называется ее размерностью. Одно значение СЕИ приходного ордера содержит по од-

ному значению атрибутов ДАТА, КПМ, СКЛАД и собрание СЕИ ТАБЛ. Собрание ТАБЛ. включает три значения, в каждое значение входят по одному значению атрибутов ШМ, КМПН, КМПС, ЦЕНА, СУММА. В примере определено два значения СЕИ ПРИХ. Одно значение СЕИ при хранении в памяти ЭВМ обычно называют записью.

### **Определение производственно-экономического показателя**

При анализе документов ставится задача разделения документа на наименьшие понятые человеком фрагменты, называемые показателями. Показатель представляет собой полное описание количественного параметра, характеризующего некоторый объект. В состав показателя должны входить атрибуты: один атрибут-основание (количественное значение показателя) и несколько атрибутов-признаков, однозначно характеризующих условия существования атрибута-основания (это разновидность СЕИ).

Структура показателя  $P$ :  $P.(P_1, P_2, \dots, P_k, Q)$ , где  $P_1, P_2, \dots, P_k$  – атрибуты-признаки;  $Q$  – атрибут-основание.

В показателях отображаются количественные свойства объектов и процессов. Существуют документы, которые не содержат атрибутов-оснований, например анкеты кадрового учета. Минимальный набор атрибутов показателя:

- 1) атрибуты, отображающие идентификаторы объектов;
- 2) атрибуты, отображающие признак времени;
- 3) атрибут, отображающий некоторые количественные свойства объекта или процесса.

Производственно-экономические показатели позволяют создавать структуру базы данных с минимальной избыточностью, если все сведения в информационной системе разделить на производственно-экономические показатели, а потом объединить атрибуты родственных показателей следующим образом: в памяти ЭВМ один файл отводится под группу показателей с одинаковым составом атрибутов-признаков.

Перечислим атрибуты-основания из нашего примера: КМПН, КМПС, ЦЕНА, СУММА.

Документ состоит из четырех показателей:

$P1.(ШМ, КПМ, СКЛАД, ДАТА, КМПН)$ ;

$P2.(ШМ, КПМ, СКЛАД, ДАТА, КМПС)$ ;

$P3.(ШМ, ЦЕНА)$ ;

$P4.(ШМ, КПМ, СКЛАД, ДАТА, СУММА)$ .

Эти показатели образуют в базе данных два файла:  $F1$  с атрибутами ШМ, ЦЕНА и  $F2$  с атрибутами ШМ, КПМ, СКЛАД, ДАТА, КМПН, КМПС, СУММА.

Существует аналогия между показателями и переменными с индексами. Так, показатель  $P3.(ШМ, ЦЕНА)$  соответствует переменной  $C_i$ , где  $C$  – цена материала  $i$ -го ШМ. Показатель  $P2$  соответствует переменной  $КМПС_{ijmn}$ , где  $i$  – ШМ,  $j$  – СКЛАД,  $m$  – КПМ,  $n$  – ДАТА. Тогда  $СУММА_{Aijmn} = КМПС_{ijmn} \cdot C_i$ .

## 2.7. Основные операции, производимые с составной единицей информации

Существует пять основных операций с СЕИ.

**Нормализация** – операция преобразования СЕИ с произвольной многоуровневой структурой в СЕИ с двухуровневой структурой (двухуровневая СЕИ) (рис. 2.16).

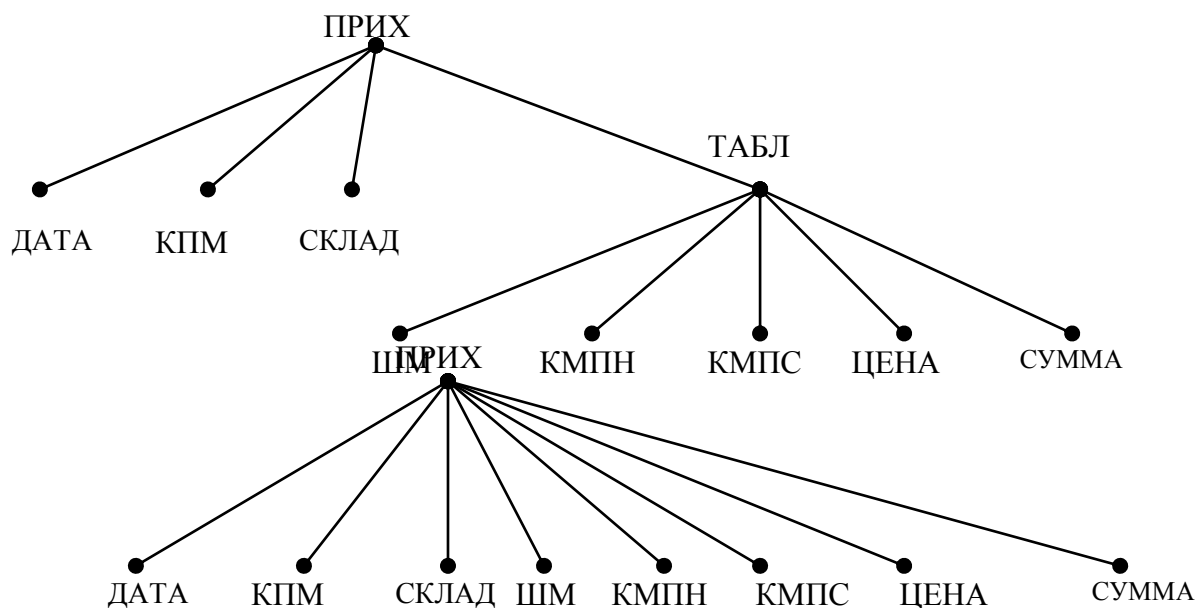


Рис. 2.16

Общее число значений в нормализованной СЕИ должно быть равно произведению размерностей всех СЕИ в исходном описании структуры. Нормализованная структура приходного ордера имеет вид, показанный на рис. 2.17.

ПРИХ							
ДАТА	КПМ	СКЛАД	ШМ	КМПН	КМПС	ЦЕНА	СУММА
01.10.01	311	02	2613	10	10	400	4000
01.10.01	311	02	7911	7	6	500	3000
02.10.01	312	02	7911	8	7	500	3500
02.10.01	312	02	5001	5	5	560	2800
02.10.01	312	02	2613	9	9	400	3600

ПРИХ(5).(ДАТА, КПМ, СКЛАД, ШМ, КМПН КМПС, ЦЕНА, СУММА)

Рис. 2.17

**Свертка** – операция преобразования СЕИ с двухуровневой структурой в СЕИ с произвольной многоуровневой структурой. Если в результате этой операции мы приходим к первоначальной СЕИ, то это будет взаимнообратная свертка. В результате свертки могут быть получены и другие ненормализованные СЕИ, но они должны иметь конкретный производственно-экономический смысл, например карточка складского учета. Их должно быть три (по числу материалов разного шифра (ШМ)) – 2613, 7911, 5001. Карточка складского учета представлена на рис. 2.18.

КАРТ

СКЛАД	ШМ	ЦЕНА
02	2613	400

ДАТА	КПМ	Кол-во материала, поступившего по накладной	Кол-во материала, по- ступившего на склад	СУММА
01.10.01	311	10	10	4000
02.10.01	312	9	9	3600

КАРТ(3).(СКЛАД, ШМ, ЦЕНА, ТАБЛ(2).(ДАТА, КПМ, КМПН, КМПС, СУММА))

Рис. 2.18

**Декомпозиция и композиция** – взаимнообратные СЕИ. **Декомпозиция** – разделение СЕИ на несколько составных единиц информации с различными структурами, несущими определенный экономический смысл. Причем условие декомпозиции состоит в том, что множество атрибутов составной единицы информации до декомпозиции, должно быть равно множеству атрибутов после декомпозиции. Например, СЕИ приходного ордера можно разделить на две: ЦЕНЫ (1). (ШМ, ЦЕНА); ПРИХ 1(1); (ДАТА, КПМ, СКЛАД, ШМ, КМПН, КМПС, СУММА) (рис. 2.19).

Суть композиции состоит в объединении нескольких СЕИ с разными структурами в одну при условии, что множество атрибутов, входящих в СЕИ до композиции и после, должны совпадать. Пример: композицию СЕИ ЦЕНЫ и ПРИХ1 приводят к исходной СЕИ (ПРИХ.).

ЦЕНЫ						
ШМ	ЦЕНА					
2613	400					
5001	560					
7911	500					
...	...					
...	...					

ПРИХ1						
ДАТА	КПМ	СКЛАД	ШМ	КМПН	КМПС	СУММА
01.10.01	311	02	2613	10	10	4000
01.10.01	311	02	7911	7	6	3000
02.10.01	312	02	7911	8	7	3500
02.10.01	312	02	5001	5	5	2800
02.10.01	312	02	2613	9	9	3600

Рис. 2.19

**Выборка** – это операция выделения подмножества значений СЕИ, которые удовлетворяют заранее поставленным условиям выборки, извлечения значения определенного атрибута, входящего в СЕИ по заданному условию выборки. Условия выборки могут быть сложными, например при получении статистических сводок.

Пример выборки по условию: ШМ = 7911, СУММА > 3000 руб., КМПН <> КМПС ^ ДАТА = 02.10.2001 г.

**Корректировка** – проведение одной из операций:

- 1) исключение старых значений СЕИ;
- 2) ввод новых значений атрибутов в СЕИ;
- 3) замена старых значений на новые;
- 4) одновременное внесение изменений в определенное множество СЕИ.

Пример корректировки: исключаем ШМ = 2613, включаем ШМ = 3115.

## ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПРИ ИЗУЧЕНИИ ГЛАВЫ 2

**Задание 2.1.** Даны реквизиты документа:

<i>Расход материалов</i>		
Участок	$K_{mat}$ – код материала	Цена
Деталь	Количество	Сумма
Операция		

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.2.** Даны реквизиты документа:

<i>Расчет загрузки оборудования</i>		
Цех	Количество станков	Коэффициент загрузки станка
Квартал	Код детали	
Код станка	План производства деталей	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.3.** Даны реквизиты документа:

<i>Ведомость сборки изделия</i>		
Код изделия	Цена изделия	Количество деталей на узел
Код узла	Цена детали	Количество деталей на изделие
Код детали		

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.4.** Даны реквизиты документа:

<i>Расчет численности рабочих</i>		
Участок	Норма времени на выпуск продукции	Плановое рабочее время
Квартал	Процент выполнения норм	Количество рабочих
	Процент потерь времени	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.5.** Даны реквизиты документа:

*Оборотная ведомость движения материалов*

Склад	Цена	Расход
Дата	Начальный остаток	Конечный остаток
Код материала	Приход	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.6.** Даны реквизиты документа:

*Ведомость отгруженных изделий*

Дата отгрузки	Цена	Сумма всего
Код изделия	Общая стоимость изделий	
Количество отгружено	Стоимость комплекта тары	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.7.** Даны реквизиты документа:

*Производственная программа участка*

Месяц	Код детали	План выпуска
Участок	План запуска	Остаток деталей
	Остаток прошлого месяца	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.8.** Даны реквизиты документа:

*Месячная программа участка*

Цех	Год	План выпуска на месяц
Участок	Код детали	План выпуска на квартал
Месяц	Расценка	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.9.** Даны реквизиты документа:

<i>Рапорт о выработке</i>		
Дата	Код детали и операции	Расценка
Участок	Разряд работы	Сумма
Табельный номер рабочего	Принято деталей	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.10.** Даны реквизиты документа:

<i>Счет за электроэнергию</i>		
ФИО	Дата	Сумма
Номер счета	Показание счетчика на дату	
Адрес	Расход электроэнергии	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.11.** Даны реквизиты документа:

<i>Нормативная калькуляция на детали</i>		
Участок	Расход материалов, руб.	Норматив себестоимости
Деталь	Норма времени	
Операция	Норматив заработной платы	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.12.** Даны реквизиты документа:

<i>Выпуск деталей</i>		
Цех	Количество станков	Фактический выпуск
Квартал	Код детали	
Код станка	План производства деталей	



Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.13.** Даны реквизиты документа:

*Обеспеченность материалами по цеху*

Цех	Приход материала	Отклонение
Месяц	Потребность на месяц	
Код материала	Остаток прошлого месяца	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.14.** Даны реквизиты документа:

*Акт о браке*

Дата	Операция	Себестоимость единицы брака
Участок	Табельный номер виновника	Сумма потерь от брака по участку
Деталь	Количество брака	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.15.** Даны реквизиты документа:

*Листок учета простоев*

Дата	Табельный номер рабочего	Сумма оплаты
Цех	Код причины простоя	Продолжительность простоя
	Процент оплаты простоя	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.16.** Даны реквизиты документа:

*План поставок на год*

Год	Адрес	План поставок, руб.
Изделие	Цена изделия	
Потребитель	План поставок, шт.	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.17.** Даны реквизиты документа:

<i>Ведомость на выдачу заработной платы</i>		
Год, месяц	Табельный номер рабочего	Сумма удержания
Цех	Начислено по табельному номеру	Сумма выдано
ФИО рабочего	Сумма аванса	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.18.** Даны реквизиты документа:

<i>Расход материалов на изготовление изделия»</i>		
Код изделия	Единица измерения	Количество изделий
Месяц	Норма расхода на изделие	Фактический расход материала за месяц
Код материала	Фактический расход на изделие	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.19.** Даны реквизиты документа:

<i>Лимитная карта на отпуск материалов</i>		
Дата	Цех-получатель	Цена
Код материала	Отпущено материала	Сумма
Цех-отправитель	Остаток материала	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.20.** Даны реквизиты документа:

<i>Карточка водителя</i>		
Табельный номер	Разряд	Часы работы
ФИО водителя	Номер путевого листа	Оплата по тарифу
Номер автомашины	Дата	Надбавка за ремонт

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.21.** Даны реквизиты документа:

<i>Кассовый отчет кинотеатра</i>		
Кинотеатр	Фильм	Число проданных мест
Число мест	Режиссер	Выручка от фильма
Дата		

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.22.** Даны реквизиты документа:

<i>Нормы расхода материалов</i>		
Цех	План выпуска деталей	Год, месяц
Код материала	Цена материала	
Код детали	Норма расхода материала на деталь	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.23.** Даны реквизиты документа:

<i>Учет готовой продукции предприятия</i>		
Код предприятия	Цена продукции	Процент налога на добавленную стоимость
Код продукции	Годовой оборот предприятия	
Год	Прибыль	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.24.** Даны реквизиты документа:

<i>Акт о ликвидации основных средств</i>		
Название объекта	Код цеха	Сумма износа
Год изготовления	Дата ликвидации	Остаточная стоимость
Инвентарный номер	Первоначальная стоимость	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.25.** Даны реквизиты документа:

*Ведомость резки заготовок*

Код детали	Выпуск изделия за год	Цех-получатель детали
Код изделия	Количество деталей из заготовки	Норма отходов на деталь
Код заготовки		

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.26.** Даны реквизиты документа:

*Нормативная калькуляция на изделие*

Участок	Изделие	Норматив себестоимости
Деталь	Норма времени	
Операция	Норматив заработной платы	

Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выберите самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

**Задание 2.27.** Даны реквизиты документа:

*Рапорт о выработке*

Дата	Код детали и операции	Расценка
Участок	Разряд работы	Сумма
Табельный номер рабочего	Принято деталей	

Выполните тест. Укажите номер правильного ответа.

1. Реквизит «Принято деталей» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

2. Реквизит «Расценка» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

3. Реквизит «Дата» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

**Задание 2.28.** Даны реквизиты документа:

<i>Ведомость отгруженных изделий</i>		
Дата отгрузки	Цена	Сумма всего
Код изделия	Общая стоимость изделий	
Количество отгружено	Стоимость комплекта тары	

Выполните тест. Укажите номер правильного ответа.

1. Реквизит «Цена» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

2. Реквизит «Код изделия» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

3. Реквизит «Отгружено» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

**Задание 2.29.** Даны реквизиты документа:

<i>Счет за электроэнергию</i>		
ФИО	Дата	Сумма
Номер счета	Показание счетчика на дату	
Адрес	Расход электроэнергии	

Укажите номер правильного ответа.

1. Реквизит «Адрес» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

2. Реквизит «ФИО» представляет:

- а) объект; б) свойство объекта;
- в) свойство взаимодействия.

3. Реквизит «Сумма» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

**Задание 2.30.** Даны реквизиты документа:

<i>Нормативная калькуляция на детали</i>		
Участок	Расход материалов, руб.	Норматив себестоимости
Деталь	Норма времени	
Операция	Норматив зарплаты	

Укажите номер правильного ответа.

1. Реквизит «Деталь» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

2. Реквизит «Норма времени» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

3. Реквизит «Участок» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

**Задание 2.31.** Даны реквизиты документа:

<i>Оборотная ведомость движения материалов</i>		
Склад	Цена	Расход
Дата	Начальный остаток	Конечный остаток
Код материала	Приход	

Укажите номер правильного ответа.

1. Реквизит «Склад» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия

2. Реквизит «Дата» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

3. Реквизит «Расход» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

**Задание 2.32.** Даны реквизиты документа:

<i>Производственная программа участка</i>		
Месяц	Код детали	План выпуска
Участок	План запуска	Остаток деталей
	Остаток прошлого месяца	

Укажите номер правильного ответа.

1. Реквизит «Код детали» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

2. Реквизит «План выпуска» представляет:

- а) объект;
- б) свойство объекта; в) свойство взаимодействия.

3. Реквизит «Месяц» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

**Задание 2.33.** Даны реквизиты документа:

<i>Расчет загрузки оборудования</i>		
Цех	Количество станков	Коэффициент загрузки станка
Квартал	Код детали	
Код станка	План производства деталей	

Укажите номер правильного ответа.

1. Реквизит «Цех» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

2. Реквизит «Количество станков» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

3. Реквизит «Код детали» представляет:

- а) объект;
- б) свойство объекта;
- в) свойство взаимодействия.

## ГЛАВА 3. МОДЕЛИ ДАННЫХ И ОПЕРАЦИИ НАД ОТНОШЕНИЯМИ

Модели данных – формализованное описание структуры составной единицы информации в памяти ЭВМ и множество операций, которые можно выполнить с СЕИ, хранящейся в памяти ЭВМ [4; 6].

Модель данных задается:

описанием множества информационных конструкций, допускаемых в данной модели данных;

множеством операций, которые можно выполнить с информационными данными, которые хранятся в данной модели данных;

описанием множества ограничений при хранении информационных данных в памяти ЭВМ.

Существует три основных модели данных:

реляционная,

сетевая,

иерархическая.

Основной элемент модели данных, с которым оперирует программист, – отношение. Отношение наиболее общего вида имеет реляционная модель. Отношение в сетевой и иерархической модели данных – частные случаи отношения в реляционной модели данных.

Отношение – структура связи между СЕИ и атрибутами, отображающими объекты с разных сторон, между СЕИ и атрибутами, описывающими различные свойства производственно-экономического объекта.

### 3.1. Реляционная модель данных

Реляционная модель данных обладает характеристиками:  
информационная конструкция – СЕИ с двухуровневой структурой;  
допустимое множество операций – это операции реляционной алгебры: проекция, выборка, объединение, пересечение, вычет, соединение, деление;

в качестве ограничений выступают функциональные зависимости между атрибутами в одном и том же отношении.

Пусть имеется множество объектов  $R$  одного класса. Свойства объектов описываются множеством атрибутов  $A_1, A_2, \dots, A_i, \dots, A_n$ . Обозначим множество производственно-экономических объектов класса  $R$  как  $R$ . Их свойства описывают множество атрибутов  $R(A_1, A_2, \dots, A_i, \dots, A_n)$ . Всему классу объектов соответствует множество кортежей, называемое отношением. Обозначим отношение, описывающее класс объектов, также через  $R$ .

Один объект класса  $R$  описывается строкой значений этих атрибутов (называется кортежем)  $A_i(a_1, a_2, \dots, a_i, \dots, a_n)$ , где  $a_i$  – значение атрибута  $A_i$ .

Множество кортежей составляют отношения в реляционной модели данных.



$R(A_1; A_2; \dots A_i; \dots A_n)$  – схема отношения. Для каждого компонента кортежа должна быть указана его связь с соответствующим атрибутом. В реляционной модели данных для обеспечения этой связи предусмотрено условие: порядок компонентов кортежа должен совпадать с порядком следования атрибутов в схеме отношения. В разные моменты времени одной схеме отношения могут соответствовать разные значения атрибутов отношения. Множество значений отношения можно представить в виде таблицы, составленной по определенным правилам:

название таблицы и перечень названий граф соответствуют схеме отношения;

кортеж отношения соответствует строке таблицы;

все строки таблицы различны;

порядок строк и столбцов может быть произвольным.

Схема отношений задает порядок следования компонент отношения, а этот порядок задается порядком следования атрибутов. Отношение реляционной модели данных можно записать в виде таблицы (рис. 3.1).

$A_1$	$A_2$	...	$A_i$	...	$A_n$
$a_{11}$	$a_{12}$	...	$a_{1i}$	...	$a_{1n}$
$a_{21}$	$a_{22}$	...	$a_{2i}$	...	$a_{2n}$
...	...	...	...	...	...
$a_{m1}$	$a_{m1}$	...	$a_{mi}$	...	$a_{mn}$

Рис. 3.1

Столбец – домен значений определенного атрибута. Строка – кортеж, описывающий определенный объект из заданного множества объектов.  $r \in R$  – объект, включающий в себя множество объектов  $r$ .

Реляционная модель данных и операции, производимые с ней, можно описать с помощью теории множеств. Результатом применения теории множеств в реляционной модели данных стала реляционная алгебра.

Существует два основных метода обработки отношений:

процедурное описание обработки, когда результирующее отношение задается последовательностью процедур обработки исходного отношения;

декларативное описание обработки данных, когда задаются свойства результирующего отношения и путем логических операций система управления БД сама своими средствами выполняет логические операции по созданию результирующего отношения.

### 3.2. Процедурное описание процессов обработки данных

Система отношений и операций над ними определяет процедуры реляционной алгебры: проекцию, выборку, объединение, пересечение, вычитание, соединение и деление.

**Проекция** – процедура, с помощью которой из исходного отношения в результирующее отношение переносятся атрибуты, заданные условием проекции.

Если  $T$  – результирующее отношение,  $W$  – исходное отношение,  $X$  – условие проекции (перечень атрибутов, который переносится в результирующее отношение), то  $T = W [X]$ .

Пример: пусть имеется исходное отношение  $W$  (рис. 3.2), требуется получить результирующее отношение  $T1$  о фактическом выпуске продукции (рис. 3.3)

$W$			
Завод	Изделие	План	Факт
«Динамо»	ЭДВ-12	100	100
«Динамо»	ЭДВ-10	150	140
«Ротор»	ЭДВ-12	200	210

Рис. 3.2

$T1 = W [\text{Завод, Изделие, Факт}]$

$T1$		
Завод	Изделие	Факт
«Динамо»	ЭДВ-12	100
«Динамо»	ЭДВ-10	140
«Ротор»	ЭДВ-12	210

Рис. 3.3

**Операция выборки** – это операция, переносящая из исходного отношения в результирующее те строки из исходного, которые удовлетворяют условиям выборки.

Алгоритм операции выборки с примерами:

а) <имя атрибута> <знак сравнения> <значение>: План > 150;

б) <имя атрибута\_1> <знак сравнения> < имя атрибута\_2>:  
Факт > План.

$T = W [P]$ , где  $P$  – условие выборки.  $T2 = W [\text{Изделие} = \text{ЭДВ-12}]$  (рис. 3.4).

$T2$		
Завод	Изделие	Факт
«Динамо»	ЭДВ-12	100
«Ротор»	ЭДВ-12	210

Рис. 3.4

**Объединение** – операция, при которой в результирующее отношение переносятся строки исходных отношений, присутствующие или в первом, или во втором исходном отношении.

**Пересечение** – операция, при которой в результирующее отношение переносятся строки исходных отношений, присутствующие и в первом, и во втором исходном отношении одновременно.

**Вычитание** – операция, при которой в результирующее отношение переносятся строки первого исходного отношения (за исключением одинаковых строк в первом и втором исходном отношении).

$R1$  и  $R2$  – исходные отношения (рис. 3.5, 3.6),  $RA$  – объединение (рис. 3.7),  $RB$  – пересечение,  $RC$  – вычитание. Тогда  $RA = R1 \cup R2$ ,  $RB = R1 \cap R2$  (рис. 3.8),  $RC = R1 \setminus R2$  (рис. 3.9).

$R1$		
$G$	$B$	$D$
1g	2	2d
3g	6	2d
7g	7	3d

Рис. 3.5

$R2$		
$G$	$B$	$D$
1g	2	1d
3g	6	2d

Рис. 3.6

$RA$		
$G$	$B$	$D$
1g	2	2d
3g	6	2d
7g	7	3d
1g	2	1d

Рис. 3.7

$RB$		
$G$	$B$	$D$
3g	6	2d

Рис. 3.8

$RC$		
$G$	$B$	$D$
1g	2	1d
7g	7	3d

Рис. 3.9

**Соединение отношений** – операция над двумя исходными отношениями, заключающаяся в том, что каждая строка первого исходного отношения последовательно сравнивается с каждой строкой второго; если удовлетворяется условие соединения, то эти строки сцепляются и образуют очередную строку в результирующем отношении  $T$ .

Условие соединения имеет вид

<имя атрибута\_1(первое исходное отношение)> <знак сравнения>  
< имя атрибута\_2(второе исходное отношение)>.

$T = R1 [P] R2$ , где  $R1, R2$  – исходные отношения;  $P$  – условие соединения.

Выделяют так называемое натуральное соединение, которое наиболее часто используется.

Условия соединения: 1) знак сравнения – это равенство (=); 2) <имя атрибута\_1> и <имя атрибута\_2> должны совпадать, а результат должен содержать объединение списков атрибутов исходных отношений. Натуральное соединение  $T = R1 \cdot R2$ .

Пример: отношение  $W$  дополним данными о ценах на продукцию  $W2$ .

Значения  $W2$  показаны на рис. 3.10.  $T3 = W [Изд. = Изд.] W2$  (рис. 3.11).

W2	
Изделие	Цена
ЭДВ-10	1000
ЭДВ-11	2000
ЭДВ-12	3000

Рис. 3.10

T3				
Завод	Изделие	План	Факт	Цена
«Динамо»	ЭДВ-12	100	100	3000
«Динамо»	ЭДВ-10	150	140	1000
«Ротор»	ЭДВ-12	200	210	3000

Рис. 3.11

Как видно из примера, в результате натурального соединения будет отсутствовать второй столбец с именем «Изделие».

**Операцию деление отношений** рассмотрим на примере. Пусть задано отношение  $T4$ . Необходимо найти ФИО программистов, знающих два языка – «Си» и «Фортран». Операция выборки будет безуспешной, так как в одной строке нет информации о двух языках программирования. Выполнить процедуру можно путем вычисления образов  $im$  значений  $C$  и «Фортран» и последующего пересечения этих образов:

$im \text{ ФИО } (C) = (\text{Иванов, Петров, Сидоров});$

$im \text{ ФИО } (\text{Фортран}) = (\text{Иванов, Сидоров});$

$im \text{ ФИО } (C) \wedge im \text{ ФИО } (\text{Фортран}) = (\text{Иванов, Сидоров}).$

Такая связка операций взятия образа и пересечения полученных множеств называется операцией деления. В примере  $T4$  – отношение- делимое,  $Y$  – отношение-делитель, результирующее отношение  $T5$ . Условное обозначение операции:  $T5 = D (T4, Y)$ , где  $D$  – знак операции деления.

Отношение-делимое  $T4$  показано на рис. 3.12, отношение- делитель  $Y$  – на рис. 3.13.

$T4$	
ФИО	Язык программирования
Иванов	С
Иванов	«Фортран»
Иванов	«Пролог»
Петров	С
Петров	«Фортран»
Сидоров	С
Сидоров	«Пролог»

Рис. 3.12

$Y$
Языки программирования
«Си»
«Фортран»

Рис. 3.13

Определите, кто знает языки «Си» и «Фортран». Результирующее отношение  $T5$  выглядит, как показано на рис. 3.14.

$T5$
ФИО
Иванов
Петров

Рис. 3.14

### 3.3. Нормализация отношений

Множество атрибутов могут группироваться в отношении произвольным образом. Произвольные группировки могут задавать неэффективные отношения, которые страдают избыточностью информации [5; 7]. Необходимо выбрать критерий группировки атрибутов в отношении. Таких критериев три:

образованные отношения должны обладать минимальной избыточностью информации;

при корректировке отношений не должна происходить потеря информации или искажение смысла;

перестройка набора отношений при добавлении в базу данных новых атрибутов должна занимать минимальное количество операций.

Нормализация представляет собой один из наиболее изученных способов преобразования отношений, позволяющий улучшить характеристики базы данных по перечисленным критериям. Процесс преобразования отношений базы данных к той или иной нормальной форме называется нормализацией отношений. Нормальные формы нумеруются последовательно от единицы («1») по возрастанию (чем выше номер, тем больше ограничений должно соблюдаться). В процессе нормализации происходит переход из одной нормальной формы в другую; чем выше порядковый номер нормальной формы, тем более эффективны отношения с точки зрения избыточности хранимой информации. Отношение в первой нормальной форме (1НФ) – это обычное отношение с двухуровневой структурой. Преобразование ненормализованного отношения к 1НФ – это операция нормализации, рассмотренная для СЕИ:

1 НФ

2 НФ            номера нормальных формы отношений, чем больше номер, тем

3 НФ            больше ограничений должно соблюдаться.

Первая НФ – составная единица информации с двухуровневой структурой. Эта форма всегда достижима. Для достижения 2НФ, 3НФ и т. д. следует применять правила нормализации с использованием понятия функциональных зависимостей.

### **3.3.1. Функциональная зависимость и ключи**

Функциональные зависимости определяются для атрибутов, находящихся в одном и том же отношении, удовлетворяющих первой нормальной форме (1НФ).

Простейший случай функциональной зависимости охватывает два атрибута. В отношении  $T(A, B)$ , где  $A$  и  $B$  – атрибуты, атрибут  $A$  функционально определяет атрибут  $B$ , если в любой момент времени каждому значению атрибута  $A$  соответствует единственное значение атрибута  $B$ . В любой момент времени каждому значению  $A$  соответствует значение  $B$  ( $A \rightarrow B$ ) или  $B$  функционально зависит от  $A$  ( $B = f(A)$ ). Отсутствие функциональной зависимости обозначается как  $A \not\rightarrow B$ . Пример: в

ФИО	Год рождения
1	1980
2	1980
3	1981

есть функциональная зависимость ФИО  $\rightarrow$  Год рождения, так как у каждого человека единственный год рождения. Обратной зависимости нет: ФИО  $\not\rightarrow$  Год рождения.

Одновременное соблюдение двух зависимостей ( $A \rightarrow B$  и  $B \rightarrow A$ ) называется взаимно-однозначным соответствием и обозначается как  $A \leftrightarrow B$ .  
Пример: в

<i>T</i>	
Завод	р/с
1	11
2	12
3	13

завод  $\rightarrow$  р/с и р/с  $\rightarrow$  завод. Это означает, что завод  $\leftrightarrow$  р/с.

Существуют следующие ситуации для атрибутов  $A$  и  $B$  некоторого отношения:

- отсутствие функциональной зависимости;
- наличие односторонней функциональной зависимости;
- наличие взаимно-однозначного соответствия.

Самым распространенной является отсутствие функциональной зависимости.

Понятие функциональной зависимости распространяется на ситуацию с тремя и более атрибутами. Группа атрибутов  $A, B, C$  функционально определяет атрибут  $D$  в отношении  $T(A, B, C, D)$ ;  $A, B, C \rightarrow D$ , если каждому сочетанию значений  $(a, b, c)$  соответствует единственное значение  $d$ . Такую функциональную зависимость обозначают  $A, B, C \rightarrow D$ . Это значит, что каждому сочетанию  $(a, b, c)$  соответствует единственное значение  $d$ . Для производственно-экономического показателя  $\Pi.(P_1, P_2, \dots, P_n, Q)$  справедлива функциональная зависимость  $P_1, P_2, \dots, P_n \rightarrow Q$ , хотя это не единственная функциональная зависимость для этих атрибутов.

С помощью функциональной зависимости определяется понятие ключа отношения. Вероятным ключом отношения называется такое множество атрибутов  $K$ , в котором каждое сочетание их значений встречается только в одной строке отношений и никакое подмножество  $K$  не обладает этим свойством. Вероятных ключей в отношении может быть несколько.

Пример ключа отношений дан на рис. 3.15.

<i>T</i>				
<i>ZEN</i>	<i>RAM</i>	<i>AST</i>		<i>BIG</i>
1A	31	DWA		73
3B	01	BUN		40
3D	10	MUN		58
4D	31	BUN		38

Рис. 3.15

Можно утверждать, что вероятным ключом отношения  $T$  является атрибут  $ZEN$ , так как его значения в столбце не повторяются. Кроме того, еще один вероятный ключ представлен парой атрибутов  $RAM$  и  $AST$ :  $ZEN \rightarrow BIG$ ;  $RAM, AST \rightarrow BIG$ . На практике атрибуты вероятного ключа

отношений связывают со свойствами тех объектов и событий, информация о которых хранится в отношении.

Если в результате корректировки отношения изменились имена атрибутов, образующих ключ, то это свидетельствует о серьезном искажении информации. Систематическая проверка свойств вероятного ключа позволяет следить за достоверностью информации в отношении. Если в отношении присутствует несколько вероятных ключей, то одновременное слежение за ними осложнено, поэтому целесообразно выбрать один из них в качестве основного (первичного).

Первичным ключом отношений называется такой вероятный ключ, по значению которого производят контроль достоверности информации в отношении. Первичный ключ часто называют просто «ключ». Каждое значение первичного ключа есть только в одной строке отношения. Значение любого атрибута в этой строке также единственное, поскольку используется 1НФ. Если через  $K_{нк}$  обозначим множество атрибутов первичного ключа в  $R(A, B, C, \dots, J)$ , то справедливы функциональные зависимости  $K_{нк} \rightarrow A$ ,  $K_{нк} \rightarrow B$ ,  $K_{нк} \rightarrow C, \dots, K_{нк} \rightarrow J$ . Первичный ключ найден, если группа атрибутов определяет все атрибуты по отдельности и группу сократить нельзя.

Для множества функциональных зависимостей существует ряд закономерностей, которые выражаются теоремами. Зная теоремы, можно из исходного множества функциональных зависимостей получить производные зависимости. Рассмотрим теоремы для атрибутов, находящихся в одном и том же отношении:

- 1) если  $A \rightarrow BC$ , то  $A \rightarrow B$  и  $A \rightarrow C$ ;
- 2) если существуют зависимости  $A \rightarrow B$ ,  $B \rightarrow C$ , то  $A \rightarrow C$ ;
- 3) если  $A \rightarrow B$ , то  $AC \rightarrow BC$ , где  $C$  – произвольный атрибут;
- 4) если  $A \rightarrow B$  и  $BC \rightarrow D$ , то  $AC \rightarrow D$ .

Пример применения теоремы: пусть имеются  $A, B$ . Тогда  $AB \rightarrow A$  и  $AB \rightarrow B$ .

Если заранее известно, что вероятный ключ в отношении только один, его можно найти простым способом. Вероятный ключ – это набор атрибутов, которые не встречаются в правых частях всех функциональных зависимостей, т. е. из полного списка атрибутов отношения необходимо вычеркнуть атрибуты, которые встречаются в правых частях всех функциональных зависимостей. Оставшиеся атрибуты образуют первичный ключ.



### 3.3.2. Вторая и третья нормальные формы отношения

Отношение имеет 2НФ, если оно соответствует 1НФ и не содержит неполных функциональных зависимостей. Неполная функциональная зависимость имеет следующие свойства:

1. Вероятный ключ отношения функционально определяет некоторый неключевой атрибут.

2. Часть вероятного ключа функционально определяет этот же неключевой атрибут:  $K = C, E$ . Однако  $K \rightarrow A, E \rightarrow A$ .

Отношение, не соответствующее 2НФ, имеет избыточность хранимых данных (рис. 3.16).

T1			
Цех	Изделие	Цена	План
1	M22	5	200
1	M23	6	100
2	M22	5	300
2	K23	7	400

Рис. 3.16

Избыточность состоит в том, что цена изделия указывается столько раз, сколько цехов производят одинаковое изделие: ЦЕХ, ИЗДЕЛИЕ  $\rightarrow$  ЦЕНА. Часть вероятного ключа отношения ИЗДЕЛИЕ  $\rightarrow$  ЦЕНА. Переход к 2НФ и, соответственно, устранение избыточных данных связано с созданием двух отношений вместо исходного:  $T11 = T1(\text{ЦЕХ, ИЗДЕЛИЕ, ПЛАН})$  (рис. 3.17). Получаем:  $T12 = T1(\text{ИЗДЕЛИЕ, ЦЕНА})$  (рис. 3.18)

T11		
Цех	Изделие	План
1	M22	200
1	M23	100
2	M22	300
2	K23	400

Рис. 3.17

T12	
Изделие	Цена
M22	5
M23	6
K23	7

Рис. 3.18

Отношение соответствует 3НФ, если оно соответствует 2НФ и среди его атрибутов отсутствуют транзитивные функциональные зависимости. Транзитивные функциональные зависимости – это две функциональные зависимости: вероятный ключ отношения функционально определяет не-

ключевой атрибут; этот неключевой атрибут функционально определяет другой неключевой атрибут.

Если  $K$  – ключ отношения,  $A$  и  $B$  – неключевые атрибуты и справедливы зависимости  $K \rightarrow A$ ,  $A \rightarrow B$ , то эти функциональные зависимости называются транзитивными. Частный случай транзитивной функциональной зависимости – неполная функциональная зависимость, когда  $K = C, E$ ,  $K \rightarrow E$  и  $E \rightarrow A$ .

Пример. Ключ отношения  $T2$  – ФИО. Тогда транзитивная функциональная зависимость выглядит так: ФИО  $\rightarrow$  Группа и Группа  $\rightarrow$  Факультет. Отношение  $T2$  (рис. 3.19) находится в 2НФ, но не в 3НФ.

$T2$		
ФИО	Группа	Факультет
1. Иванов	31	ФИТ
2. Петров	31	ФИТ
3. Сидоров	32	ФИТ
4. Федин	32	ФИТ

Рис. 3.19

Избыточность данных в отношении  $T2$  связана с тем, что принадлежность группы к факультету указывается столько раз, сколько обучающихся в этой группе. Переходы  $T2$  к отношениям в 3НФ:  $T21 = T2(\text{ФИО}, \text{Группа})$  (рис. 3.20) и  $T22 = (\text{Группа}, \text{Факультет})$  (рис. 3.21).

$T21$	
ФИО	Группа
1. Иванов	31
2. Петров	31
3. Сидоров	32
4. Федин	32

Рис. 3.20

$T22$	
Группа	Факультет
31	ФИТ
32	ФИТ

Рис. 3.21

### 3.4. Сетевая модель данных

Сетевая модель данных – это множество веерных отношений и всех прочих ( $W(R,S)$ ).

Веерное отношение – это пара отношений  $R$  и  $S$ : одно – основное (исходное) отношение  $R$ , другое – зависимое (порожденное) отношение  $S$ ;

между ними существует взаимосвязь. Каждое значение зависимого отношения связано с одним значением основного отношения при условии, что каждое значение  $S$  связано с единственным значением  $R$  (рис. 3.22). Допустимые в сетевой модели данных операции представляют собой различные варианты выборки.

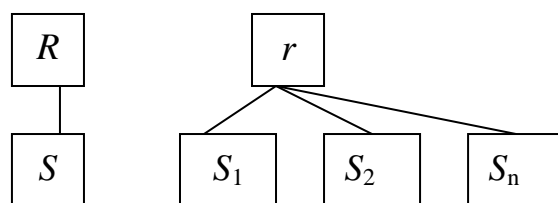


Рис. 3.22

Существует два вида сетевых моделей: двухуровневая, многоуровневые.

Многоуровневые модели данных имеют сложную структуру. Они могут отображаться большим количеством информационных конструкций. Для многоуровневой сетевой модели не существует никаких ограничений. Одно отношение может быть основным в любом количестве веерных отношений и одновременно зависимым в других веерных отношениях. Зависимости между отношениями могут быть в том числе и циклическими. Однако пользователю очень сложно воспринимать циклическую модель, поэтому в практических работах наибольшее распространение получили двухуровневые сетевые модели данных.

### **3.4.1. Двухуровневые сетевые модели данных**

Каждое отношение двухуровневой модели данных может находиться в одном из следующих состояний:

- вне веерного отношения;
- основное в некотором множестве веерных отношений.
- зависимое в некотором множестве веерных отношений.

Для двухуровневых сетей одно и то же отношение не может быть одновременно основным и зависимым.

Одно значение веерного отношения в памяти ЭВМ называют записью. Записи связывают друг с другом в процессе решения производственно-экономических задач с помощью служебных атрибутов информации в записи (адресов связи). Адрес связи – это служебный атрибут, в котором хранится начальный адрес записи в сетевой модели или адрес следующей обрабатываемой записи. С помощью адреса связи получают цепочки обрабатываемых записей, а в простейшем случае для веерного отношения – кольцевую цепь обрабатываемых записей, называемую веером, в которых роль ручки веера играет запись основного отношения.

Следует всегда помнить, что одно отношение не может быть одновременно основным в одном веерном отношении и зависимым в другом, о чем мы говорили выше.

Существуют дополнительные ограничения для двухуровневой сетевой модели данных:

вероятный ключ основного отношения должен быть одноатрибутным;

веерное отношение существует, если ключ основного отношения является частью ключа зависимого отношения.

Пример простой сетевой модели данных дан на рис. 3.23. Запись обозначена прямоугольником, адрес связи – стрелкой.

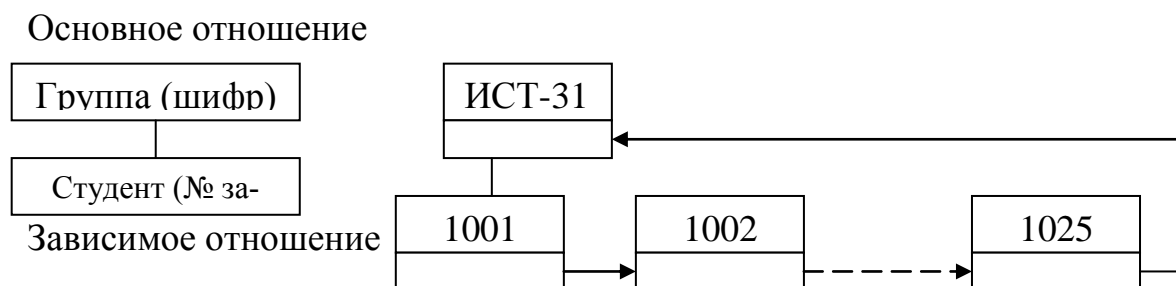


Рис. 3.23

Пример сетевая модели данных из двух веерных отношений приведен на рис. 3.24.

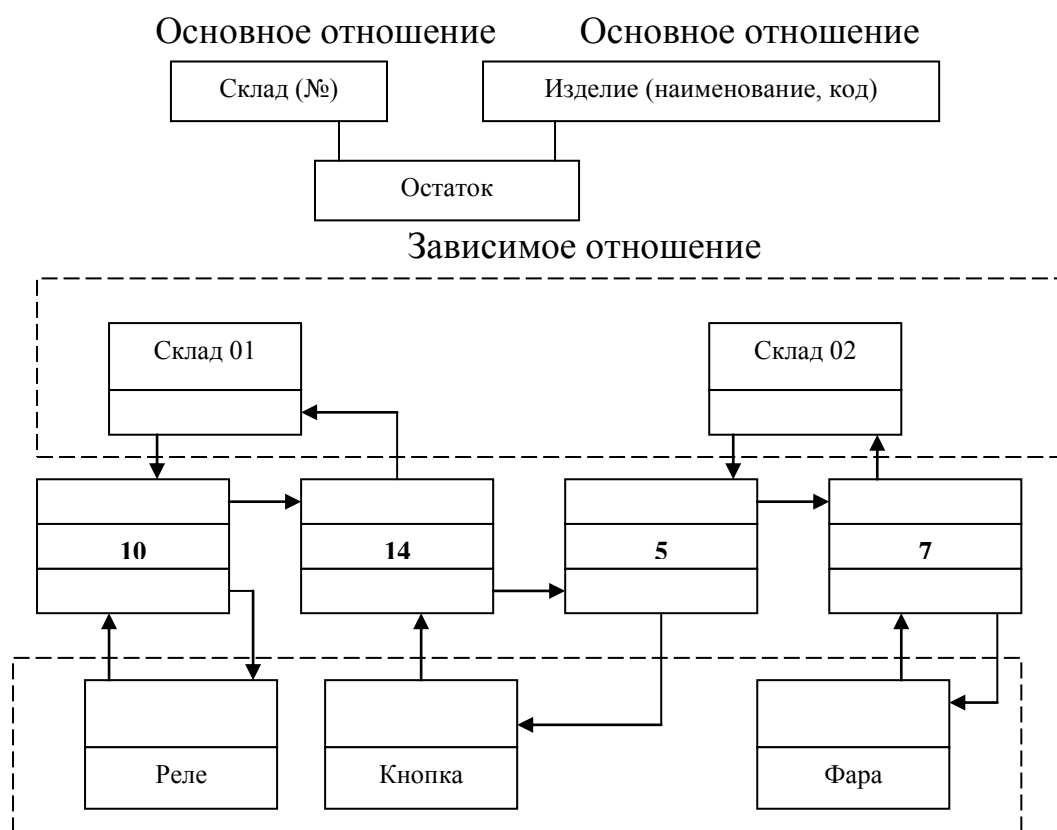


Рис. 3.24

Каждое отношение, которое используется в сетевой модели, имеет вероятные ключи отношений. Они классифицируются на полные и частич-

ные, внешние и внутренние. Полный ключ располагается в зависимом отношении, а частичный – в основном. Часть вероятного ключа, расположенного вне данного отношения (в отношениях более высокого уровня), называется внешним ключом, а внутри – внутренним.

Операции, которые проводятся с сетевой моделью данных, – это различные виды выборки информации.

Выборка информации может осуществляться двумя способами:

с помощью базового языка СУБД;

включающего языка.

В качестве включающего языка используется любой язык программирования, на котором пишется процедура выборки информации в сетевой базе данных. Все базы данных классифицируются в зависимости от модели данных.

Существуют три класса сетевых баз данных:

неиерархическая (все связи неиерархические, к любому отношению существует путь от другого отношения);

иерархическая (частный случай сетевой базы данных; эти базы имеют один корень, у них внутренний и полный ключи совпадают, связи только иерархические);

смешанная база данных (пути от одного отношения к другому могут проходить по любым связям).

### ***3.4.2. Способ поиска информации в сетевой модели данных***

Навигационный способ поиска данных – единственный способ поиска данных в сетевой модели. Основное понятие – «текущая запись». Текущая запись – это значение той записи, на которой была остановлена операция по обработке данных; через эту текущую запись, которая должна относиться к основному типу отношений, происходит поиск информации. Возможны два подхода к обработке данных: применяется базовый язык внутри конкретной СУБД или используется включающий язык применяемого средства программирования.

При навигационном способе доступа обращение ведется через текущую запись в отношении базы данных. Текущая запись после выполнения некоторой операции является значением отношения, на котором операция завершилась. Рассмотрим пример навигационного способа доступа к информации (сотрудник – зарплата) (рис. 3.25). Вначале выполняется первая операция: в качестве текущей устанавливается запись со значением первичного ключа «Иванов». Вторая операция – получить запись в зависимом отношении для сотрудника «Иванов», а именно о зарплате (основной или дополнительной на заданную дату).

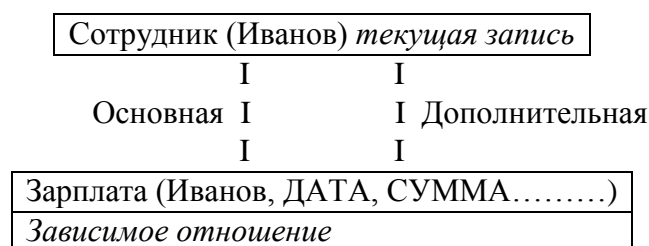


Рис. 3.25

### 3.5. Иерархическая модель данных

Исторически иерархическая модель данных в информационных технологиях появилась первой. Это связано с тем, что структура производственно-экономических документов подчинена принципам иерархического дерева.

Иерархическая модель данных – это множество веерных отношений и всех прочих, имеющих ограничения:

в иерархической модели данных имеется одно отношение, которое не является зависимым ни в одном веерном отношении. Это отношение носит название «Корень дерева»;

любое другое отношение, кроме корневого, является зависимым только в одном веерном отношении.

Вершины на графе иерархической БД соответствуют отношениям, которые называются терминальными.

Пример иерархической модели данных: корневое отношение – «Факультет», левая ветвь дерева – «Группа – Студент», правая ветвь дерева – «Кафедра – Преподаватель» (рис. 3.26). В прямоугольниках находятся ключи соответствующих отношений.

Рассмотрим укладку значений отношений иерархической базы данных в памяти ЭВМ. Информацию организуют в линейную последовательность значений, благодаря чему достигается минимальный расход памяти ЭВМ:

Фак =	Гр =	Ст =	Ст =	...	Гр =	Ст =	Ст =	...	Каф
= ФИТ	= ПИ-11	= 91001	= 91002		= ИС-32	= 9111	= 9112		= ИС

Фак = ФИТ составляет первую запись в памяти ЭВМ, Фак = ХТФ – вторую и т. д. Таким образом последовательно заполняют записи для каждого факультета.

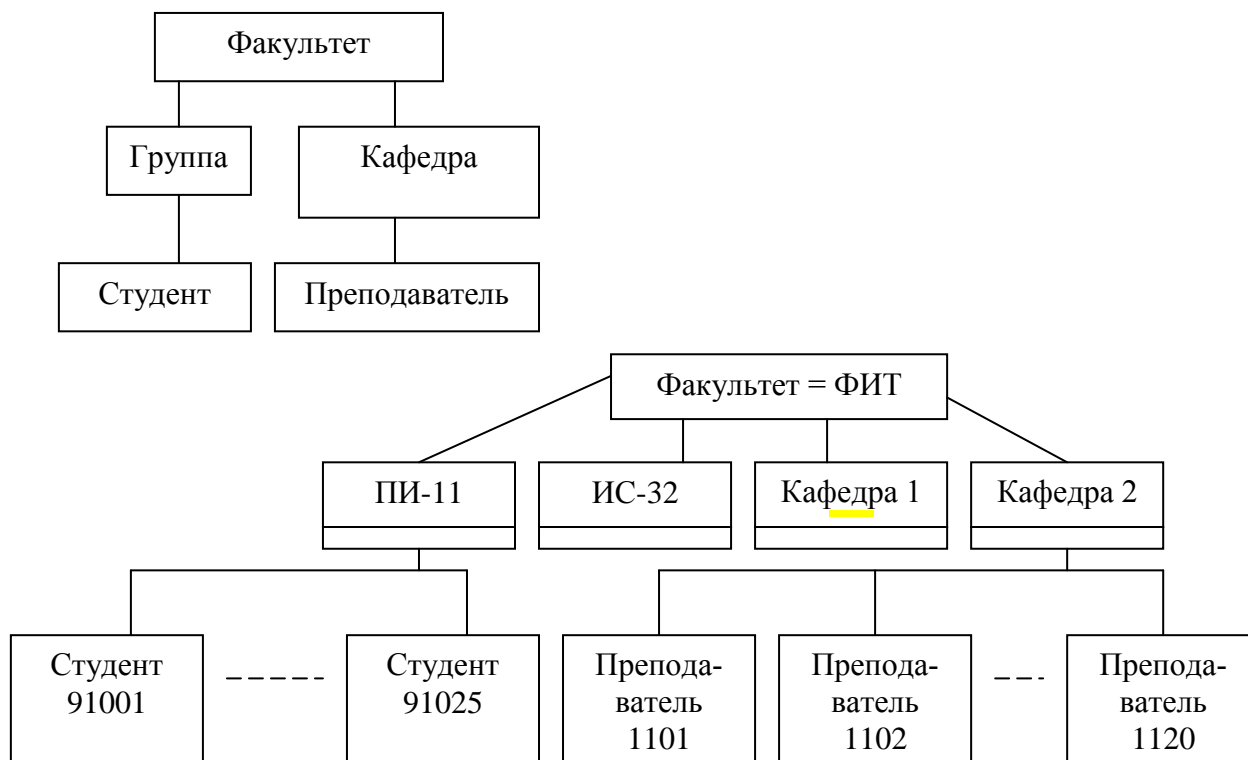


Рис. 3.26

В иерархической базе данных можно осуществлять только операции выборки (самые различные) с использованием включающего языка. В процессе выборки поиск данных осуществляется через корневое отношение.

### 3.6. Преимущества и недостатки различных моделей данных

Рассмотренные модели данных имеют присущие каждой из них достоинства и недостатки, поэтому выбирать модель данных необходимо с учетом каждой конкретной задачи, характеристик входной информации и базового программного обеспечения, порядка использования результатов решения, структуры технических средств.

Преимущества реляционной модели данных:

простота информационной конструкции (представляет собой таблицу);

теоретическое обоснование на основе использования процедур реляционной алгебры (методы нормализации позволяют получать базу данных с минимальной избыточностью);

независимость данных (минимальные изменения в прикладных программах при изменении структуры БД).

Недостатки реляционной модели данных:

низкое быстродействие при выполнении некоторых операций (например, соединения);

большой расход памяти ЭВМ на размещение БД.

Преимущества иерархической модели данных:

простота информационной конструкции (иерархическое дерево – структура производственно-экономического документа);

независимость данных (ошибка данных в одном атрибуте не влияет на достоверность данных в других атрибутах);

минимальный расход памяти ЭВМ.

Недостатки иерархической модели данных:

не универсальна (многие взаимосвязи данных невозможно реализовать в иерархической модели данных);

допустим только навигационный способ доступа к данным;

доступ к данным осуществляется только через корневое отношение.

Преимущества сетевой модели данных:

универсальность (предоставляет самые широкие возможности по сравнению с остальными моделями);

доступ к данным – через любое отношение основного типа.

Недостатки сетевой модели данных:

сложность сетевой модели (для каждой конкретной модели – свое множество понятий, вариантов их взаимосвязей и особенностей реализации);

допустимость только одного способа доступа к данным (навигационного).

### **3.7. База знаний, модели знаний**

Информационные системы могут использовать данные, которые в явном виде хранятся в памяти ЭВМ. Если необходимо принять какое-либо решение, то оно должно быть сформулировано и размещено в памяти ЭВМ (в явном виде хранится в БД).

Однако в большинстве случаев решение в явном виде отсутствует. Тогда использование системы управления БД результатов не даст. Но существует теория и прикладные программные системы, посредством которых можно накапливать знания, хранить их в памяти ЭВМ и из них получать новые знания. Информационные системы, которые способны накапливать знания и в процессе своего функционирования получать новые знания из старых, получили название систем искусственного интеллекта. Наибольшее распространение получили ЭС, которые предназначены для принятия решений в различных областях человеческой деятельности. Причем это решение не хранится в явном виде в памяти ЭВМ. Главным элементом интеллектуальных систем являются знания, которые хранят, используя для этого модели представления знаний в БЗ.

Любая БЗ должна содержать следующие характеристики:



описание закономерностей в предметной области, которые позволяют выводить новые факты и знания для рассматриваемого объекта;

сведения о структуре информационной системы и применяемой модели БЗ (это так называемая метайнформация, под которой понимают информацию об используемых средствах и методах обработки и хранения обрабатываемой информации);

сведения, обеспечивающие понимание входного языка, механизма перевода входных запросов во внутренний язык интеллектуальной системы;

методы логического вывода новых знаний, которые должны быть понятны пользователю.

Принятие решений может быть основано как на количественных выводах, которые хранятся в БД, так и на основе знаний, которые хранятся в БЗ и позволяют производить качественные рассуждения.

Существуют три основные модели знаний:

продукционная;

использующая фреймы;

семантическая сеть.

**Продукционная**, основанная на продукционных правилах, состоит из трех компонентов:

набора продукционных правил, с помощью которых описываются закономерности рассматриваемых объектов (это БЗ);

исходных фактов о рассматриваемом объекте и результатов выводов, полученных из этих фактов;

механизмов логического вывода новых знаний (программных средств, преобразующих внутренний язык описаний закономерностей и фактов в новые факты, которые можно использовать во внешнем языке представления знаний).

Каждое правило имеет условную и заключительную части, например «Если (условие), то (действие)». Под условием понимается некоторое предложение – образец, по которому осуществляется поиск в БЗ. Заключительная часть – это действия, осуществляемые в случае успешного выполнения условия (успешного исхода поиска). Действия могут быть промежуточными, выступающими далее как условия, и терминальными (целевыми), завершающими работу системы. Примеры промежуточных и терминальных действий показаны на рис. 3.27. Если  $\Phi_1 \wedge \Phi_2 \vee \Phi_3$ , то  $\Phi_k$ . Если срок окупаемости  $T_{ок} > 3$  лет  $\wedge$  Прибыль  $< 1000$  тысяч рублей, то этот проект является нерентабельным.  $\Phi_i$  – это факты или действия; «И», «ИЛИ» – логические процедуры.

Существует два типа продукционных систем – с прямым и обратным выводом. Прямой логический вывод реализует стратегию от фактов к заключению или от данных к поиску цели (дерево анализируется снизу

вверх). Если  $\Phi 4 \& \Phi 5$ , то  $\Phi 1$ ; если  $\Phi 6 \wedge \Phi 7$ , то  $\Phi 3$ ; если  $\Phi 1 \wedge \Phi 2 \wedge \Phi 3$ , то  $\Pi$ . При обратном выводе выдвигаются гипотезы, которые могут быть подтверждены или опровергнуты на основании фактов, поступающих в рабочую память. Обратный вывод целей предназначен для подтверждения истинности факта, отсутствующего в рабочей памяти. В случае обратного вывода условием останова системы является окончание списка правил, которые относятся к доказываемым целям. Имеется цель – нужно определить подтверждающие факты (дерево анализируется сверху вниз). При прямом выводе останов происходит по окончании списка применимых правил. Существуют программные оболочки, которые позволяют анализировать эти правила (*GURU*, *EXSYS* и другие экспертные системы).

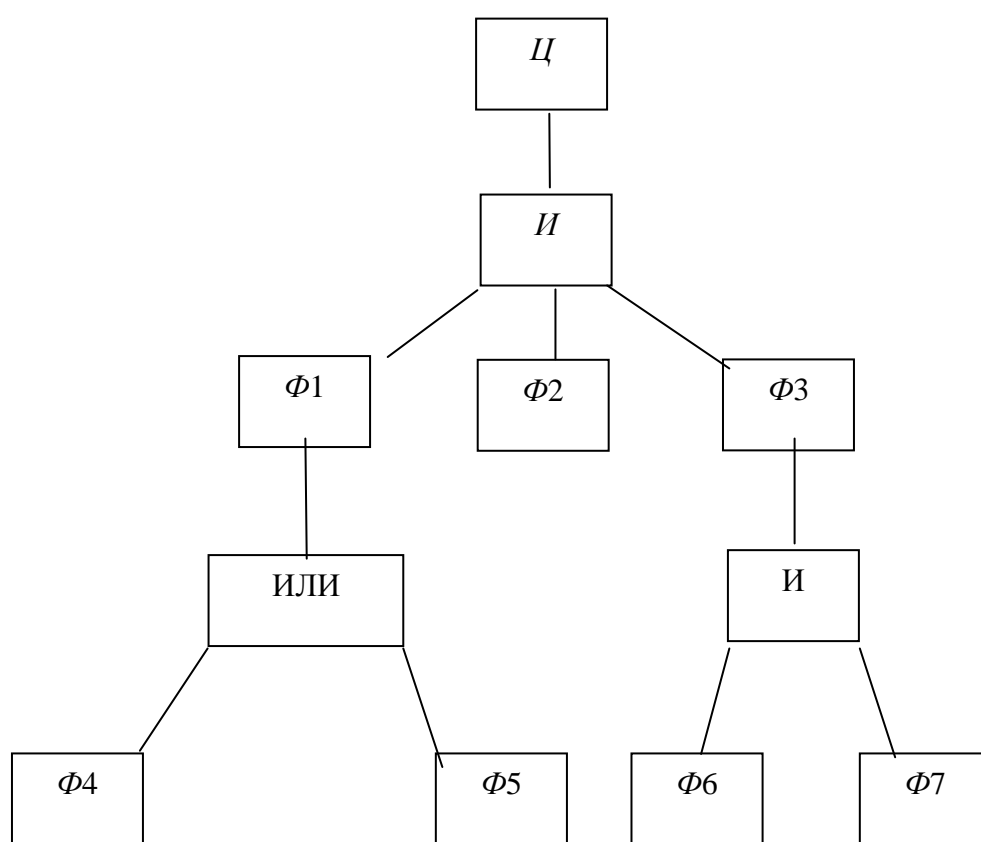


Рис. 3.27

**Фреймовая модель:** в основе теории фреймов лежит фиксация знаний путем сопоставления новых фактов с рамками, определенными для каждого объекта в виде отдельных фреймов. Сущность модели – в том, что какой-либо факт или знание сравнивается с рамками тех знаний, которые уже имеется. Эта модель во многом аналогична человеческому мышлению. В основе модели лежит таблица фрейма, строки которой называются слотом и являются отдельной единицей знаний об объекте, которому отведен данный фрейм (табл. 3.1).

Таблица 3.1

Таблица фрейма

Наименование слота	Указатель наследования	Указатель типа	Значение слота
Слот 1			
Слот 2			
...			
Слот $N$			

Слот есть предназначенный для фиксации знаний об объекте элемент данных, которому отведен данный фрейм. Указатель наследования показывает, какую информацию об атрибутах слотов из фреймов более верхнего уровня наследуют слоты с теми же именами во фрейме нижнего уровня (например, та же информация, уникальная, независимая). Указатель типа данных определяет характер используемой информации, которая хранится в каждом слоте: текстовая, числовая, программная процедура, исполнимые программные модули и др.

Фреймы обычно организуются в виде иерархического дерева (рис. 3.28):

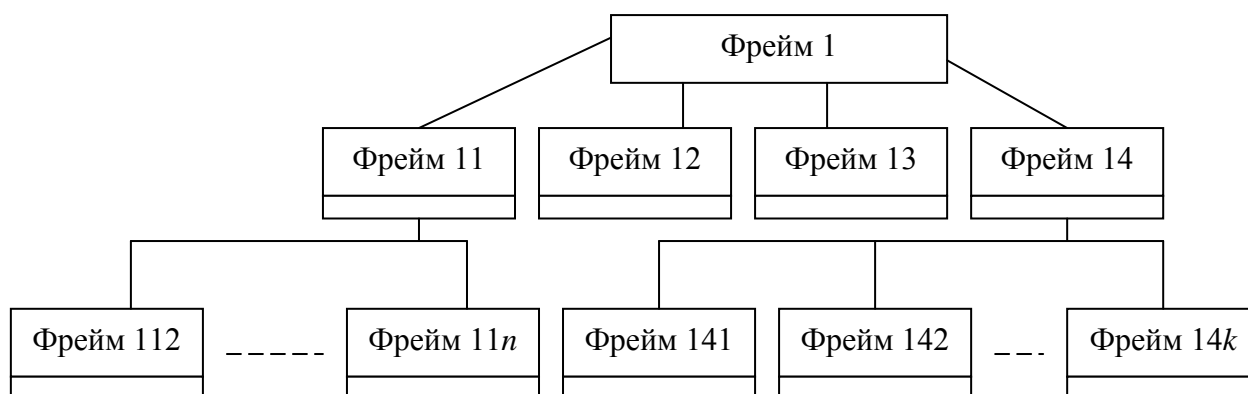
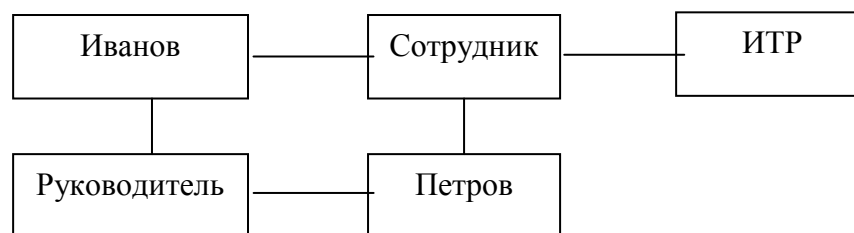


Рис. 3.28

**Семантические сети** (рис. 3.29) представляют объекты и отношения между объектами, которые характеризуют заданный объект и взаимосвязь этого объекта с другими и процессами их взаимосвязи; отображают смысловые характеристики хранимой информации. Особенность этих сетей: модель знаний и запрос к этой модели имеют одинаковую структуру. На основании вопроса к БЗ строится семантическая сеть, отображающая структуру вопроса. Ответ получается путем простого наложения семантической сети запроса на семантическую сеть модели знаний.



Сеть логического вывода показана на рис. 3.30.



Рис. 3.30

На схеме (см. рис. 3.30) в виде прямоугольников отображены объекты или понятия, а связи или стрелки характеризуют отношения между объектами или понятиями, которые могут быть однородными, т. е. с одним типом отношений, неоднородными, т. е. с различными типами отношений. Типы отношений вводит разработчик семантической модели знаний.

### 3.8. Алгоритм нормализации к 3НФ

Исходные данные – множество всех реквизитов базы данных. Метод – создание отношений, в которых соблюдается одна функциональная зависимость либо ни одной.

Для реализации метода необходимо выполнить ряд шагов.

Шаг 1 – получить исходное множество функциональных зависимостей для реквизитов рассматриваемой базы данных.

Если исходные функциональные зависимости не удастся определить путем анализа содержательных ролей реквизитов, придется использовать перечисление и отбраковку допустимых вариантов функциональных зависимостей. Для определения исходных функциональных зависимостей рассматриваются все сочетания по два реквизита и в каждом случае доказывается или отвергается функциональная зависимость. Затем рассматриваются сочетания:

по три реквизита, где первые два могут функционально определять третий;

четыре реквизита, где первые три могут функционально определять четвертый и т. д.

Применение теорем о функциональных зависимостях позволяет сократить количество рассматриваемых вариантов. Перечисление вариантов заканчивается, когда сочетания реквизитов станут содержать первичный ключ.

Шаг 2 – получить минимальное покрытие множества функциональных зависимостей. В минимальном покрытии должны отсутствовать зависимости, которые являются следствием оставшихся зависимостей по теоремам  $T1-T6$ . В частности, требуется объединить функциональные зависимости с одинаковой левой частью в одну зависимость. Обозначим полученное минимальное покрытие функциональных зависимостей через  $F = \{f_1, \dots, f_i, \dots, f_k\}$ .

Шаг 3 – определить первичный ключ отношения.

Шаг 4 – для каждой функциональной зависимости  $f_i$  создать проекцию исходного отношения  $R_i = R[X_i]$ , где  $X_i$  – объединение реквизитов из левой и правой частей  $f_i$ .

Шаг 5. Если первичный ключ исходного отношения не вошел полностью ни в одну проекцию, полученную на шаге 4, необходимо создать отдельное отношение из реквизитов ключа.

Для практического применения алгоритма нормализации до 3НФ необходимо решить два вопроса:

«как учесть наличие взаимно-однозначных соответствий?»;

«как сократить объем перебора вариантов при первоначальном определении множества функциональных зависимостей?».

Для взаимно-однозначных соответствий принято выделение старшего (по объему представляемого понятия) реквизита, который затем представляет все реквизиты взаимно-однозначного соответствия.

Для ускорения шага 1 алгоритма нормализации не рассматриваются такие зависимости, которые являются следствием уже найденных зависимостей и теорем  $T1-T6$ , и используется ряд закономерностей об отсутствии функциональных зависимостей. Среди них теорема:

«Если  $A \rightarrow B$  и  $D \not\rightarrow C$ , то  $ABD \not\rightarrow C$ »

и правило, согласно которому функциональные зависимости с реквизитом-основанием в левой части не требуется рассматривать (в противном случае алгоритм станет создавать отношения, состоящие только из реквизитов-оснований и не имеющие экономического смысла).

## ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПРИ ИЗУЧЕНИИ ГЛАВЫ 3

**Задание 3.1.** По данным табл. 3.1 выполните действия:  
 убедитесь, что табл. 3.1 удовлетворяет требованиям к нормализованному файлу;  
 укажите схему и экземпляр файла;  
 для каждого из реквизитов сформулируйте описание или точное определение его домена;  
 ответьте, какие факты, объекты или события отражают строки таблицы;  
 в пустой строке (1) дайте буквенное обозначение каждому из реквизитов с учетом того, основание это или признак;  
 реализуйте табл. 3.2 средствами СУБД, укажите машинные средства задания характеристик табл. 3.2.

Таблица 3.2

Файл «Бакалея»

Номер строки	Код товара	Вид товара	Наименование товара	Расфасовка
1				
2	10100	Чай	Баловень	50 пакетов
3	10150	Чай	Баловень	100 пакетов
4	10200	Чай	Принцесса Нури	Гранулы
5	10300	Чай	Принцесса Гита	50 г
6	10400	Чай	Дилма	100 г
7	10500	Чай	Ахмад	100 г
8	10600	Чай	3 слона	50 г
9	10700	Чай	Пиквик	50 пакетов
10	10800	Чай	Брук Бонд	100 г
11	20090	Кофе	Nescafe	50 г
12	20100	Кофе	Nescafe	100 г
13	20200	Кофе	Nescafe	250 г
14	20300	Кофе	Jacobs	50 г
15	20400	Кофе	Jacobs	100 г
16	20700	Кофе	Pele	100 г
17	20800	Кофе	Monterey	100 г
18	50064	Джем	Белорусский	250 г
19	50300	Конфитюр	Нижегородский	300 г
20	50400	Повидло	Арзамасское	500 г
21	50500	Шоколад	Российский	100 г
22	50900	Сливки	Украинские	5 г

**Задание 3.2.** Получите (средствами СУБД) проекцию файла  $F0$  (см. табл. 3.3) на столбцы  $D$  и  $E$ . Реализуйте рассмотренную проекцию  $SQL$ -запросом средствами СУБД Access.

Таблица 3.3

Файл  $F0$ 

$A$	$B$	$C$	$D$	$E$
$a1$	$b1$	$c1$	$d1$	$e1$
$a2$	$b2$	$c2$	$d2$	$e2$
$a3$	$b3$	$c3$	$d3$	$e3$
$a4$	$b4$	$c4$	$d4$	$e4$

**Задание 3.3.** Пусть имеется файл  $F1$  со схемой  $CX(F1) - \langle A, B \rangle$  (бинарное отношение) и экземпляром

$$EX(F1) = \{ \langle a1, b1 \rangle; \langle a2, b2 \rangle; \langle a3, b3 \rangle; \langle a4, 1 \rangle; \langle a5, b3 \rangle \}.$$

Постройте проекции  $F2 = F1[A]$  и  $F3 = F1[B]$ . Получите результат с помощью  $SQL$ -запроса.

**Задание 3.4.** Постройте в СУБД Access два варианта проекции файла  $F0$  (см. табл. 3.3):

- а) все проекции на один реквизит;
- б) все проекции на три реквизита.

**Задание 3.5.** Постройте в СУБД Access бинарные проекции файла  $F0$  (см. табл. 3.3), имеющие:

- а) двухреквизитный ключ;
- б) четыре строки;
- в) три строки.

Докажите, что этот файл не имеет более коротких проекций.

**Задание 3.6.** Пусть имеется запрос  $RELF0 = F0 [E = \langle e2 \rangle]$ . Файл  $F0$  показан в табл. 3.3. Назовите этот запрос ВКАЕЗ. Отобразите его средствами языка  $SQL$  в таблицу. Сделайте это в нескольких вариантах, меняя набор столбцов в операторе  $SELECT$ . В необходимых случаях добавьте операцию  $GROUP BY$ . Реализуйте запрос ВКАЕЗ экранными средствами СУБД Access.

**Задание 3.7.** Реализуйте операции поиска в файле «Бакалея» (см. табл. 3.2) и  $F0$  (см. табл. 3.3) таким образом, чтобы результирующие файлы содержали: а) одну запись; б) две записи.

Реализуйте различные операции поиска в файле «Бакалея» и  $F0$ , варьируя поисковые условия, проекции и способы задания параметров поиска.

**Задание 3.8.** Пусть имеется файл  $F1$  со схемой  $CX(F1) = (A, B, C)$  и экземпляром  $EX(F1) = \{ \langle a1, b1, c1 \rangle, \langle a2, b1, c2 \rangle \}$ .

Выполните проекцию на подсхемы:

$$CX(F11) = (A, B);$$

$$CX(F12) = (D, C).$$

Выполните естественное соединение  $F2 = F11 \cdot F12$  и убедитесь, что  $F1 \neq F2$ , Почему это произошло?

**Задание 3.9.** Пусть имеются отношения  $T1$  и  $T2$ :

$T1$		
$A$	$B$	$C$
$a1$	8	4
$a2$	1	3
$a3$	6	5
$a1$	2	4

$T2$	
$A$	$D$
$a1$	5
$a2$	4
$a3$	2

Выполните соединения:

- 1)  $T3 = T1[B < D]T2$ ;
- 2)  $T4 = T1[C > D]T2$ .

**Задание 3.10.** Получите естественные соединения файлов  $F1$  (см. задание 3.8) и  $F0$  (см. табл. 3.3), а также ранее полученных отдельных проекций этих файлов. Проанализируйте случаи естественного соединения без потерь, ловушки связей, декартова произведения, построив соответствующие запросы и таблицы средствами СУБД Access.

**Задание 3.11.** Реализуйте *SQL*-запрос

```
SELECT ABCDE.A, ABCDE.B, Sum(ABCDE.G) AS Sum_G INTO
ProjectionABGSum
FROM ABCDE
GROUP BY ABCDE.A, ABCDE.B;
```

для таблицы  $ABCDE$  (табл. 3.4). Постройте также этот запрос средствами экранного сопряжения СУБД Access.

Таблица 3.4

Таблица $ABCDE$							
$A$	$B$	$C$	$D$	$E$	$F$	$G$	$H$
$a1$	$b1$	$c1$	$d1$	$e2$	$f1$	27	32
$a2$	$b2$	$c2$	$d1$	$e2$	$f1$	44	18
$a3$	$b3$	$c3$	$d2$	$e5$	$f1$	17	11
$a4$	$b1$	$c4$	$d3$	$e3$	$f1$	56	61
$a5$	$b3$	$c3$	$d3$	$e3$	$f1$	43	14
$a6$	$b4$	$c2$	$d7$	$e6$	$f6$	13	77



**Задание 3.12.** Для создания таблицы *BFSumH* нужно реализовать запрос к таблице *ABCDE*

```
SELECT ABCDE.B, ABCDE.F, Sum(ABCDE.H) AS Sum_H  
FROM ABCDE INTO BFSumH  
GROUP BY ABCDE.B, ABCDE.F;
```

Реализуйте SQL-запрос

```
SELECT ProjectionABGSum.A, BFSumH.B, BFSumH.F INTO JoinB  
FROM BFSumH INNER JOIN ProjectionABGSum ON BFSumH.B =  
ProjectionABGSum.B;
```

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

**Задание 3.13.** Реализуйте SQL-запрос

```
SELECT ABCDE.A, ABCDE.B, ABCDE.F, ABCDE.E FROM ABCDE  
WHERE (((ABCDE.E)=«e2» Or (ABCDE.E)=«e3»));
```

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

**Задание 3.14.** Реализуйте SQL-запрос

```
SELECT ABCDE.A, ABCDE.B, Sum(ABCDE.H) AS Sum^H FROM  
ABCDE INTO ABSumH  
GROUP BY ABCDE.B;
```

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

**Задание 3.15.** Реализуйте SQL-запрос

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.B, ABCDE.F, Sum(ABCDE.H) AS Sum_H  
FROM ABCDE  
GROUP BY ABCDE.B;
```

**Задание 3.16.** Реализуйте SQL-запрос

```
SELECT ProjectionABGSum.A, ABSumH.B, ABSumH.A INTO JoinA  
FROM ABSumH INNER JOIN ProjectionABGSum ON ABSumH.A =  
ProjectionABGSum.A;
```

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

**Задание 3.17.** Реализуйте SQL-запрос

```
SELECT ABCDE.B, ABCDE.F, Avg(ABCDE.H) AS Avg_H FROM  
ABCDE  
GROUP BY ABCDE.B, ABCDE.F;
```

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

**Задание 3.18.** Реализуйте *SQL*-запрос  
*SELECT ABCDE.A, ABCDE.B, ABCDE.F, ABCDE.E*  
*FROM ABCDE*  
*WHERE (((ABCDE.E)=«e3» Or (ABCDE.F)=«f1»));*

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

**Задание 3.19.** Реализуйте *SQL*-запрос  
*SELECT ABCDE.A, ABCDE.B, ABCDE.F, ABCDE.E*  
*FROM ABCDE*  
*WHERE (((ABCDE.E)=«e3» And (ABCDE.F)=«f1»));*

Постройте также этот запрос средствами экранного сопряжения СУБД Access.

**Задание 3.20.** Рассмотрите отношение

W2		
Продукция	Цена продукции	Комплекующее изделие
Эдв-12	40	Вк-15
Эдв-12	40	Р-20
Эдв-30	20	Вк-15
Эдв-30	20	Р-20
ЗВИ	120	Р-20

Постройте не менее двух вариантов *SQL*-запросов, результатом которых служили бы двухреквизитные проекции отношения W2. Укажите функциональные зависимости отношения W2, вероятный ключ отношения W2,

**Задание 3.22.** Предложите алгоритм проверки наличия функциональной зависимости для двухреквизитного отношения. Реализуйте этот алгоритм средствами известного вам языка программирования.

**Задание 3.23.** Докажите или опровергните наличие функциональных зависимостей:

- 1)  $DIN \rightarrow KROL$ ;
- 2)  $IMH \rightarrow KROL$ ;
- 3)  $FOUN \rightarrow IMH$ ;
- 4)  $KROL \rightarrow FOUN$ ;
- 5)  $DOT, KROL \rightarrow DIN$

в отношении

<i>KROL</i>	<i>FOUN</i>	<i>DDIN</i>	<i>DDOT</i>	<i>IMH</i>
54	32	<i>LLO</i>	<i>KKU</i>	<i>VV3</i>
34	78	<i>OOL</i>	<i>VVD</i>	<i>ZZ2</i>
45	27	<i>KKE</i>	<i>KKE</i>	<i>RR9</i>

94	52	<i>LLO</i>	<i>VVP</i>	<i>PP3</i>
54	65	<i>KKE</i>	<i>WWA</i>	<i>EE7</i>
45	32	<i>OOL</i>	<i>AAD</i>	<i>VV3</i>
34	78	<i>KKE</i>	<i>WWA</i>	<i>ZZ2</i>
94	85	<i>OOL</i>	<i>KKI</i>	<i>AA8</i>

**Задание 3.24.** Докажите или опровергните наличие функциональных зависимостей:

- 1)  $POR \rightarrow KOV$ ;
- 2)  $CHEK \rightarrow POR$ ;
- 3)  $TRIG \rightarrow CHEK$ ;
- 4)  $POR, CHEK \rightarrow KOV$ ;
- 5)  $SKB, KOV \rightarrow TRIG$

в отношении

*T6*

<i>POR</i>	<i>TRIG</i>	<i>CHEK</i>	<i>KOV</i>	<i>SKB</i>
<i>RA</i>	<i>2T</i>	<i>F4</i>	<i>GN</i>	<i>BX</i>
<i>XO</i>	<i>1A</i>	<i>E7</i>	<i>NI</i>	<i>WX</i>
<i>CW</i>	<i>9X</i>	<i>G8</i>	<i>VD</i>	<i>AX</i>
<i>KP</i>	<i>5E</i>	<i>U6</i>	<i>QW</i>	<i>BXi</i>
<i>WA</i>	<i>9X</i>	<i>E7</i>	<i>EP</i>	<i>RX</i>
<i>RA</i>	<i>5E</i>	<i>U6</i>	<i>KA</i>	<i>XY</i>
<i>AV</i>	<i>1A</i>	<i>G8</i>	<i>KR</i>	<i>AX</i>
<i>WE</i>	<i>9X</i>	<i>F4</i>	<i>EP</i>	<i>BX</i>

**Задание 3.25.** Докажите или опровергните наличие функциональных зависимостей:

- 1)  $ELIT \rightarrow SMID$ ;
- 2)  $LID \rightarrow KATR$ ;
- 3)  $SMID \rightarrow INES$ ;
- 4)  $INES, ELIT \rightarrow LID$ ;
- 5)  $KATR, LID \rightarrow SMID$

в отношении

*T7*

<i>INES</i>	<i>KATR</i>	<i>LID</i>	<i>ELIT</i>	<i>SMID</i>
<i>LU</i>	01	35	<i>UW</i>	70
<i>GN</i>	02	54	<i>WA</i>	60
<i>WA</i>	03	35	<i>YW</i>	70
<i>KU</i>	01	35	<i>GO</i>	40
<i>LU</i>	02	60	<i>AR</i>	60
<i>GN</i>	03	35	<i>CE</i>	20
<i>WA</i>	01	35	<i>SS</i>	30
<i>AR</i>	02	54	<i>DP</i>	10

**Задание 3.26.** Докажите или опровергните наличие функциональных зависимостей:

- 1)  $KOS \rightarrow KOGR$ ;
- 2)  $KOGR \rightarrow WK$ ;
- 3)  $DUBL, PRIN \rightarrow WI$ ;
- 4)  $WKII \rightarrow PRIN$ ;
- 5)  $KOS, WK \rightarrow DUBL$

в отношении

*T8*

<i>KOS</i>	<i>KOGR</i>	<i>WK</i>	<i>DUBL</i>	<i>PRIN</i>
<i>AK</i>	61	<i>WA</i>	570	<i>WS</i>
<i>AT</i>	44	<i>PR</i>	751	<i>WA</i>
<i>AK</i>	20	<i>AI</i>	856	<i>RI</i>
<i>AU</i>	73	<i>WA</i>	536	<i>TX</i>
<i>AT</i>	15	<i>PR</i>	751	<i>AX</i>
<i>AK</i>	94	<i>FY</i>	456	<i>PI</i>
<i>AU</i>	30	<i>CM</i>	846	<i>CI</i>
<i>AT</i>	52	<i>VD</i>	751	<i>WS</i>

**Задание 3.27.** Укажите функциональные зависимости в отношениях:

*T0*(ФИО, отдел, руководитель отдела);

*T1*(ФИО, отдел, должность);

*T2*(отдел, проект, число исполнителей);

*T3*(электродвигатель, мощность, покупатель, купленное количество);

*T4*(преподаватель, кафедра, факультет);

*T5*(автор, статья, журнал);

*T6*(преподаватель, кафедра, дисциплина);

*T7*(автомобиль, грузоподъемность, расход топлива);

*T8*(отрасль, предприятие, бригада);

*T9*(изделие, деталь, вес изделия).

Найдите вероятные ключи.

**Задание 3.28.** Укажите следствия из множества функциональных зависимостей  $\langle A \rightarrow B, B \rightarrow C \rangle$ .

**Задание 3.29.** Укажите следствия из множества функциональных зависимостей  $\langle X \rightarrow X1, X \rightarrow X2, X \rightarrow X3, X \rightarrow X4 \rangle$

**Задание 3.30.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
ФИО вкладчика	Номер $\rightarrow$ ФИО
Номер сберкнижки (номер)	Номер, дата $\rightarrow$ расход
Дата	Номер, дата $\rightarrow$ остаток
Приход	Номер, дата $\rightarrow$ приход, расход, остаток
Расход	Номер, дата $\rightarrow$ приход
Остаток	

**Задание 3.31.** Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Цех	Цех, код_станка $\rightarrow$ кол-во
Год	Цех, код_станка, код_детали $\rightarrow$ кол-во
Код_станка	Цех, год, код_детали $\rightarrow$ план
Количество станков (кол-во)	Цех, код_станка, год $\rightarrow$ кол-во
Код_детали	
План производства деталей	

**Задание 3.32.** Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
ФИО служащего	ФИО, дата $\rightarrow$ должность
Должность	ФИО $\rightarrow$ должность
Дата	ФИО, дата $\rightarrow$ зарплата
Зарплата	ФИО, имя_ребенка $\rightarrow$ возраст
Имя_ребенка	
Возраст ребенка	

**Задание 3.33.** Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Табельный номер (таб. №)	Участок $\rightarrow$ цех
ФИО рабочего	Таб. № $\rightarrow$ цех
Цех	Таб. № $\rightarrow$ участок
Участок	Таб. №, дата $\rightarrow$ сумма
Дата	Таб. № $\rightarrow$ ФИО
Сумма зарплаты (сумма)	Таб. № $\rightarrow$ ФИО, участок

**Задание 3.34.** Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений

средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Магазин	Книга, магазин → издательство
Книга	Магазин, книга, дата → кол-во
Цена	Книга → цена
Издательство	Книга → цена, издательство
Дата	Книга → издательство
Количество продано (кол-во)	

**Задание 3.35.** Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Отправитель	Получатель, изделие → адрес
Получатель	Отправитель, получатель, изделие → кол-во
Адрес получателя (адрес)	Получатель → адрес
Изделие	Отправитель, изделие → цена
Цена	Изделие → цена
Количество на месяц (кол-во)	

**Задание 3.36.** Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Аэропорт отправления	№ рейса, дата → № самолета
Номер рейса (№ рейса)	№ самолета → кол-во
Количество мест (кол-во)	№ рейса → пункт назначения
Бортовой номер самолета	№ рейса → пункт назначения, аэропорт
Пункт назначения	№ рейса → аэропорт
Дата вылета (дата)	

**Задание 3.37.** Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
ФИО студента	ФИО → группа
Дата поступления в вуз	ФИО → факультет
Факультет	ФИО → дата
Группа	Группа → факультет
Место работы	Место работы → зарплата
Зарплата	ФИО → группа, дата

**Задание 3.38.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
№ больницы	№ больницы, пациент → № палаты
№ палаты	№ больницы, пациент → врач
Пациент	Пациент → домашний адрес
Домашний адрес	№ больницы, пациент → № палаты, врач
Лечащий врач (врач)	
Диагноз	

**Задание 3.39.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Дата матча (дата)	Дата, команда-хозяин → счет
Команда-хозяин	Дата, команда-хозяин → число очков
Команда-гость	Дата, команда-хозяин → счет, число очков
Счет матча	Дата, команда-хозяин, команда-гость → счет
Число очков в чемпионате	

**Задание 3.40.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
ФИО студента	ФИО → группа
Дата поступления в вуз (дата)	ФИО → факультет
Факультет	ФИО → дата
Группа	Группа → факультет
Общественная работа	ФИО → группа, дата

**Задание 3.41.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
ФИО служащего	ФИО → отдел
Источник финансирования темы (фин.)	ФИО → учреждение
Отдел	Тема работы → фин.
Учреждение	Фин. отдел → учреждение

**Задание 3.42.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Порт	Судно, дата отплытия → порт
Судно	Судно, дата отплытия → порт назначения
Грузоподъемность	Судно, дата отплытия → порт, порт назначения
Дата отплытия	Судно → грузоподъемность
Порт назначения	

**Задание 3.43.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Дисциплина	Время, аудитория → дисциплина
Преподаватель	Время, преподаватель, дисциплина → аудитория
время занятий	Время, преподаватель → аудитория
Аудитория	Время, ФИО → аудитория
ФИО студента (ФИО)	

**Задание 3.44.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Учреждение	ФИО → учреждение
Отдел	ФИО → отдел
Тема	Код оборудования → отдел
Код оборудования	Отдел → учреждение
ФИО сотрудника	Тема, ФИО → продолжительность
Продолжительность работы	Код оборудования → учреждение

**Задание 3.45.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

<i>Реквизиты</i>	<i>Функциональные зависимости</i>
Отдел	ФИО → отдел
ФИО сотрудника	Телефон → отдел
Номер комнаты	Телефон → номер комнаты
Телефон	Отдел → учреждение
Тема работы	Тема, ФИО → продолжительность
Продолжительность работы	Номер комнаты → отдел



**Задание 3.46.** Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
Завод	Продукция $\rightarrow$ цена
Продукция	Завод, продукция $\rightarrow$ выпуск
Комплектуемое изделие	Завод, продукция $\rightarrow$ цена
Цена продукции	
Выпуск продукции за год	

**Задание 3.47.** Рассмотрите отношение  $ABT(P1, P2, Q3)$ . Это отношение описывает городские автобусные маршруты, т. е. содержит количество строк по числу автобусных маршрутов.  $P1$  обозначает номер автобусного маршрута,  $P2$  – марка автобуса,  $Q3$  – вместимость автобуса<sup>^</sup>

ABT		
P1	P2	Q3
18	ЛАЗ-677	42
28	ЛАЗ-677	42
26	Икарус 260	56
30	ЛИА3-684	45
507	ПАЗ-672	21

Пусть  $P1$  – ключ,  $P1 \rightarrow P2$ . Рассмотрите отношение  $ABT1 = ABT[P2, Q3]$ , содержащее вместимость для каждой марки автобуса. Здесь имеет место  $P2 \rightarrow Q3$ . Операции проецирования и естественного соединения позволяют выбрать вариант файловой структуры при создании реляционной базы данных. Возможны варианты:

- 1) создаем отношение  $ABT$  со схемой  $CX(ABT) = (P1, P2, Q3)$ ;
- 2) создаем отношения  $ABT1, ABT2$  со схемами  $CX(ABT1) = (P2, Q3)$  и  $CX(ABT2) = (P1, P2)$ , соответственно являющиеся проекциями отношения  $ABT$ . При этом для получения отношения  $ABT$  необходимо использовать операцию естественного соединения отношений  $ABT1$  и  $ABT2$ .

Создайте отношение  $ABT$  средствами СУБД, запросы с помощью  $SQL$ , формирующие указанные проекции  $ABT1$  и  $ABT2$ . Выполните естественное соединение проекций  $ABT1$  и  $ABT2$ . Убедитесь в отсутствии ловушки связей. Установите номер нормальной формы у отношений  $ABT, ABT1$  и  $ABT2$ .

## ГЛАВА 4. ОРГАНИЗАЦИЯ ДАННЫХ В ПАМЯТИ ЭВМ

В памяти ЭВМ обычно осуществляют раздельное хранение значений каждой СЕИ, которое называется записью. Множество записей образуют массив или файл (обычно массив – это данные в оперативной памяти ЭВМ, а файл – данные на внешних устройствах). Под организацией значений данных понимают относительно устойчивый порядок расположения записей в памяти ЭВМ и способ обеспечения взаимосвязи между записями [8]. Организация данных может быть линейной и нелинейной.

### 4.1. Линейная организация данных в памяти ЭВМ

При линейной организации данных каждая запись, кроме первой и последней, связана с одной предыдущей и одной последующей. При нелинейной организации данных количество предыдущих и последующих записей может быть произвольным.

Среди линейных методов выделяют последовательную и цепную организацию данных. При последовательной организации данных записи располагаются в памяти строго одна за другой без промежутков в той последовательности, в которой они обрабатываются. Записи могут быть фиксированной, переменной и неопределенной длины. Записи фиксированной длины имеют одинаковую заранее известную длину. Длина записи переменной длины указывается в самой записи. В записи неопределенной длины окончание записи отмечается специальным символом – разделителем.

Адреса промежуточных записей фиксированной длины в массиве задаются формулой  $A(i) = A(1) + (i - 1) \cdot L$ , где  $A(1)$  – начальный адрес первой записи;  $A(i)$  – адрес  $i$ -й записи;  $L$  – длина одной записи.

Для массива записей переменной и неопределенной длины такой формулы не существует. Они занимают меньший объем памяти, но их обработка ведется с меньшей скоростью, так как затруднено обнаружение следующей записи.

В структуре записей последовательного массива выделяются один или несколько ключевых атрибутов, по которым происходит доступ к другим атрибутам.

Последовательный массив, состоящий из записей фиксированной длины с одним ключевым атрибутом  $P(i)$ , где  $i$  – номер записи (ключевой атрибут обычно является атрибутом признаком), выглядит так:

$P(1)$	$P(2)$	$P(3)$	.....	$P(i)$	.....	$P(m)$
--------	--------	--------	-------	--------	-------	--------

Алгоритмы обработки данных последовательного массива:

1. Формирование данных массива (сортировка данных).
2. Поиск данных в массиве.
3. Корректировка массива.
4. Последовательная обработка записей.

### **Сортировка данных**

Данные обычно возникают в неупорядоченной форме. Перед обработкой данные упорядочиваются по значениям ключевых атрибутов (это основная работа по формированию данных). Процедура упорядочения файла называется сортировкой. Отсортированные данные позволяют организовать быструю обработку нескольких массивов. Упорядочение по возрастанию значений ключевого атрибута:  $P(i) \leq P(i + 1)$ .

Массив, обладающий наибольшей неопределенностью, называется случайным массивом. Можно определить множество возможных состояний такого массива:  $M! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot M$ . Таким образом, минимальное число сравнений, необходимое для упорядочения массива из  $M$  записей:  $C = \log_2 M!$ .

После преобразований по формуле Стирлинга получаем:  $C = M \cdot (\log_2 M - 1,43)$ . Можно также записать пропорциональность:  $C \approx M \cdot \log_2 M$ . Значит, время сортировки массива обладает такой же пропорциональностью:  $T \approx M \cdot \log_2 M$ .

## **4.2. Методы поиска и корректировки при линейной организации данных**

### **4.2.1. Методы поиска: определение, виды, сравнение**

Поиском называется процедура выделения из некоторого множества записей определенного подмножества записей, которые удовлетворяют некоторому заранее поставленному условию. Простейшее условие поиска – поиск по совпадению, т. е. равенства значения ключевого атрибута  $P(i)$  и некоторого заранее заданного значения  $q$ .

Базовым методом доступа к массиву данных является ступенчатый поиск, который может быть:

- последовательным,
- двухступенчатым,
- $n$ -ступенчатым,
- бинарным.

Последовательный поиск является простейшим методом. Искомое значение ключевого атрибута  $q$  сравнивается с ключом первой записи, потом второй и т. д., пока  $q$  не станет равен или больше ключа очередной записи.

Двухступенчатый поиск осуществляется в массиве из  $M$  записей. Выбирается шаг поиска  $d1$ . Значение заданного ключа  $q$  сравнивается с последовательным рядом величин ключевых атрибутов массива  $P(1)$ ,  $P(1 + d1)$ ,  $P(1 + 2 \cdot d1)$ ,  $P(1 + 3 \cdot d1)$ , ...,  $P(1 + k \cdot d1)$ , ... до тех пор, пока не будет достигнуто неравенство  $P(1 + m \cdot d1) \geq q$ . Здесь заканчивается первая ступень поиска. На второй ступени  $q$  последовательно сравнивается со

всеми ключами, которые имеют номер  $1 + m \cdot d1$  и меньше до тех пор, пока в процессе сравнения не будет достигнут ключ, равный  $q$ .

При  $n$ -ступенчатом поиске в массиве из  $M$  записей заранее выбирают две константы:  $n$  – количество ступеней;  $S$  – число шагов на одной ступени поиска.

На первом этапе ключевые атрибуты для сравнения с искомым ключом  $q$  выбираются из массива, начиная с  $P(1)$ , с шагом  $d1 = M/S$  (округление в меньшую сторону). Когда будет выполнено неравенство  $P(1 + k \cdot d1) > q$ , назначают шаг  $d2 = d1/S$  и организуется сравнение с этим шагом, начиная с  $P(1 + k \cdot d1)$  в сторону уменьшения. Описанные действия повторяются  $n$  раз, причем шаг последней ступени поиска  $dn = 1$ .

Если  $S = 2$ , то такой поиск называется бинарным (это частный случай  $n$ -ступенчатого поиска). Для осуществления бинарного поиска вводятся понятия: левая граница интервала поиска  $A$  и правая граница интервала поиска  $B$ . Сначала интервал поиска охватывает весь массив, т. е.  $A = 0$ ,  $B = M + 1$ . Вычисляется середина интервала  $i$  по формуле  $i = (A + B)/2$  с округлением в меньшую сторону.

Ключ  $i$ -й записи  $P(i)$  сравнивают с  $q$ . Если  $P(i) = q$ , то поиск заканчивается, если  $P(i) > q$ , то записи  $i + 1, i + 2, \dots, M$  не содержат ключа  $q$ . Сокращаем наполовину интервал поиска, приняв  $B = i$ . Аналогично при  $P(i) < q$  необходимо принять  $A = i$ . Далее середину интервала вычисляют заново и все действия повторяют. Если будет достигнут нулевой интервал, то требуемой записи в массиве нет. В заключение сравним времена поиска данных для трех основных методов:

для последовательного метода поиска время поиска пропорционально количеству записей в массиве:  $T1 \approx M$ ;

метода  $n$ -ступенчатого поиска:  $T2 \approx \sqrt[n]{M}$ ;

бинарного метода поиска:  $T3 \approx \log_2 M$ .

Отсюда можно сделать вывод о том, что при достаточно большом числе записей  $M$  в массиве данных бинарный метод поиска выполняется быстрее.

#### **4.2.2. Методы корректировки данных**

Корректировка массива данных может касаться одной записи или группы записей. Отдельная запись может быть включена в массив данных и исключена из него. Кроме того, в запись могут быть внесены изменения значений отдельных атрибутов. Перед корректировкой в начале должен быть выполнен поиск корректируемой записи. Новая запись не может сразу занять место, поэтому необходимо сначала выполнить пересылку записей, чтобы освободить его. Пересылка начинается с последней записи (она опускается на одно место вниз), затем предпоследняя занимает место последней и т. д., пока не освободится место для новой записи. При исключении записей также нужна пересылка записей. Поэтому включение и исключение в последовательном массиве данных – длительная процедура и

осуществлять ее необходимо по мере накопления изменений, переносимых определенным массивом, а не по одной записи.

### 4.3. Списковая (цепная) организация данных

Цепная организация данных относится к классу линейных способов организации данных. Списком называется множество записей, занимающих произвольные участки в памяти ЭВМ. Последовательность обработки этих участков задается с помощью адресов связи. При цепной (списковой) организации записи связывают в цепи при решении производственно-экономических задач. Связывание отдельных записей цепи осуществляется с помощью служебных атрибутов записи, которые называют адресами связи. Адрес связи (служебный атрибут) указывает физический адрес начальной записи или адрес последующей обрабатываемой в цепи записи. В списке выделяется собственная информация (записи с содержательными сведениями для решения производственно-экономических задач) и ассоциативная (служебная) информация (в ней хранятся адреса связей).

Рассмотрим варианты списковой организации данных.

**Совместное хранение служебной и производственно-экономической информации** показано на рис. 4.1. Чтобы войти в цепь, следует иметь указатель цепи или указатель списка (главную запись), в котором будет указан физический адрес первой обрабатываемой записи, первая запись имеет адрес связи со второй записью и т. д.,  $\emptyset$  – конец списка записей, 1, 2, ...,  $M$  – производственно-экономическая информация.

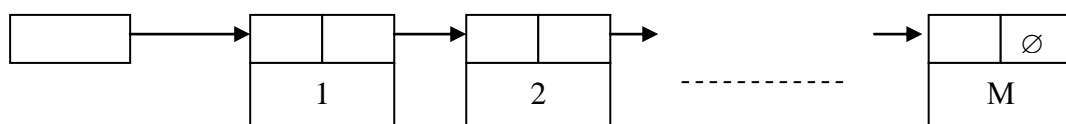


Рис. 4.1

**Раздельное хранение записей и адресов связи** иллюстрирует рис. 4.2.

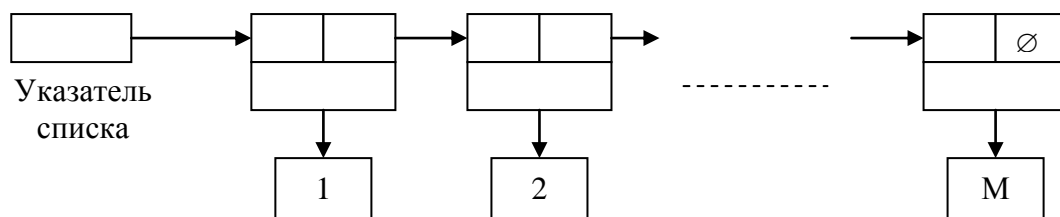


Рис. 4.2

Обычно формируют сначала неупорядоченный список, а затем его сортируют. Время формирования упорядоченного списка  $T \approx \log 2M$ . Для

поиска данных в однонаправленном списке используется один метод – последовательный поиск, время которого  $T \approx M$ . Для ускорения доступа к списку обрабатываемых записей в цепи используют двунаправленный (рис. 4.3) и кольцевой списки (рис. 4.4):

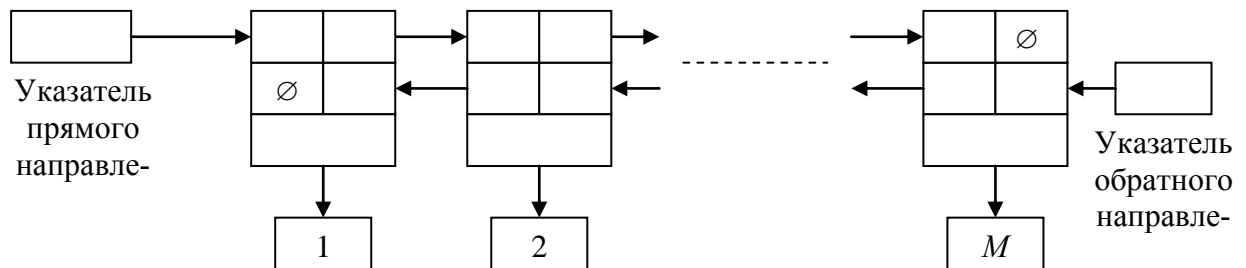


Рис. 4.3

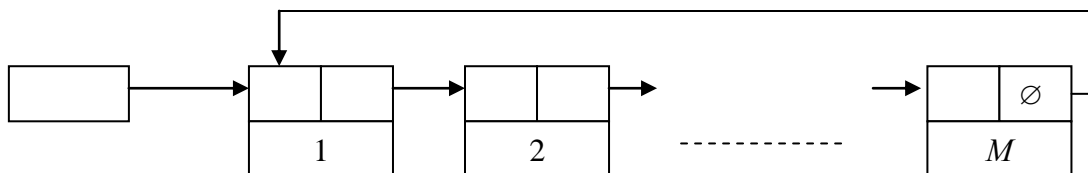


Рис. 4.4

Адреса связи бывают разных видов:

*NEXT* (указывают на следующую обрабатываемую запись  $N$ ; рис. 4.5);

*PRIOR* (указывают на предыдущую запись  $P$ ;

*MASTER* (указывают на указатель списка (главную запись)  $M$ ).

Аи на рис. 4.5 – атрибут идентификации.

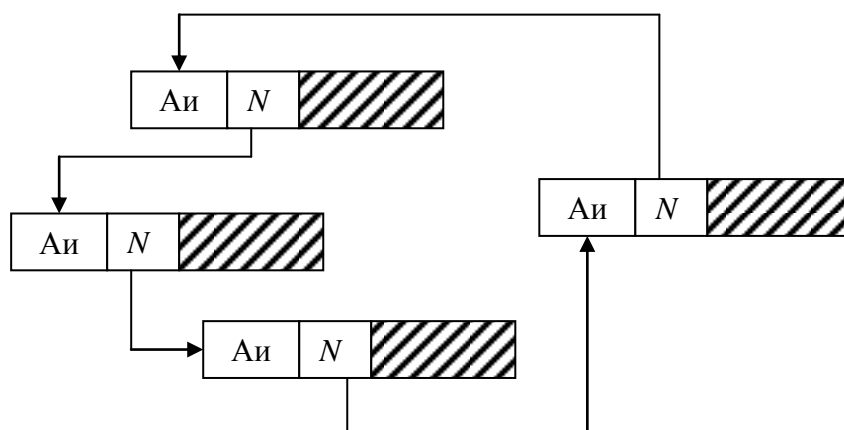


Рис. 4.5

*NEXT* и *MASTER* показаны на рис. 4.6.

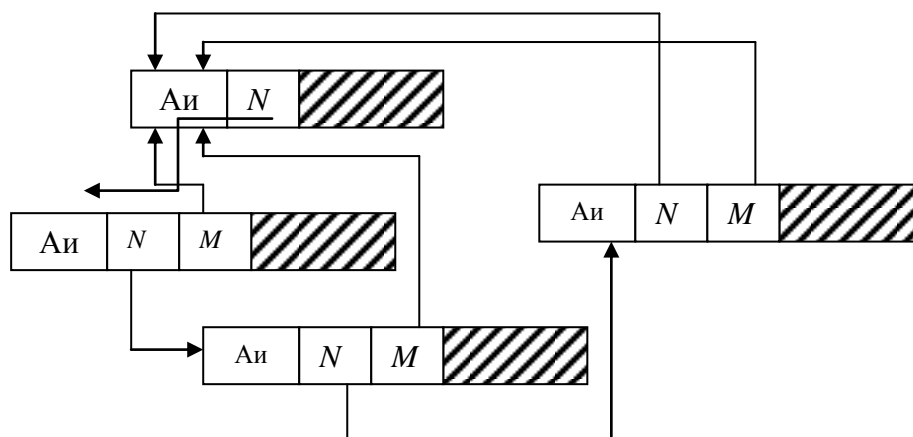


Рис. 4.6

В оперативной памяти используются специальные рабочие таблицы – таблицы цепи (рис. 4.7), имеющие поля:

*MASTER* – главная запись;

*NEXT* – следующая;

*CURRENT* – текущая;

*PRIOR* – предыдущая.

<i>MASTER</i>	0001
<i>NEXT</i>	0002
<i>CURRENT</i>	0001
<i>PRIOR</i>	0000

<i>MASTER</i>	0001
<i>NEXT</i>	0003
<i>CURRENT</i>	0002
<i>PRIOR</i>	0001

<i>MASTER</i>	0001
<i>NEXT</i>	0004
<i>CURRENT</i>	0003
<i>PRIOR</i>	0002

<i>MASTER</i>	0001
<i>NEXT</i>	0001
<i>CURRENT</i>	0004
<i>PRIOR</i>	0003

Рис. 4.7

#### 4.4. Древовидная организация данных

Древовидной организацией данных (деревом) называется множество записей, которые могут располагаться на произвольных участках памяти ЭВМ, связанных с помощью адресов связи и удовлетворяющим условиям:

существует одна запись, которая называется корнем дерева, она не связана ни с какой другой записью верхнего уровня;

к любой записи  $i$ -го уровня ведет адрес связи только от одной записи  $(i-1)$ -го уровня.

Количество уровней, из которых состоит дерево, – ранг дерева, множество записей  $i$ -го уровня, которые адресуются от общей записи  $(i-1)$ -го уровня образуют группу записей; максимальное число записей в группе – порядок дерева. На рис. 4.8 изображено дерево, которое имеет ранг, равный 4, и порядок, равный 3.

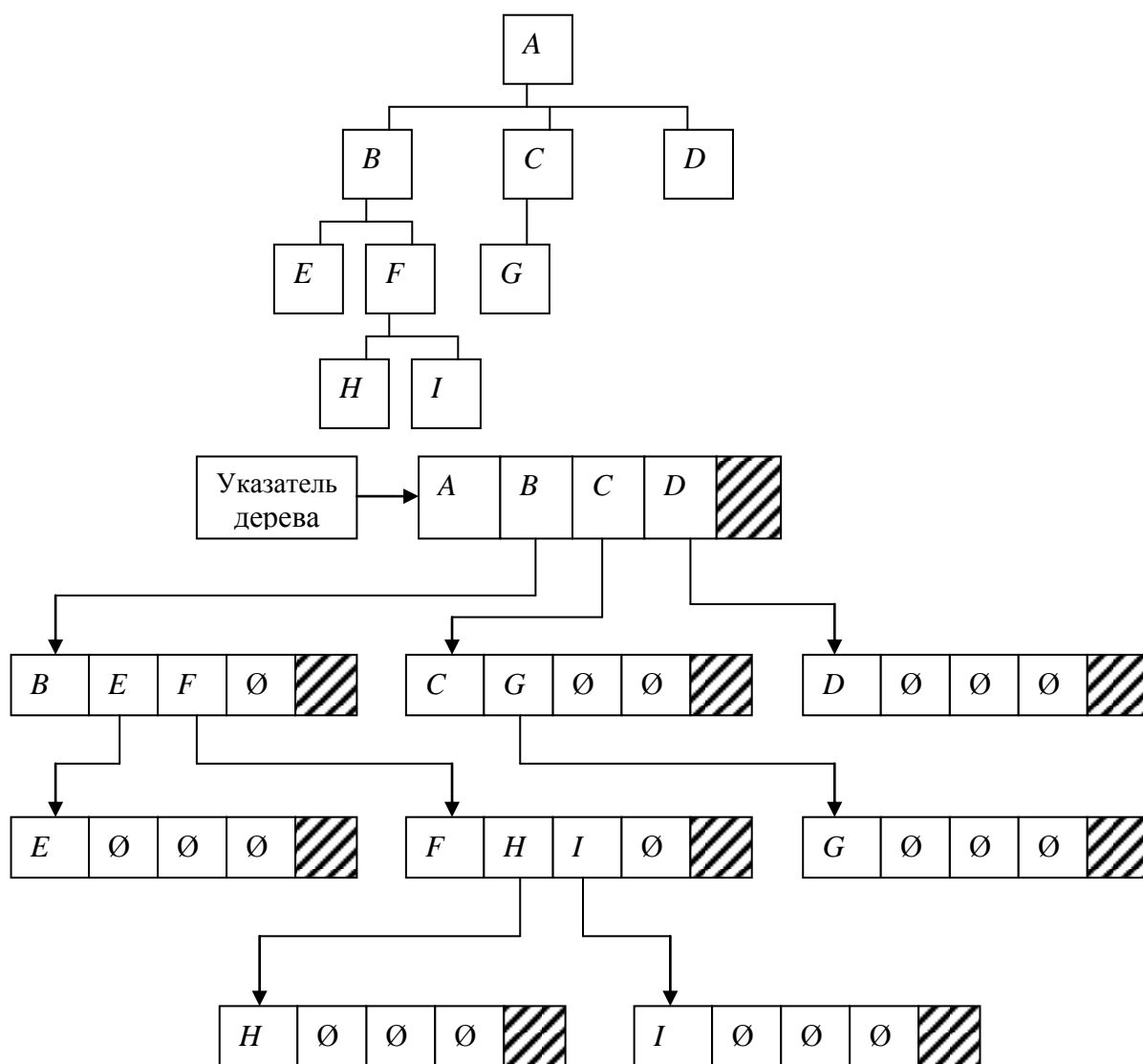


Рис. 4.8



#### 4.4.1. Организация данных в виде бинарного дерева

Бинарное – это такое дерево записей, у которых порядок равен 2. Оно упорядочивается по значению ключевого атрибута. Каждая запись имеет два адреса связи: левый и правый. У каждой записи различают две ветви дерева: левую и правую. Записи, у которых заполнены два адреса связи, называются полными, записи с одним заполненным адресом связи – неполными, записи с двумя незаполненными адресами связи – концевыми.

Пример бинарного дерева (рис. 4.9): запись *A* имеет ключевой атрибут 23, запись *B* – 10, *C* – 27, *D* – 8, *E* – 18, *F* – 32, *G* – 15, *H* – 21, *I* – 30, *J* – 32.

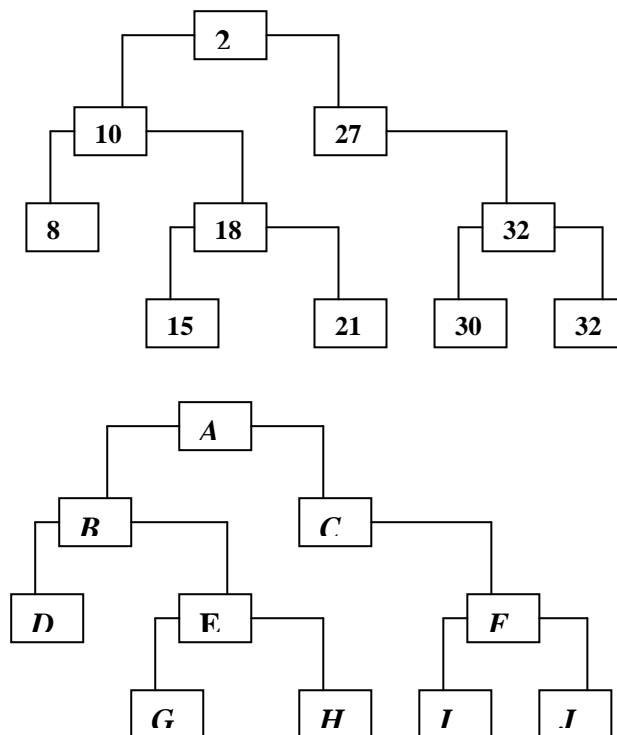


Рис. 4.9

Приведенный пример содержит следующие записи: *A*, *B*, *E*, *F* – полные; *C* – неполная, *D*, *G*, *H*, *I*, *J* – концевые. Адреса связи делятся на левые и правые. Так, адрес от *E* к *G* – левый, от *E* к *H* – правый. У записи *C* правая ветвь состоит из записей *F*, *I*, *J*, левая ветвь пустая.

Дерево упорядочивается следующим образом: значение ключевого атрибута каждой записи должно быть больше значений ключевых атрибутов записей, которые расположены в ее левой ветви. Ключевой атрибут данной записи должен быть меньше или равен ключевым атрибутам записей, которые расположены в правой ветви. Это позволяет различать левые и правые ветви дерева.

Поиск данных в бинарном дереве по совпадению начинается в его корне.

Алгоритм поиска записи в бинарном дереве, если ключевой атрибут искомой записи – *q*:

1) сравниваем  $q$  со значением корня  $p_1$ :  
если  $q < p_1$ , то просмотр осуществляется по левой ветви корня дерева;

$q > p_1$ , то по правой ветви;

2) на  $i$ -м шаге поиска  $q$  сравниваем с  $p_i$ . Запись найдена:

если  $q = p_i$ ;

$q < p_i$ , то осуществляем переход к записи, расположенной на левой ветви записи  $p_i$ ;

$q > p_i$ , то переходим к записи, расположенной на правой ветви записи  $p_i$ .

Допустим,  $q = 15$ . Сравниваем  $q$  с корнем дерева: если  $q < 23$ , то осуществляем просмотр по левой ветви дерева, если  $q > 10$ , то по правой ветви записи  $B$ , если  $q < 18$ , то просматриваем левую ветвь записи  $E$ . Искомая запись  $G$  найдена.

Рассмотренные организации данных (последовательная, цепная и бинарное дерево) предназначены для обработки записей с одним ключевым атрибутом. Они взаимозаменяемы и их можно сравнить по критериям (табл. 4.1).

Таблица 4.1

#### Критерии сравнения

Критерий сравнения	Методы организации данных		
	последовательная	цепная	бинарное дерево
1. Время формирования массива данных	$\sim M \cdot \log_2 M$	$\sim M \cdot \log_2 M$	$\sim M \cdot \log_2 M$
2. Время поиска	$\sim \log_2 M$	$\sim M$	$\sim \log_2 M$
3. Время корректировки	$\sim M$	$\sim M$	$\sim \log_2 M$
4. Объем дополнительной памяти	0	$\sim M$	$\sim M$

Сравнение различных способов организации данных в памяти ЭВМ по критериям:

по первому (времени формирования массива данных): преимущество имеют цепная организация и бинарное дерево, так как сортировке подвергается не вся информация, а только служебная (адреса связи). В последовательном массиве при сортировке пересылаются записи целиком;

по второму (времени поиска): преимущество имеют последовательный массив и бинарное дерево (есть аналитические выражения для вычисления адреса записи);

по третьему (времени корректировки): лучшие характеристики имеет бинарное дерево.

по четвертому (объему дополнительной памяти): преимущество – за последовательным массивом.

Как видно, абсолютно безупречного метода организации данных не существует. Однако бинарное дерево применяют наиболее часто, так как

этот метод позволяет минимизировать время формирования, поиска и корректировки информации.

## 4.5. Методы ускорения доступа к данным

От того, как быстро осуществляется доступ к информационным данным, зависит быстродействие всей системы управления базами данных, а значит эффективность ее функционирования. Существуют традиционные методы доступа к данным, которые были рассмотрены при анализе различных моделей данных. Ускорение доступа к данным достигается нетрадиционными способами:

- нетрадиционными методами поиска данных;
- нетрадиционными методами размещения данных;
- использованием дополнительных массивов информации, которые ускоряют доступ к данным.

Все указанные способы можно применять также при доступе к информации по нескольким ключевым атрибутам одновременно. Рассмотрим примеры этих методов.

В качестве первого примера возьмем последовательный массив, в котором расположены последовательно записи с адресами 1, 2, 3,...,  $i$ ,..., $M$ :

$P(1)$	$P(2)$	...	$P(i)$	...	$P(M-1)$	$P(M)$
--------	--------	-----	--------	-----	----------	--------

Значение ключевого атрибута  $q$  нужной записи  $P(i)$  будем искать методом интерполяции (нетрадиционный поиск). Будем интерполировать с помощью линейной функции, которую построим по двум точкам  $(1, P(1))$  и  $(M, P(M))$ . Для произвольной точки  $(i, P(i))$  в соответствии с линейной интерполяцией справедливо соотношение  $P(M) - P(1) / P(i) - P(1) = M - 1$ , где  $i = 1 + ((P(i) - P(1)) \cdot (M - 1)) / P(M) - P(1)$ .  $q = P(i)$ , следовательно,  $i = 1 + ((q - P(1)) \cdot (M - 1)) / P(M) - P(1)$ .

Найденный номер записи  $i$  является приближительным для записей неопределенной и переменной длины. В этой окрестности последовательно ищут значение  $q$ .

Следующая группа методов использует специальную расстановку записей в памяти ЭВМ. Расстановка записей происходит в соответствии с так называемой адресной функцией (другие названия – рандомизирующая функция, хеш-функция). Эти методы организации данных часто называют методами рандомизации.

Адресной функцией называется зависимость  $i = f(P)$ , где  $P$  – значение ключевого атрибута в записи;  $i$  – номер (адрес) записи.

Адресная функция может вырабатывать одинаковое значение  $i$  для значений  $P$ , принадлежащих разным записям, которые называются синонимами.

Простейшая адресная функция имеет вид  $i = P - a$ , где  $a$  – константа.

Пусть  $\underline{P}$  – минимальное значение ключевого атрибута,  $\overline{P}$  – максимальное значение ключевого атрибута. Тогда  $a = \underline{P} - 1$ ,  $P = \underline{P} + \overline{P} - 1$ . Данные занимают участок памяти. Если  $\underline{P} = 11$ , то  $a = \underline{P} - 1 = 10$ ,  $P = 19$ .

Записи-синонимы связываются в цепочки с помощью адресов связи и занимают дополнительную память. Пример записи-синонима (рис. 4.10):  $i = P - 10$  ( $a = 10$ ).

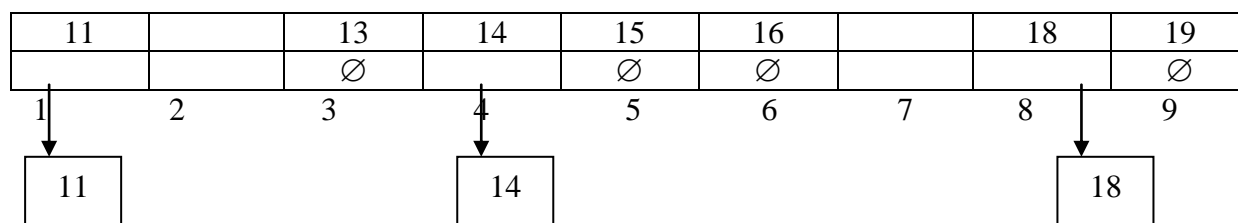


Рис. 4.10

Для ускорения поиска записей может быть использована дополнительная информация в виде массива индексов. Применяя этот метод, в оперативной памяти ЭВМ создают массив дополнительной информации, который носит название массива индексов, и в этом массиве указывают ключевой атрибут и физический адрес расположения записи с данным ключевым атрибутом. Этот массив занимает мало места и расположен в оперативной памяти ЭВМ, поэтому по ключевому атрибуту можно быстро найти запись. Обычно не все записи имеют адрес в массиве индексов, а только выборочные, через которые определяются и другие.

Существуют три разновидности массивов индексов:

сплошная индексация – для каждого ключевого атрибута каждой записи имеется адрес ее расположения;

индексно-последовательный массив – в нем имеются не все адреса расположения записей, а с определенным интервалом, который задается шагом арифметической прогрессии  $d > 1$ .

ключи записей, информация о которых выносится в индекс, – приближенно образуют арифметическую прогрессию. Применяется этот метод в случаях, когда значения ключевых атрибутов представляют собой приближенную арифметическую прогрессию. В этом случае не следует хранить значения всех ключевых атрибутов, так как адрес расположения записи вычисляется по адресной функции с учетом шага арифметической прогрессии.

При использовании сплошной индексации применяются инвертированные массивы по ключевым атрибутам (инвертированный, т. е. отсортированный по ключевым атрибутам с указанием адреса расположения соответствующей записи). Пример массива дан в табл. 4.2.

Таблица 4.2

## Инвертированный массив

Адрес записи	Ключевые атрибуты		Неключевые атрибуты
	$A$	$B$	
400	$a_1$	$b_1$	$x_1, y_1$
500	$a_1$	$b_2$	$x_2, y_2$
600	$a_2$	$b_2$	$x_3, y_3$
700	$a_3$	$b_1$	$x_4, y_4$
800	$a_1$	$b_3$	$x_5, y_5$
900	$a_2$	$b_1$	$x_6, y_6$

Инвертированные массивы для ключей  $A$  и  $B$  показаны на рис. 4.10.

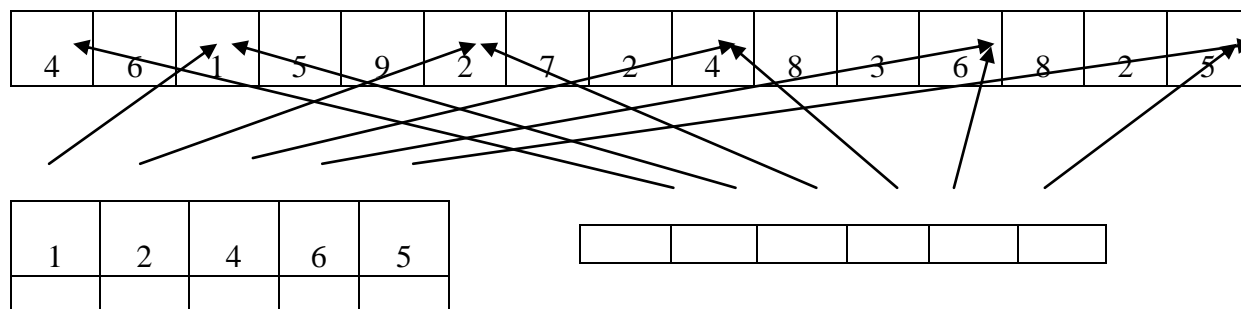
$a_1$	
	400
	500
	800
$a_2$	
$b_1$	
	400
	700
	900
$b_2$	
	500
	600
$b_3$	
	800
	600
	900
$a_3$	
	700

Индексно-последовательный массив представляет собой последовательный массив, отсортированный по значениям ключевого атрибута, к которому создается дополнительный массив индексов. В массиве индексов хранятся ключевые атрибуты и адреса записей, которые расположены с шагом арифметической прогрессии  $d$ . Массив индексов размещают в оперативной памяти ЭВМ.

Пример массива (рис. 4.11):  $d = 3$  – каждая третья запись будет иметь адрес в массиве индексов с шагом  $d$  арифметической прогрессии. Поиск нужного ключевого атрибута  $q$  происходит в две стадии:

1) в массиве индексов;

2) в основном массиве между двумя соседними индексами.



Рандомизация индекса  
(второй вид массива).

Массив индексов

(третий вид массива)

Рис. 4.11

Ключи записей возникнут тогда, когда ключевые атрибуты будут составлять приближенную арифметическую прогрессию. Если ключи записей приближенно образуют арифметическую прогрессию, то можно получить адресную функцию для индекса (рандомизация индекса). Индекс с номером  $n$  хранит адрес записи основного массива, ключ которой равен или больше

$q$  – значения ключевого атрибута:  $q = P(1) + Z \cdot n$ , где  $Z$  – шаг арифметической прогрессии;  $P(1)$  – значение ключа первой записи основного массива. Следовательно,  $n = (q - P(1)) / Z$ .

Значение ключей в таком индексе хранить не надо. Поиск осуществляется в две стадии:

определяют индекс по формуле расчета  $n$ , данной выше, результат округляют в меньшую сторону;

запись между  $n$ -м и  $(n+1)$ -м индексами.

Пример для  $Z = 10$ :  $q = 58 \Rightarrow n = \frac{58 - 14}{10} \approx 4$ .

## ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ К ГЛАВЕ 4

1. Можно ли определить бинарное дерево как организацию данных, в которой каждый элемент, кроме корня, имеет только один предшествующий элемент и наличие корня обязательно?
2. Дайте определение корня в древовидной организации данных, не используя понятия «уровень».
3. Всегда ли выполняется неравенство между числом записей в дереве  $M$  и произведением порядка дерева  $p$  на его ранг?
4. Сколько адресов связи содержат однонаправленное и двунаправленное дерево порядка  $p$ , состоящее из  $M$  записей?
5. Удовлетворяет ли одиночная запись определению дерева?
6. Если произвольная организация данных содержит контур из адресов связи, существует ли для нее понятие «ранг»?
7. Какой вид имеет упорядоченное бинарное дерево, построенное по указанному выше алгоритму (см пункт 4.4.1), для массива, который отсортирован: а) по возрастанию; б) убыванию?
8. Могут ли различные неупорядоченные массивы из  $M$  записей приводить к упорядоченным бинарным деревьям одинаковой конфигурации?
9. Всегда ли поиск заканчивается в висячей вершине?
10. Сколько соседних элементов имеет каждая запись упорядоченного бинарного дерева?
11. Какая причина может препятствовать построению симметричного бинарного дерева?
12. Существует ли соответствие между упорядоченными бинарными деревьями и цепной организацией данных?
13. Как можно использовать упорядоченные бинарные деревья для подсчета частоты встречаемости слов в тексте?
14. Если один из соседних элементов переставить на место другого, в каких случаях упорядоченность бинарного дерева сохранится, а в каких — нет?

## ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПРИ ИЗУЧЕНИИ ГЛАВЫ 4

**Задание 4.1.** Докажите, что если запись  $B$  располагается на правой ветви записи  $A$ , а запись  $C$  – на левой ветви записи  $B$ , то ключ записи  $C$  имеет промежуточное значение между ключами  $A$  и  $B$ .

**Задание 4.2.** Докажите:

что, если вновь включаемая в дерево запись  $B$  адресуется из записи  $A$ , то записи  $A$  и  $B$  являются соседними друг для друга;

записи  $B$  и  $A$  соседние, то либо запись  $B$  расположена на ветви записи  $A$ , либо запись  $A$  – на ветви записи  $B$ .

**Задание 4.3.** Постройте упорядоченные бинарные деревья для пяти массивов со следующими значениями ключевых признаков:

- 1) 56, 46, 49, 76, 49, 97, 75, 49, 8, 59, 46, 80, 15, 46, 61;
- 2) 48, 14, 54, 85, 72, 41, 20, 76, 64, 40, 19, 44, 17, 59, 87;
- 3) 69, 40, 70, 85, 45, 96, 25, 18, 47, 56, 42, 44, 70, 52, 94;
- 4) 51, 17, 22, 82, 98, 50, 79, 44, 20, 41, 46, 80, 29, 55, 61;
- 5) 44, 47, 61, 54, 27, 74, 14, 40, 55, 50, 24, 47, 28, 15, 42.

Подсчитайте число сравнений при поиске в этих пяти деревьях. В первом варианте – ключевой признак 49, во втором – 85, в третьем – 44, в четвертом – 79, в пятом – 54. Усредните полученные данные и сделайте выводы.

**Задание 4.4.** Определите максимальное число полных вершин в симметричном дереве ранга  $t$ .

**Задание 4.5.** Проставьте в вершинах бинарных деревьев (рис. 4.13–4.14) ключевые признаки, имеющие значения от 1 до 12, так, чтобы деревья стали упорядоченными.

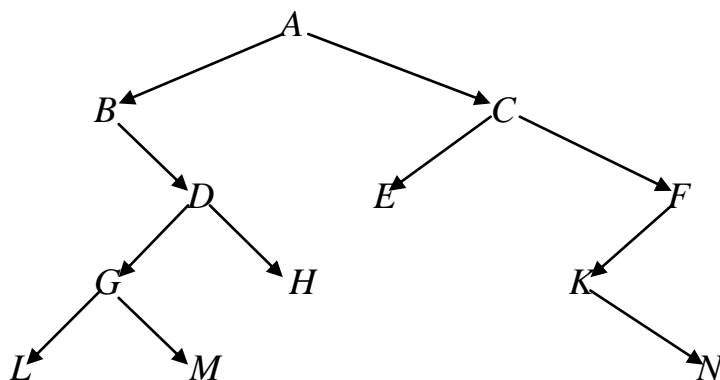


Рис. 4.12



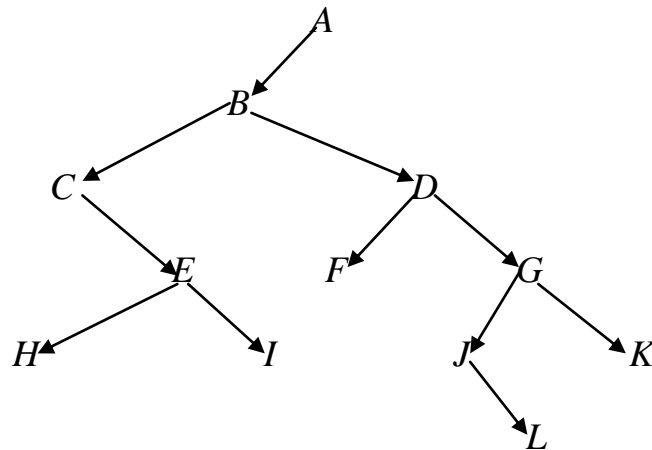


Рис. 4.13

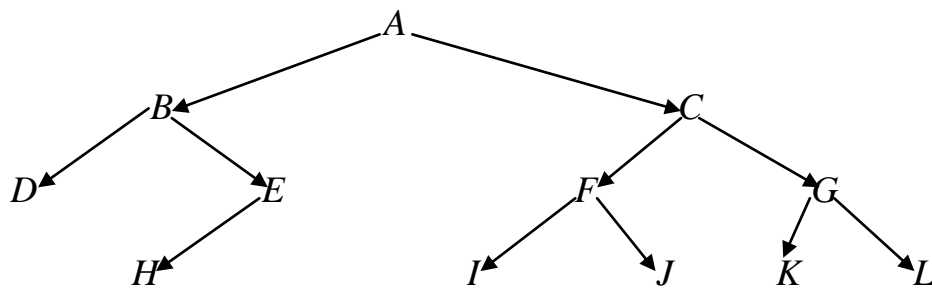


Рис. 4.14

**Задание 4.6.** В дереве (см. рис. 4.13) ключевые признаки изменяются от 1 до 15, вершина  $F$  имеет ключ 9. Проставьте остальные ключевые признаки, чтобы обеспечить упорядоченность дерева.

**Задание 4.7.** Если в бинарном дереве  $a_2$  полных вершин и  $a_1$  неполных, то сколько в нем концевых вершин  $a_0$ ?

**Задание 4.8.** Сколькими разными способами можно присоединить к бинарному дереву из  $M$  записей одну новую запись?

**Задание 4.9.** Сколько разных конфигураций получается после исключения одной висячей вершины из упорядоченного бинарного дерева, содержащего  $M$  записей?

**Задание 4.10.** Сформулируйте алгоритм перечисления вершин упорядоченного бинарного дерева таким образом, чтобы получилась последовательность записей, отсортированных по возрастанию ключа. Как изменится алгоритм, если следует получить массив, упорядоченный по убыванию ключевого признака?

**Задание 4.11.** Изобразите все возможные бинарные деревья из четырех записей, все деревья порядка 4 из шести записей, все деревья порядка 4 из семи записей.

## **ГЛАВА 5. СПОСОБЫ И МЕТОДЫ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ НА ЭТАПЕ ПРОЕКТИРОВАНИЯ**

### **5.1. Последовательность разработки информационной системы**

Установлены стадии создания автоматизированных информационных систем:

- предпроектная, включающая разработку ТЭО и ТЗ на создание информационной системы;

- разработка проектов информационной системы, включающая разработку технического и рабочего проектов или техно-рабочего проекта системы;

- ввод в эксплуатацию информационной системы, включающий в себя проведение монтажно-наладочных работ, завершение мероприятий по подготовке предприятия к внедрению информационной системы, опытной эксплуатации и приемо-сдаточных испытаний системы.

**Предпроектная стадия** включает в себя работы по обследованию системы управления организацией, позволяющие определить целесообразность создания предлагаемой информационной системы. Предпроектное обследование существующей системы управления предполагает создание документа «Технико-экономическое обоснование», в котором должна быть обоснована экономическая эффективность от внедрения автоматизированной информационной системы для разных вариантов состава функциональных и обеспечивающих подсистем, а также показатели экономической эффективности: капиталовложения, годовой экономический эффект, срок окупаемости, инвестиции.

Технико-экономическое обоснование состоит из документов:

- исходные положения и планируемые технико-экономические данные;

- Обоснование цели создания информационной системы;

- Построение схем документооборота на предприятии;

- Обоснование комплексов задач (подсистем) управления и обеспечивающих средств их реализации;

- Перечень организационно-технических мероприятий по созданию информационной системы.

Техническое задание на создание информационной системы разрабатывается предприятием-заказчиком с участием организации-разработчика. В ТЗ формулируются требования к различным элементам разрабатываемой информационной системы (к информационному обеспечению, программному, техническому и др.).

Техническое задание на создание автоматизированной информационной системы обычно состоит из разделов:

«Основание для разработки»;  
«Общие положения»;  
«Основные требования к компонентам разрабатываемой информационной системе»;  
«Стадии разработки и внедрения»;  
«Обоснование необходимости и математическая постановка предлагаемых задач автоматизации».

При **разработке технического проекта** ищут решение всех производственно-экономических задач, описывают функциональные подсистемы, в составе которых представлены эти задачи. В постановке решения производственно-экономической задачи должны быть прописаны следующие характеристики: организационно-экономическая сущность задачи, ее место в подсистеме, связь с другими задачами; даны описание ИО, формы входных и выходных документов, классификация и кодирование технико-экономической информации. Основу составляет МО: экономико-математические модели, по которым решается задача, алгоритмы решения задачи в виде блок-схем алгоритмов и их описания, контрольный пример.

Содержание документов технического проекта:

1. Описание организационной структуры управления с указанием изменений взаимосвязей между подразделениями в связи с внедрением нового комплекса задач.

2. Описание постановки и решения комплексов задач, содержащее: информационно-функциональную схему решения каждой задачи; блок-схемы алгоритмов решения задач и их описание; диаграммы потоков данных и документов до и после внедрения предлагаемой информационной системы.

3. Описание информационной базы, в которой даны сведения о: структуре и составе СЕИ; выборе и обосновании модели данных (знаний); разработке моделей данных на уровне сущностей и связей; концептуальной и даталогической модели данных; системах классификации и кодирования технико-экономической информации; формах входных и выходных документов.

4. Решения по составу комплекса технических средств.

5. Решения по необходимому составу инструментального программного обеспечения.

При **разработке рабочего проекта** следует помнить, что целью рабочего проекта является разработка технической документации, необходимой для отладки и передачи заказчику всех перечисленных в ТЗ компонентов автоматизированной информационной системы. К этим компонентам относятся описание структуры необходимого комплекса технических средств, информационной базы автоматизированной системы, базового и прикладного ПО (комплексы программ, с помощью которых решаются задачи из утвержденного перечня функциональных подсистем).

Состав документов рабочего проекта:

Руководство программисту;

Руководство пользователя;

Эксплуатационная программа или Текст программы;

Описание контрольного примера.

**Стадия ввода в эксплуатацию информационной системы** определяется приказом, изданным заказчиком и согласованным с разработчиком. К приказу прилагается согласованная с разработчиком программа проведения опытной эксплуатации, в которой определяются условия и количество испытаний комплексов задач для каждой функциональной подсистемы, порядок проверки технических средств при решении комплексов задач, порядок устранения недостатков, выявленных при опытной эксплуатации. На данной стадии проводятся работы:

опытно-промышленная эксплуатация отдельных задач и их комплексов (совместными силами заказчика и разработчика);

приемка комплексов задач в промышленную эксплуатацию;

проведение приемно-сдаточных испытаний;

приемка информационной системы в промышленную эксплуатацию, оформление акта внедрения разработанной системы в промышленную эксплуатацию по результатам опытно-промышленной эксплуатации.

## **5.2. Обследование существующей информационной системы на предприятии**

Обследование существующей информационной системы на предприятии проводится силами сотрудников предприятия-заказчика и организации-разработчика информационной системы на предпроектной стадии.

Обследование может проводиться разными способами:

изучают положения о структурных подразделениях предприятия, их правах и функциональных обязанностях, должностные инструкции и другие документы, регламентирующие работу предприятия;

проводят опросы сотрудников подразделения с помощью анкет;

наблюдают за процессом функционирования предприятия в течение определенного интервала времени и анализируют производственно-экономической деятельности и др.

Если это возможно, то лучше использовать все три перечисленных способа одновременно.

Основу предпроектного обследования составляют анализ и модификация документооборота на предприятии. Анализ заключается:

в определении перечня документов, которые разрабатываются в каждом подразделении;

рассмотрении производственной и экономической терминологии, которая используется в документах, документы приводятся к стандартным формам;

производственно-экономических показателей, которые вычисляются в обследуемых структурных подразделениях;

применяемой системы классификации и кодирования технико-экономической информации;

построение схем документооборота внутри и между структурными подразделениями;

разработке алгоритмов вычисления экономических показателей и алгоритмов взаимосвязи между показателями;

модернизации системы документооборота на основе критериев снижения избыточности информационных данных и унификации документов.

Схемы документооборота строятся для заданного подразделения предприятия. Организационная схема внутривнутриструктурных потоков информации представлена на рис. 5.1, квадрат обозначает процесс возникновения документа, а круг – отдельную операцию, производимую с документом для его оформления. Информационно-функциональная схема решения задач в конкретном подразделении показана на рис. 5.2.

Для всего предприятия в целом или крупного подразделения предприятия может быть построена схема документооборота в виде матричной информационной модели взаимосвязи подразделений, документов и производственно-экономических показателей предприятия.

Наиболее распространенной является матричная информационная модель для конкретного подразделения предприятия (рис. 5.3). Она представляет собой таблицу, которая содержит четыре квадранта и вспомогательные разделы, предназначена для характеристики документов информационной системы, экономических показателей, подразделений, которые используют эти документы, взаимосвязи между этими документами и показателями. Она может быть составлена для всего предприятия или для одного крупного подразделения (производства) предприятия.

Первый квадрант характеризует все документы, которые разрабатываются и хранятся в данном подразделении. В качестве заголовков строк выступают нормативно-справочные документы, которые хранятся в данном подразделении (ценники на продукцию, стандарты и т. п.). Основное содержание каждого разрабатываемого документа включает учет нормативных производственно-экономических показателей, которые являются строками первого квадранта. В качестве столбцов используются разрабатываемые документы и показатели в данном подразделении.

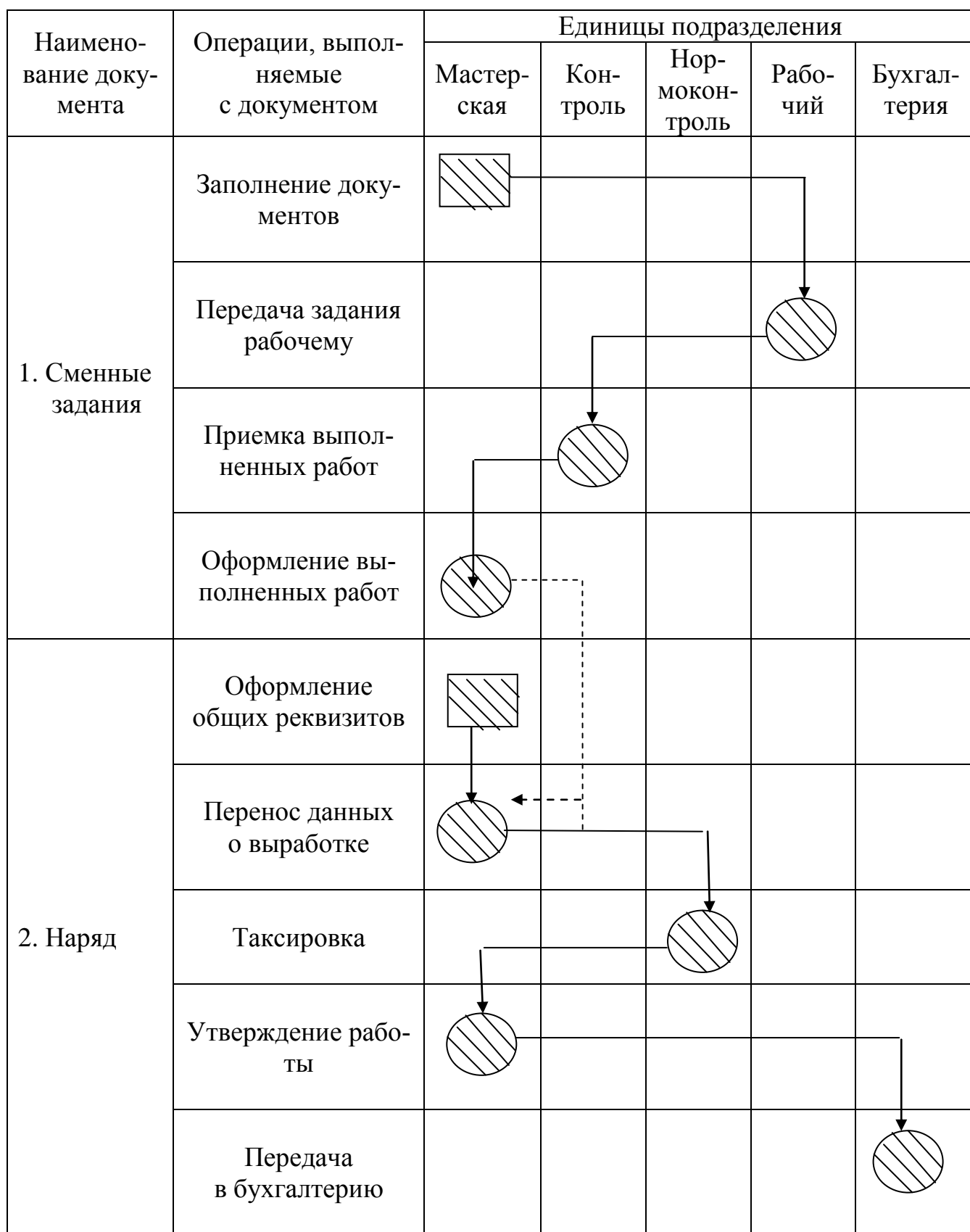
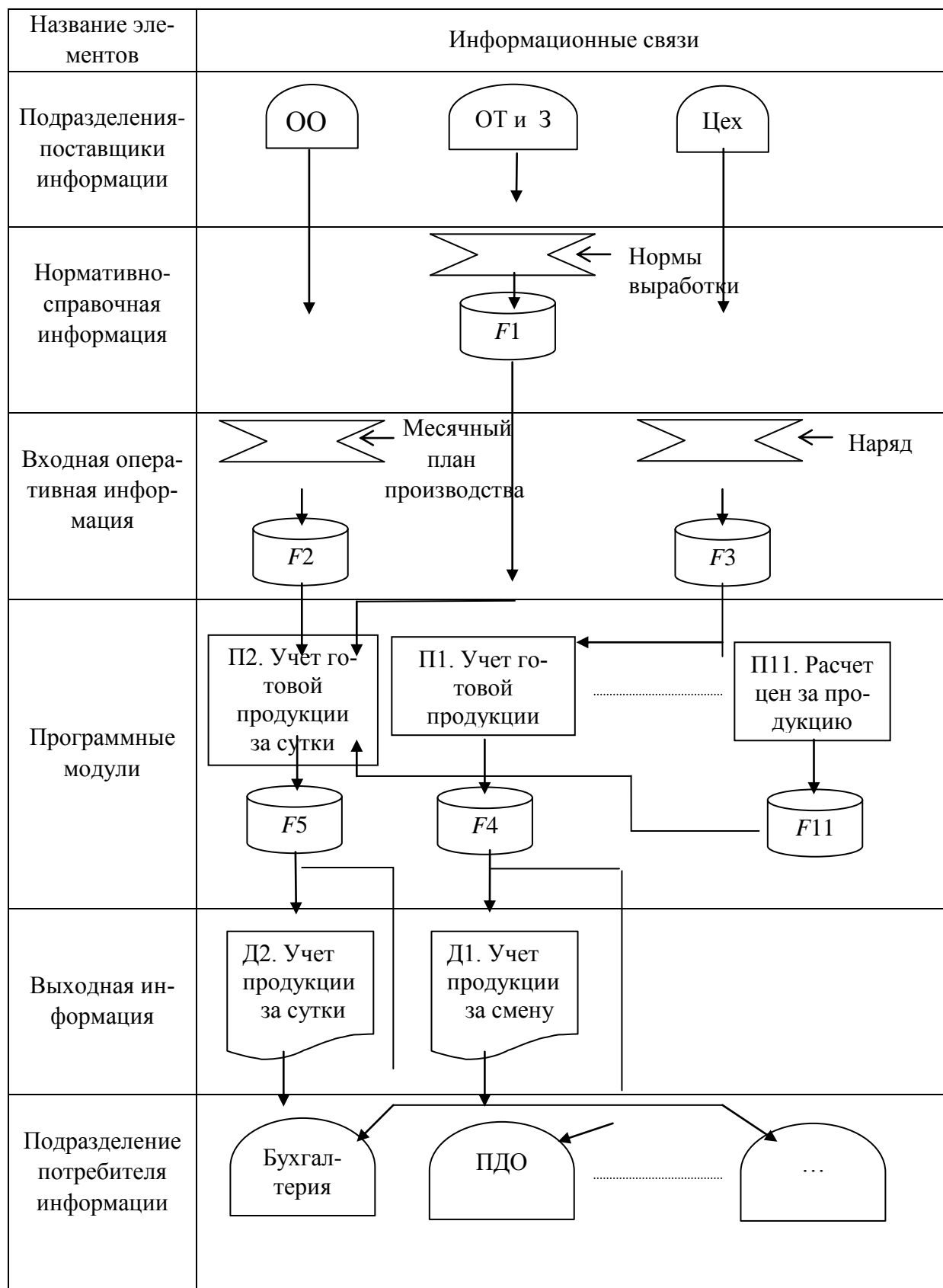


Рис. 5.1







На пересечении соответствующей строки и столбца (см. рис. 5.3) ставят отметку, если данный нормативный показатель используется для расчета производственно-экономического показателя из соответствующего столбца. Итоговый столбец показывает, сколько раз данный нормативный показатель применялся для расчета других показателей, а итоговая строка – количество нормативных показателей, которые необходимы для расчета разрабатываемого показателя. Таким образом, первый квадрант показывает взаимосвязь документов и показателей внутри данного структурного подразделения. Второй квадрант предназначен для того, чтобы показать, какие нормативные документы и показатели данного подразделения применяются в других подразделениях, использующих информацию данного подразделения.

На пересечении строки и столбца ставится отметка, если данный нормативный документ или показатель этого подразделения используется другими подразделениями. Строки первого и второго квадранта совпадают. В качестве столбцов используют документы и показатели, разрабатываемые в других подразделениях предприятия. В итоговом столбце указывают, сколько раз нормативный показатель или документ применялся в расчетах других подразделений. В итоговой строке содержится количество нормативных показателей данного подразделения, которые использовались другими подразделениями.

Третий квадрант показывает, из каких подразделений берется информация при разработке документов данного подразделения. Столбцы – разрабатываемые документы и показатели в данном подразделении. Строки третьего квадранта показывают, какие документы и показатели используются из подразделений поставщиков информации в данном подразделении. Если какие-либо документы и показатели задействованы, то на пересечении строки и столбца ставится отметка.

Четвертый квадрант предназначен для того, чтобы показать ту информацию из документов подразделений поставщиков, которая, минуя данное подразделение, употребляется в работе других подразделений. В качестве заголовков строк используют документы и показатели из подразделений поставщиков информации для данного подразделения. В качестве заголовков столбцов выступают документы и показатели, разрабатываемые в других подразделениях предприятия.

### **5.3. Параметризация информационных систем**

Под параметризацией информационных систем понимается определение параметров отдельных компонент информационных систем, параметров свойств этих компонент, взаимосвязи между параметрами. При рассмотрении этих параметров, свойств выделяют компоненты, которые определяют классы конкретных параметров информационной системы.

Классы параметров информационной системы:

1. Параметризация информационного обеспечения (БД и БЗ). В этом классе параметров рассматриваются характеристики ИО системы. Под параметрами БД подразумеваются тип модели данных, которая используется в БД (параметры реализации реляционной, сетевой или иерархической модели данных), обоснование выбора модели данных, внутренние компоненты БД и отношения между ними, функциональные зависимости между атрибутами модели данных. К параметрам БЗ относятся тип модели знаний (продукционная, фреймовая, в виде семантических сетей) и ее характеристики.

2. Параметризация ПО информационных систем:

параметры внутреннего программного обеспечения, к которым относятся тип и технические характеристики операционной системы;

    сетевого ПО;

    прикладного ПО;

    вхождения отдельных программных модулей в производственно-экономические задачи;

    вхождения производственно-экономической задачи в функциональную подсистему

    архитектура прикладного ПО (например, архитектура «клиент – сервер»).

3. Параметризация ограничений на доступ пользователей к информационным данным системы. Рассматриваются параметры, связанные с сохранением целостности данных и управлением доступа к данным в информационной системе.

4. Параметризация потоков данных и запросов. Этот класс параметров определяет характеристики типов задач, интенсивность поступления данных и запросов в информационную систему, обработки данных в информационной системе.

5. Параметризация вычислительной системы. В данный класс входят параметры комплекса технических средств информационной системы, к которым относятся параметры компьютера, линий передачи данных, терминального оборудования, другого оборудования, входящего в информационную систему.

Большинство классов параметров информационной системы представлено в данном учебном пособии и в других дисциплинах учебного плана. Рассмотрим для примера некоторые классы параметризации информационных систем.

## **5.4. Параметризация потоков данных и запросов**

Чтобы определить параметры потоков данных и запросов, необходимо классифицировать производственно-экономические задачи, которые решает система. Каждая производственно-экономическая задача характе-

ризуется параметрами: периодичностью решения задачи; временем решения; временем ожидания обслуживания; допустимым временем решения; общим временем обслуживания задачи.

В соответствии с этими параметрами все задачи делятся на классы:

1. Задачи оперативного взаимодействия – решаются по запросу в диалоговом режиме (запрос – ответ). Время поступления этих задач на обслуживание носит вероятностный характер.

2. Регулярные задачи – решаются регулярно в соответствии с расписанием решения таких задач (ежемесячный учет выпуска готовой продукции).

3. Нерегулярные задачи – фоновые задачи для выравнивания загрузки информационных систем: сортировка данных по ключевому атрибуту, отладка, поиск.

Параметры решения:

1. Задачи оперативного взаимодействия. Операции взаимодействия носят вероятностный характер и зависят от показателя загрузки ( $\rho_1$ ); показатель загрузки  $\rho_1 = \lambda_1/\mu_1$ , где  $\rho_1 < 1$ ;  $\lambda_1$  – интенсивность поступления на обслуживание;  $\mu_1$  – интенсивность обслуживания. Время обслуживания таких задач ограничено.

2. Регулярные задачи поступают на обслуживание согласно календарному плану-графику. Строятся диаграммы загрузки информационной системы по дням в течение каждого месяца и по месяцам в течение текущего года. Определяется максимальное время, необходимое для решения регулярных задач в течение рассматриваемого цикла (рис. 5.4):  $T_j \max = \max\{T_1, T_2, \dots\}$ .

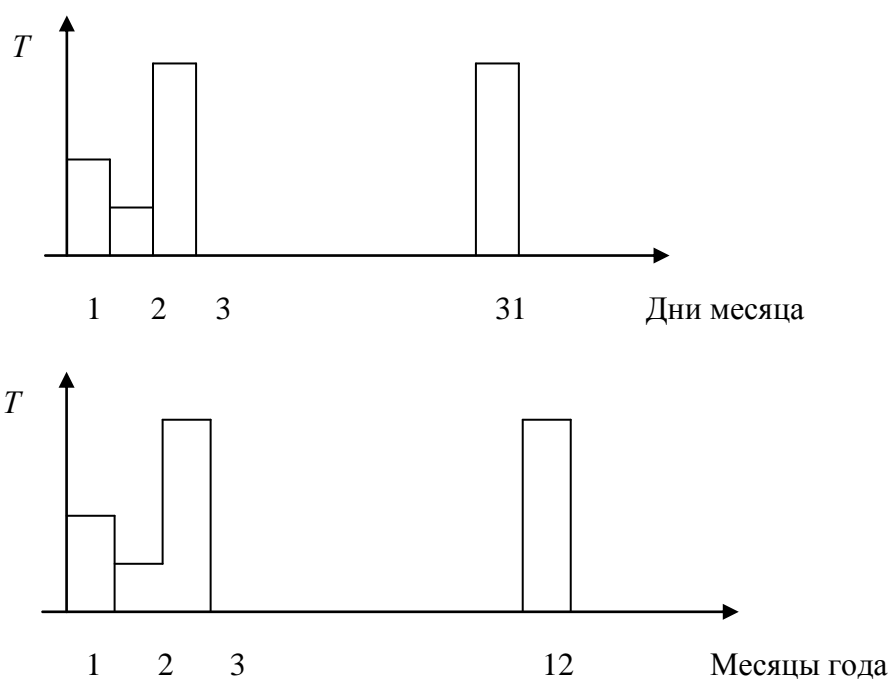


Рис. 5.4

Величина  $T_{jmax}$  является исходным значением для расчета производительности информационной системы.

3. Нерегулярные задачи – не регламентированы сроками решения и носят вероятностный характер. К нерегулярным задачам относятся задачи сортировки и корректировки данных, отладки программных модулей, исправления ошибок и др. Показатели обслуживания нерегулярных задач:

интенсивность поступления на обслуживание ( $\lambda_3$ );

интенсивность обслуживания ( $\mu_3$ );

показатель загрузки ( $\rho_3$ ).

Загрузка системы задачами этого класса  $\rho_3 = \lambda_3 / \mu_3$ . Для стационарных информационных систем  $\rho_3 < 1$ .

Задается допустимое время решения задач  $T_{доп}$ , которое необходимо для того, чтобы управляющее воздействие после принятия решения было обработано и успело оказать воздействие на производственно-экономический процесс. Этот параметр задается исходя из свойств быстрогодействия производственно-экономического процесса.

Для того чтобы найти общее время на решение задач всех трех классов  $T_{общ}$ , нужно сначала установить:

1) время на решение задач первого класса:

$$T1 = \rho_1 \cdot T_{доп};$$

2) время на решение регулярных задач:

$$T2 = T_{jmax} \cdot K_{впр} / (K_{гпс} \cdot K_{мульти}),$$

где  $K_{впр}$  – коэффициент возможности повторения расчетов;  $K_{гпс}$  – комплексный показатель надежности (коэффициент готовности программных средств), который используется для решения регулярных задач (вероятность того, что в произвольный момент времени программное средство будет готово к решению);  $K_{мульти}$  – коэффициент мультипрограммности для многопроцессорных вычислительных систем (решает параллельно несколько задач);

3) время на решение нерегулярных задач:

$$T3 = \rho_3 \cdot T_{доп}.$$

После этого складываем полученные значения:  $T_{общ} = T1 + T2 + T3$ .

Показатель общей загрузки информационной системы  $I$  рассчитывается по формуле  $I = T_{общ} / T_{доп} \cdot K_{ог}$ , где  $K_{ог}$  – коэффициент оперативной готовности комплекса технических средств, на котором функционирует информационная система.

Коэффициент  $K_{ог}$  показывает вероятность застать технические средства информационной системы в произвольный момент времени в работоспособном состоянии и начиная с этого момента времени увидеть, функционируют ли они безотказно в течение заданного времени решения производственно-экономических задач:

$$0 < K_{ог} < 1,$$

$$K_{ог} = K_{г} \cdot P(T_{доп}),$$

где  $K_r$  – коэффициент готовности технических средств;  $P(T_{\text{доп}})$  – вероятность безотказной работы системы за допустимое время решения производственно-экономической задачи.

Если  $I < 1$ , то задачи всех классов будут обслужены за  $T_{\text{доп}}$ .

Если  $I > 1$ , то производительность системы ниже требуемого уровня. Ее необходимо увеличить посредством использования более мощных компьютеров, серверов, дублировать рабочие станции. Все это обеспечит нормальное быстродействие системы при решении производственно-экономических задач.

## 5.5. Параметризация вычислительной системы

В ходе параметризации вычислительной системы осуществляется определение параметров компонент вычислительной системы, количество компьютеров, используемых в системе, параметров периферийного оборудования, сетевого оборудования, каналов связи, применяемых в системе.

### *Определение количества компьютеров или каналов связи*

Современные информационные системы являются сетевыми, следовательно, они используют различные типы каналов: витую пару, оптоволоконные и др. Основной параметр – пропускная способность, измеряемая в Кбит/с. Чем она выше, тем дороже канал связи, поэтому необходимо выбрать оптимальный вариант.

Пусть имеем заданную пропускную способность каналов связи. Тогда, чтобы рассчитать количество компьютеров, которые нужно использовать в сети, задают допустимую вероятность отказа в предоставлении поступившей задачи канала связи или допустимое время ожидания обслуживания  $t_{\text{ож. д}}$  (рис. 5.5).

При использовании данных ограничений рассчитывается  $P_n$  – вероятность занятости всех каналов связи в системе с  $n$  АРМ (компьютерами) по формуле Эрланга:

$$P_n = \frac{(\lambda \cdot \bar{\eta})^n \cdot \frac{1}{n!}}{\sum_{k=0}^n (\lambda \cdot \bar{\eta})^k \cdot \frac{1}{k!}}$$

где  $n$  – количество каналов связи (компьютеров) в информационной системе;  $\lambda$  – интенсивность поступления задач на обслуживание;  $\bar{\eta}$  – математическое ожидание времени решения одной задачи;  $k$  – количество задач, поступивших на рассматриваемом промежутке времени от 0 до  $t$ .

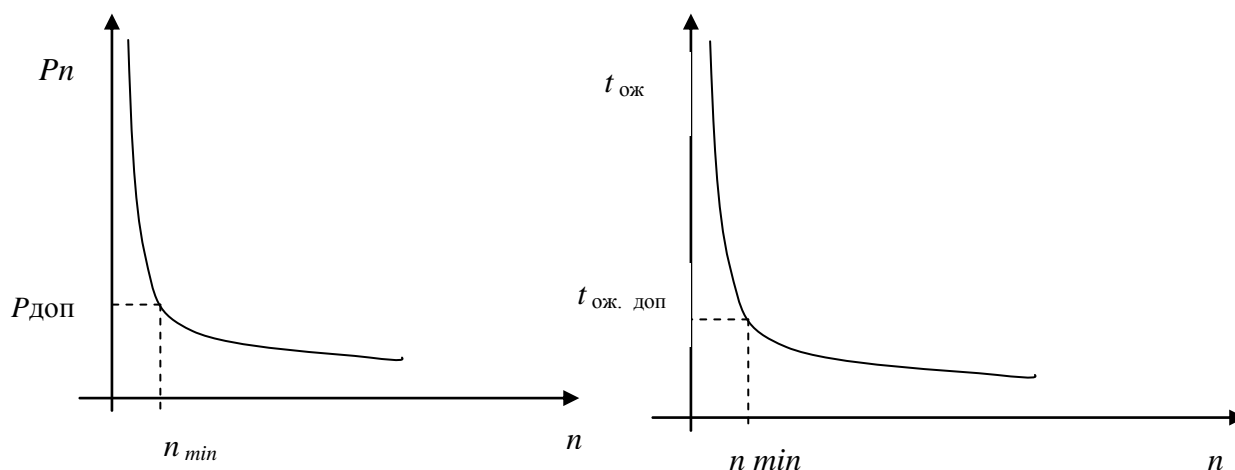


Рис. 5.5

Формулировка обратной задачи выглядит так: пусть задано количество каналов связи (компьютеров)  $n$ , тогда определяем  $P_{\text{доп}}$  и  $t_{\text{ож. д.}}$ .

## 5.6. Защита данных в информационных системах

Важнейшие аспекты защиты данных – это сохранение целостности и управление доступом.

### 5.6.1. Сохранение целостности данных

Сохранение целостности данных рассматривают в двух аспектах:

#### **организационные мероприятия:**

информационные данные должны храниться в местах, недоступных для посторонних лиц;

важная производственно-экономическая информация должна храниться в нескольких экземплярах (в потенциально опасных производственных и банковских системах используются зеркальные диски, на которых дублируют информационные данные);

информация, которая хранится в памяти ЭВМ, должна периодически копироваться на внешние запоминающие устройства (лазерные диски, флеш-карты и др);

необходимо строго придерживаться правил работы с запоминающими устройствами;

информация, которая предназначена для решения конкретной производственно-экономической задачи, должна храниться в отдельном месте.

**технологические мероприятия** (связаны с параметрами конкретной СУБД):

к некоторым подпрограммам должен быть запрещен доступ общих пользователей информационной системы;

значения производственно-экономических показателей изменяются в определенных диапазонах, от минимального до максимального значения, в случае их выхода за эти пределы необходимо произвести проверку нарушения целостности БД, т. е. выявить некорректную информацию, хранимую в информационной системе;

необходимо следить за функциональными зависимостями между атрибутами, которые хранятся в одном отношении. Если произошло нарушение функциональной зависимости, то утрачена целостность информационной базы. Проверку информационных зависимостей можно определить по вероятным ключам отношений, которые называются первичными. В большинстве СУБД предусмотрена такая проверка.

Рассматриваются следующие общие задачи управления доступа к данным:

защита от несанкционированного чтения информационных данных в базе;

от несанкционированной модификации или уничтожения данных (если имеет связь с сетью Internet).

### **5.6.2. Методы управления доступом**

**Метод программных паролей** – самый распространенный. Его надежность невысока, так как путем подбора пароля можно взломать информационную систему. Метод базируется на таблице, которая располагается в оперативной памяти (таблица).

Таблица управления доступом

Пользователи доступа	Объекты доступа					
	Программа А	Программа В	...	Файл А	Файл В	...
Бухгалтерия	–	–		Чтение	Запись	
Отдел кадров	–	Вычисление		–	Чтение	
...						
Иванов	Вычисление	–		Запись	–	
Петров		Вычисление		Чтение	–	

Другие методы – **шифрация данных** (данные в БД закодированы, их нельзя прочитать без дешифратора) и **метод автоматического обратного вызова**.

**Метод автоматического обратного вызова** применяется в сетевых информационных системах, когда пользователь БД находится на удалении. Пользователь посылает специальные коды в СУБД, которая их обрабаты-

вает. Если коды верны, то автоматически посылается вызов (разрешение) для работы с БД.

На физическом (аппаратном) уровне защиты данных пользователь должен иметь **электронный ключ** (еще один метод управления доступом), с помощью которого запускается процессор и происходит доступ в БД.

Общая структура алгоритма управления доступом представлена на рис. 5.6.

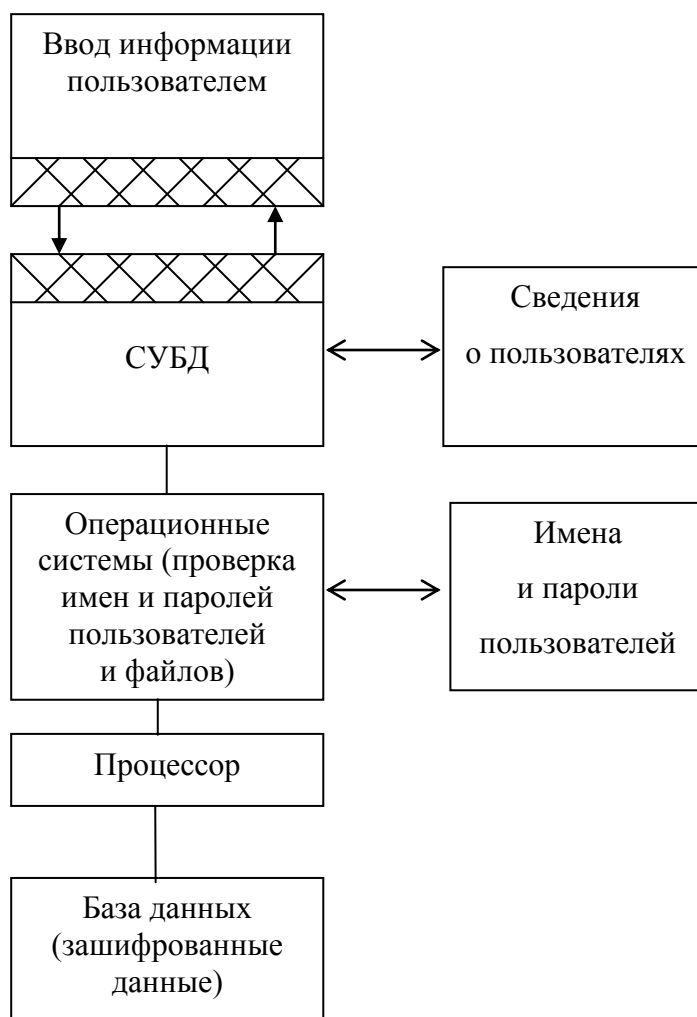


Рис. 5.6



## **ГЛАВА 6. ПРИМЕРЫ МАТЕМАТИЧЕСКОГО И АЛГОРИТМИЧЕСКОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ ДЛЯ МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ**

К МО относятся математические методы, модели и алгоритмы решения производственно-экономических задач. Эти методы, модели и алгоритмы используются при составлении ТЗ на разработку автоматизированной информационной системы для формирования математической постановки предлагаемых задач автоматизации [11; 12]. Для примеров рассмотрим следующие математические постановки задач:

1. Прогнозирование значений производственно-экономических показателей.
2. Управление запасами сырья, материалов и готовой продукции.
3. Выбор порядка обработки деталей на двух станках.

### **6.1. Математическое обеспечение задачи «Прогнозирование»**

В подсистеме планирования должны решаться задачи прогнозирования уровней производственно-экономических показателей.

Прогнозирование – это расчет ожидаемого уровня значений производственно-экономических показателей (догадка, подкрепленная знанием). При прогнозировании важна длительность реализации прогноза, так как должно быть реализовано управляющее воздействие. Например, если период короткий, то управляющее воздействие по результатам прогноза может быть не реализовано; если слишком длинный, то погрешность прогноза будет велика, следовательно, принятое решение будет неверным.

Существует три типовых задачи прогнозирования для промышленного предприятия:

прогнозирование развития предприятия, связанное с увеличением и реконструкцией его производственных мощностей; осуществляется решается на 5–10 лет вперед;

прогнозирование выпуска готовой продукции на год (функциональная подсистема технико-экономического планирования) с учетом имеющихся производственных мощностей: составляется годовая производственная программа, которая должна учитывать все факторы производственной деятельности (неравномерность поставки сырья, материалов и энергии; выход из строя единиц производственной мощности);

прогнозирование выпуска готовой продукции на квартал и на месяц с учетом возможности реализации продукции.

Методы прогнозирования:

1. Метод экспертных оценок, который впервые применили древние греки (г. Дельфы, дельфийские мудрецы). Привлекаются специалисты-эксперты в соответствующей отрасли знаний, которые и делают экспертные оценки. Для увеличения точности часто привлекают группы экспертов. В настоящее время применяются статистические методы для усреднения оценок и получения прогноза с заданным уровнем достоверности.

2. Метод анализа индексов деловой готовности Основан на анализе индексов деловой готовности (косвенных показателей), которые можно измерить в настоящее время и затем осуществить прогноз. Применяется метод в том случае, если между прошлым и будущим имеются причинно-следственные связи. Например, известна рождаемость детей, требуется спрогнозировать количество детской одежды, мест в школе через 7 лет. Если рождаемость детей увеличивается, то следует накапливать все ресурсы, связанные со школьным образованием.

3. Статистический анализ ретроспективных данных производственно-экономических показателей, т. е. выявление причинно-следственных связей между будущими и прошлыми показателями с помощью статистического анализа.

Рассмотрим модели прогнозирования, полученные методом статистического анализа ретроспективных данных.

Общая методика прогнозирования:

собираем временной ряд статистических данных о производственно-экономическом показателе (например, о зависимости спроса от времени). Количество статистических значений показателя должно быть в три раза больше, чем следует получить прогнозных значений. Часто для прогноза спроса на продукцию следует учитывать фактор сезонности;

выбираем вид прогнозирующей функции и определяем ее параметры;

анализируем погрешности полученных оценок прогнозируемых данных;

принимаем решение, использовать или нет предлагаемую функцию для решения задачи по прогнозу заданного показателя;

определяем прогнозирующую функцию:  $d_t^n = f(t)$ , где  $d_t^n$  – прогнозируемое значение зависимой переменной (некоторого показателя);  $t$  – независимая переменная (время).

Наиболее распространенные виды прогнозирующих функций:

1) не зависящие от времени:  $d_t^П = \alpha$  (рис. 6.1).

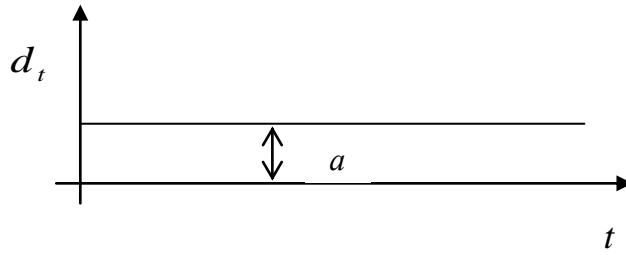


Рис. 6.1

2) линейная функция времени:  $d_t^{\Pi} = a + b \cdot t$  (рис. 6.2).

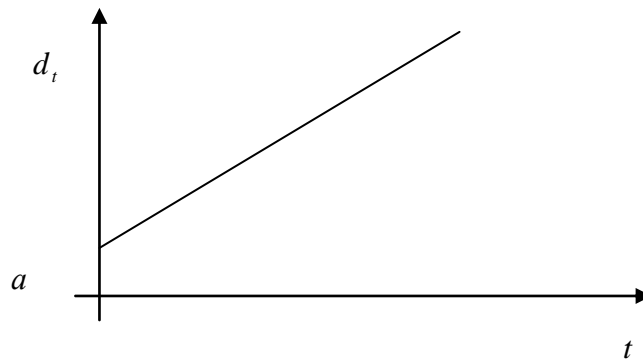


Рис. 6.2

3) циклическая функция времени:  $d_t^{\Pi} = a + u \cdot \cos \frac{2\Pi}{N} t + v \cdot \sin \frac{2\Pi}{N} t$  (рис. 6.3).

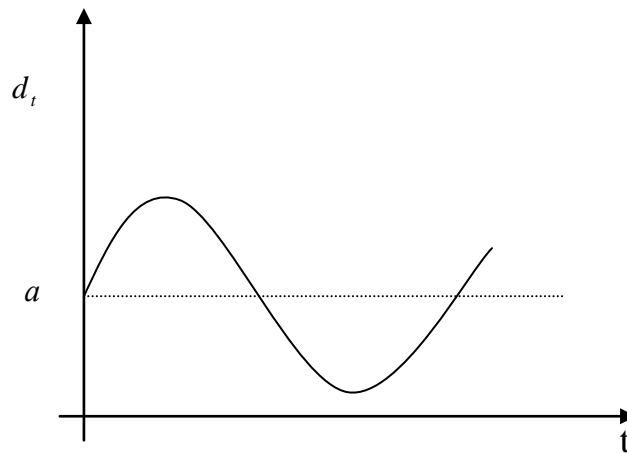


Рис. 6.3

4) линейно-возрастающая циклическая функция (рис. 6.4):  

$$d_t^n = a + b \cdot t + u \cdot \cos \frac{2\Pi}{N} t + v \cdot \sin \frac{2\Pi}{N} t$$

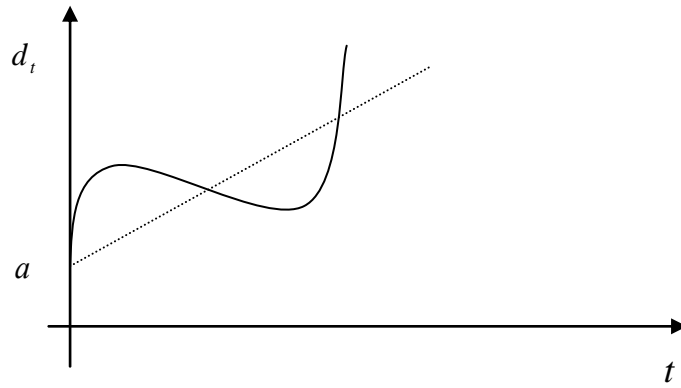


Рис. 6.4

Рассмотрим определение прогнозирующих функций методом наименьших квадратов. Метод позволяет выбрать и построить график функции по некоторой совокупности точек. При выборе необходимо достичь минимума среднеквадратического отклонения (рис. 6.5):

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (d_t - d_t^n)^2}{n - f}},$$

где  $d_t$  – фактическое значение показателя в момент времени  $t$ ;  $n$  – число наблюдений;  $f$  – число степеней свободы.

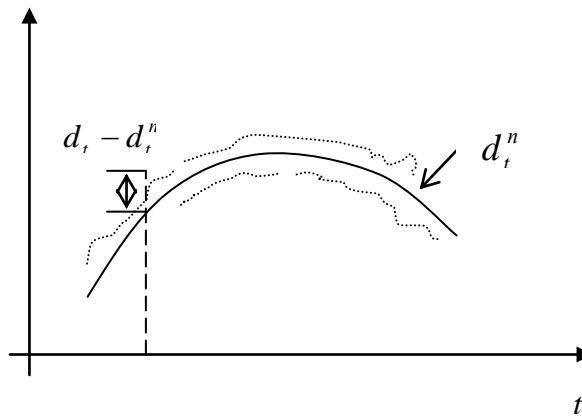


Рис. 6.5

Минимум среднеквадратического отклонения соответствует минимуму выражения

$$\min E = \sum_1^n (d_t - d_t^n)^2$$

*Пример* (для  $d_t^n = a$ ) (рис. 6.1):

$$E = \sum_1^n (d_t - a)^2; \quad \frac{\partial E}{\partial a} = 2 \sum_1^n (d_t - a) = 0; \quad 2 \sum_1^n d_t = 2na; \quad a = \left( \sum_1^n d_t \right) / n.$$

## 6.2. Математическое обеспечение задачи «Управление запасами»

Под запасами понимаются изделия, которые хранятся на складе для последующего использования или хранения. Запасы могут быть видов (рис. 6.6):

1. Сырье, полуфабрикаты, другие материалы, которые необходимо использовать в производственном процессе (буферный запас).

2. Запасы готовой продукции для сбыта (запас ГП), необходимые для сглаживания поставок между поставщиком и производителем, между производителем и потребителем.

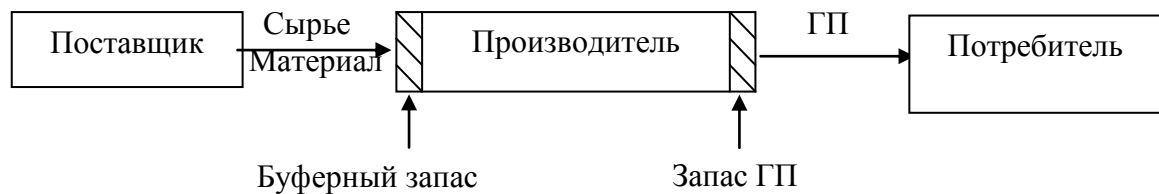


Рис. 6.6

Назначение буферного запаса:

сокращает издержки на транспортировку сырья и материалов от поставщика;

является гарантийным запасом (благодаря нему не происходит остановка производства);

с помощью него сглаживают неравномерность поставок.

Назначение запаса ГП:

устраняет непредвиденные ситуации во взаимоотношениях с потребителями;

может быть реализован партиями оптимального размера;

служит для удовлетворения непредвиденного спроса.

### ***Определение оптимального размера партии пополняемого запаса***

Виды затрат на приобретение и хранение запаса на складе бывают нескольких видов. Общее количество затрат  $C$  ищется по формуле  $C = C_n + C_{np} + C_x$ , где  $C_n$  – затраты на поставку;  $C_{np}$  – прямые затраты (стоимость поступивших изделий);  $C_x$  – затраты на хранение запасов на складе.

Затраты на поставку  $C_n$  рассчитываются по формуле  $C_n = A \cdot S / q_0$ , где  $A$  – стоимость поставки одной партии заказа;  $S$  – количество единиц изделий, которые необходимо поставить за год предприятию;  $q_0$  – оптимальный размер одной партии заказа (количество единиц продукции в одной партии);  $S / q_0$  – количество партий заказов в год.

Формула для определения прямых затрат –  $C_{np} = S \cdot c$ , где  $c$  – стоимость одной единицы изделия.

Если  $i$  – стоимость хранения одного изделия в год, то затраты на хранение запасов на складе (рис. 6.7)  $C_x = q_0 \cdot i / 2$ . Тогда

$$C_x = \frac{q_0 \cdot i}{2}, \text{ тогда } C = A \cdot \frac{S}{q_0} + S \cdot c + \frac{q_0 \cdot i}{2},$$

$$\frac{\partial C}{\partial q_0} = -A \cdot S \cdot q_0^{-2} + \frac{i}{2} = 0, \Rightarrow q_0 = \sqrt{\frac{2AS}{i}}.$$

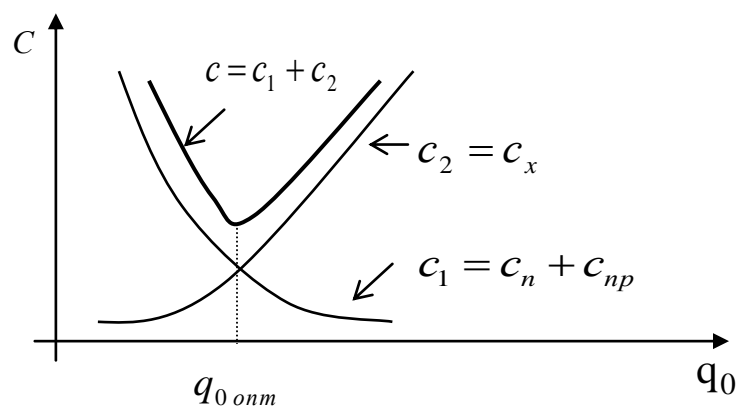


Рис. 6.7

Пример:  $A = 8,33$  усл. ед.;  $i = 0,1$  усл. ед.;  $S = 1500$  изд. Вычисляем  $q_0$ :  $q_0 = 500$  – размер партии, соответствующий минимуму суммарных годовых издержек.

Обозначим количество партий в год как  $l$ . Тогда  $l = S / q_0 = 3$ , откуда следует, что спрос постоянен в течение года. Следовательно, заказ должны размещать через каждые 83 дня, что устанавливается следующим образом:  $T_n = 250 / 3 = 83$ , где 250 – количество рабочих дней в году.

Временной график управления запасами (система управления запасами с фиксированным объемом заказа) дан на рис. 6.8, на котором  $q_{зз}$  – гарантированный запас,  $q_n$  – пороговый уровень запаса (количество изделий на складе, при котором необходимо пополнять их запас),  $q_m$  – текущее расходование заказа.

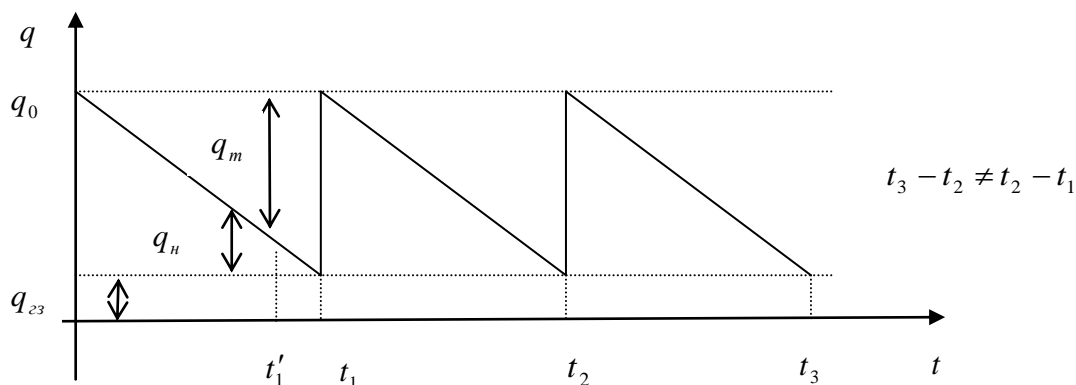


Рис. 6.8

Два типа системы управления запасами:  
с фиксированным объемом заказа (недостаток: необходимо постоянно следить за тем, чтобы  $q_0 = q_n + q_m$ );  
фиксированным временем между заказами (недостаток: необходимо поддерживать высокий уровень гарантийного запаса, не ниже расчетного уровня):  $t_3 - t_2 = t_2 - t_1$ .

### 6.3. Математическое обеспечение задачи «Выбор порядка обработки деталей на двух станках»

В подсистеме оперативного управления цехом машиностроительного предприятия решаются задачи оперативно-календарного планирования работы технологического оборудования. Если на производстве есть различные станки и одна деталь последовательно обрабатывается на нескольких станках, то могут возникнуть большие непроизводительные простои станков. Чтобы уменьшить эти простои разрабатывается оптимальный план-график обработки деталей на станках. Этот план должен обеспечивать минимизацию общего времени простоя станков. Решить описанную задачу при большом количестве станков и деталей сложно, требуется применение специальных методов оптимизации. В случае, если есть два станка ( $A$  и  $B$ ), известно время обработки детали на каждом станке и каждая деталь обрабатывается сначала на станке  $A$ , а затем на станке  $B$ , можно применить алгоритм Джонсона:

1) находим наименьшее время обработки детали; если меньше всего времени на обработку затрачивается на станке  $A$ , то эта деталь обрабатывается первой на станке  $A$ ; если же на станке  $B$ , то деталь обрабатывается последней на станке  $B$ ;

2) исключаем эту деталь из дальнейшего рассмотрения и возвращаемся к пункту 1 алгоритма.

Если порядок обработки всех деталей определен, то получится оптимальный план-график. Если время обработки двух различных деталей на одном станке совпадает (при условии, что это время меньше времени обработки на другом станке), то порядок обработки этих деталей может быть произвольным. Рассмотрим план-график для двух станков и шести деталей (табл. 6.1).

Таблица 6.1

План-график обработки деталей

Деталь	Станок	
	$A$	$B$
1	8	4
2	6	7
3	5	5
4	7	4
5	3	6
6	9	8

В соответствии с алгоритмом порядок обработки деталей следующий: деталь 5 обрабатывается на станке *A*; деталь 1 – на *B*; деталь 4 – на *B*; деталь 3 – на *A* (или *B*); деталь 2 – на *A*; деталь 6 – на *B*. Порядок обработки деталей: 5, 3, 2, 6, 4, 1.

В соответствии с этим порядком получим план-график обработки деталей (рис. 6.9).

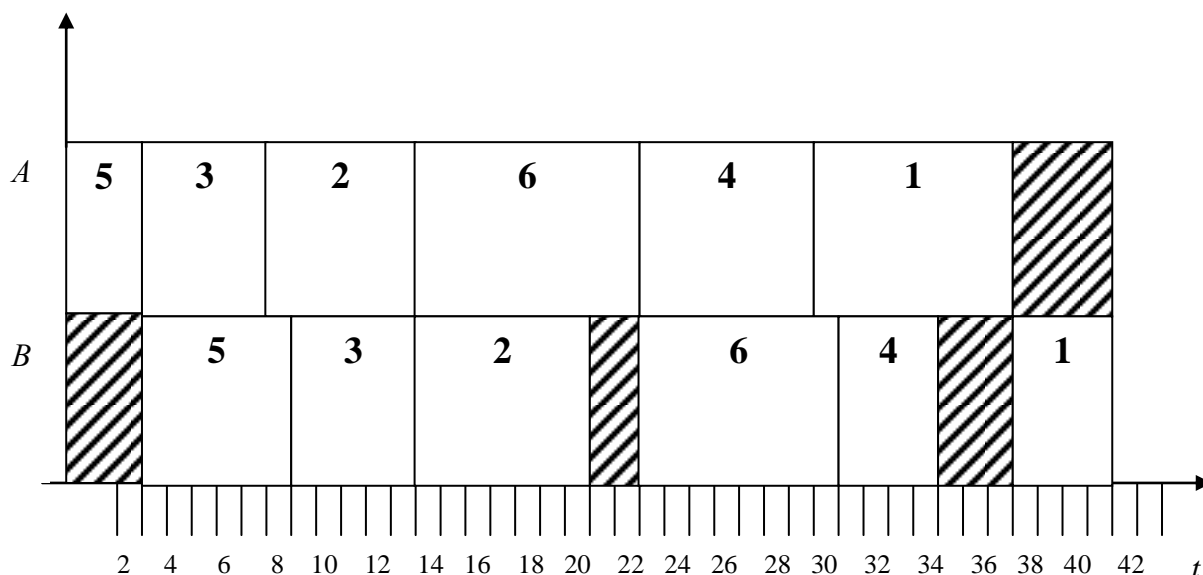


Рис. 6.9

Общий простой: станок *A* – 4 мин, станок *B* – 8 мин.

Существуют разные оптимальные решения этой задачи, поскольку время обработки некоторых деталей может совпадать. Возможны другие варианты порядка обработки деталей: 5, 2, 6, 3, 1, 4 или 5, 3, 2, 6, 1, 4.

Следует отметить, что алгоритм Джонсона дает оптимальное решение для любого числа деталей, но число станков при этом строго ограничено (их должно быть два). Выбор порядка обработки деталей более чем на двух станках значительно усложняет задачу. В этом случае используют метод полного перебора вариантов или специальные комбинаторные методы.

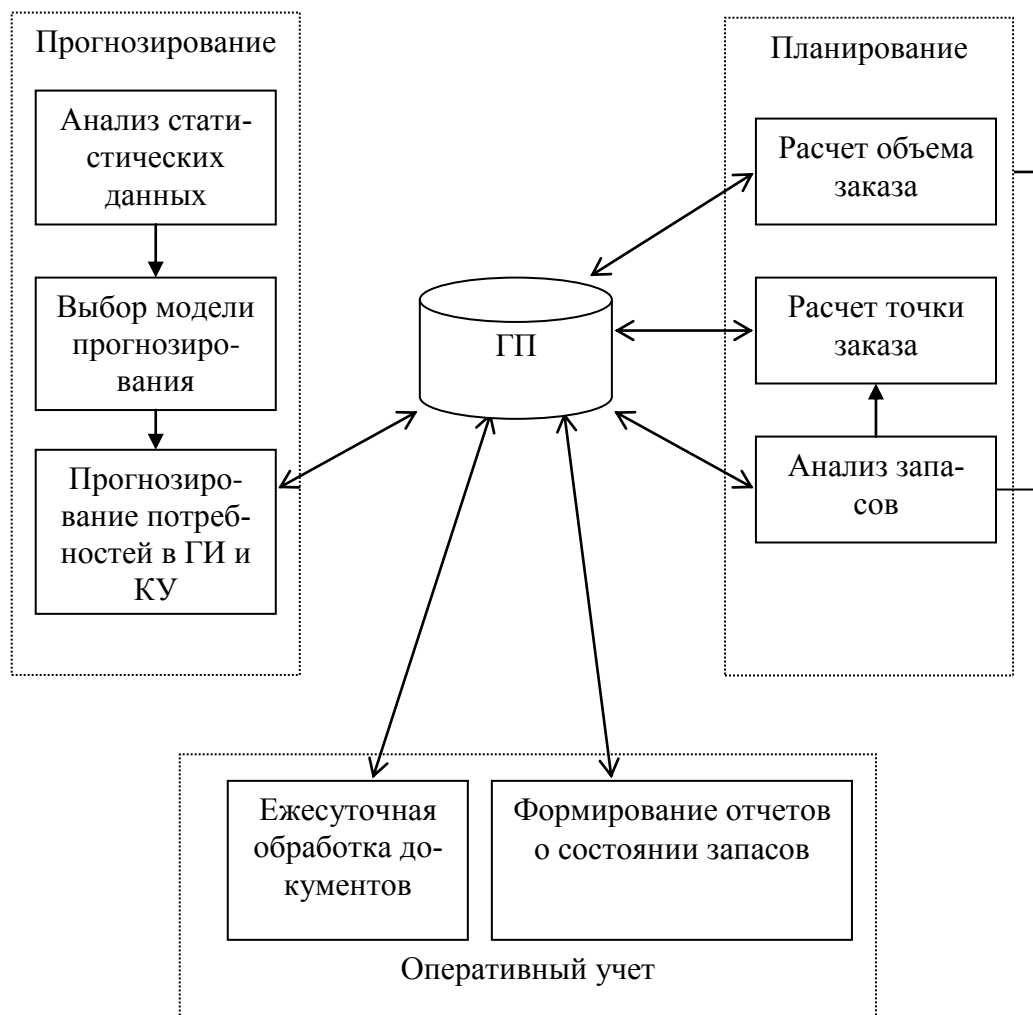
#### 6.4. Подсистема оперативного управления основным производством машиностроительного предприятия

Рассмотрим структуру и функции подсистемы оперативного управления основным производством предприятия с дискретным характером производства. К ним относятся машиностроительные, приборостроительные и другие предприятия, где собирают изделия из деталей и комплектующих. Процесс сборки изделия и изготовления деталей отражается в подсистеме оперативного управления основным производством. В подсистеме решается ряд основных задач.



## 1. Управление запасами (структурная схема – на рис. 6.10).

Назначение: составление прогноза потребностей производства в комплектующих, планирование запасов предприятия, оперативный учет этих запасов.



ГП как главный предметный массив содержит все изделия и детали (наименования и характеристики), которые есть на производстве. Необходимо определить уровень запасов, необходимых производству. Сначала определяют их количество для производства (потребность). Для этого анализируют статистические данные за прошлый год, выбирают модель прогнозирования и осуществляют прогноз. После этого запускается модуль планирования и управления запасами, рассчитывается уровень гарантийного (страхового) запаса, чтобы производство не остановилось, точки и объем заказа. Оперативный учет: обрабатывают документы о состоянии запасов, формируют отчеты.

## 2. Планирование потребностей.

Назначение: формирование плановых заказов производству, т. е. планирование количества сырья, материалов, полуфабрикатов, комплектующих для выполнения заказов. Входные данные: ГП, СИ (массив состава

ва изделий, где указываются входящие в данное изделие детали) (рис. 6.11).

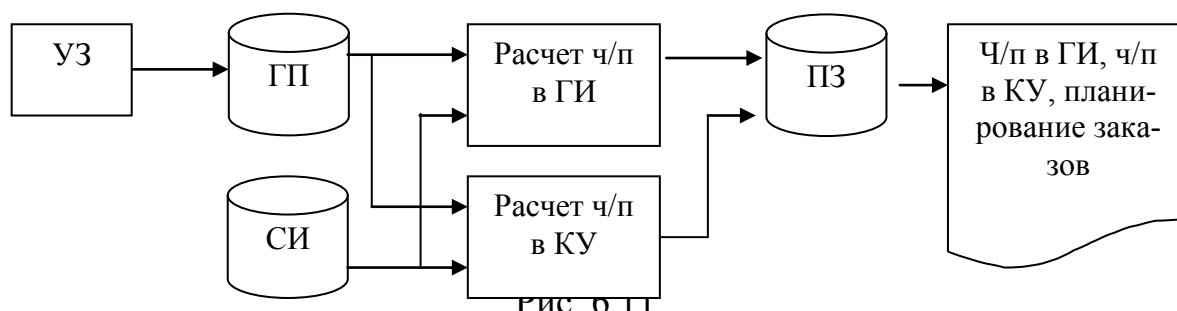


Рис. 6.11

Рассчитываются соответствующие документы (ч/п, готовые изделия, заказы).

Алгоритм определения ч/п:

обращаются к ГП, определяются потребности производства на заданный интервал времени, анализируются запущенные заказы в производство;

определяется наличие запущенных на данный момент времени заказов, определяются чистые потребности, которые равны общие «минус» наличные (запущенные);

планируются заказы.

### 3. Планирование мощности.

Назначение: расчет загрузки рабочих мест планом выпуска изделий в соответствии с ПЗ и производственной программой.

Функции ПМ: расчет календарных пооперационных графиков запуска изделий в производство; вычисление загрузки для каждого рабочего места (рис. 6.12); анализ выполнения плана загрузки по рабочим местам; составление отчетов по загрузке рабочих мест (рис. 6.13).

Дата производственного запуска каждого изделия ( $D_{зан1}, D_{зан2}$ ) определяется исходя из даты запуска предыдущего изделия на этом рабочем месте:  $D_{зан2} = D_{зан1} + t_{ож} + t_{изг} + t_{мо}$ , где  $t_{ож}$  – время ожидания;  $t_{изг}$  – время изготовления;  $t_{мо}$  – межоперационное время.

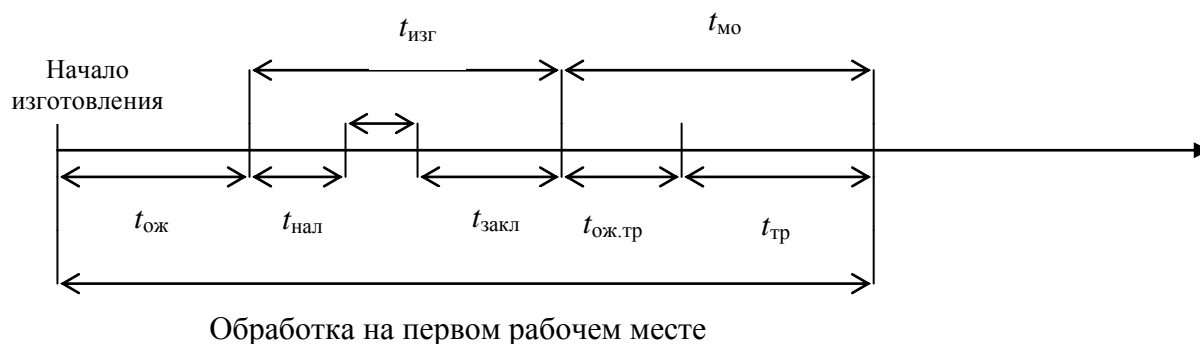


Рис. 6.12

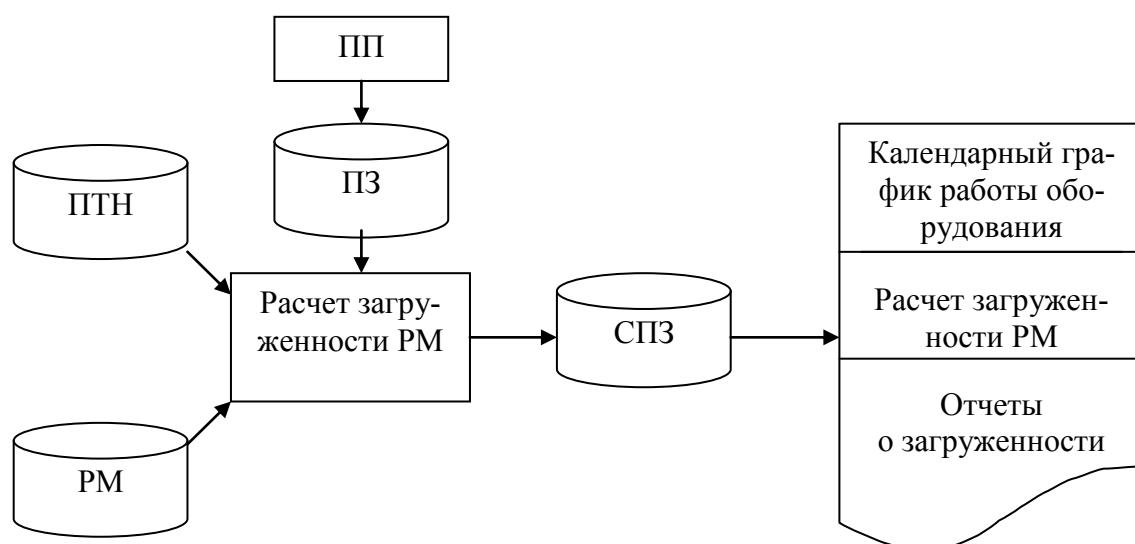


Рис. 6.13

Массив рабочих мест содержит характеристики рабочих мест.

#### 4. Управление цехом.

Назначение: выдача оперативной информации о работе цеха и отдельных рабочих мест и запуск СПЗ в производство с учетом фактического состояния оборудования и наличия комплектующих изделий (рис. 6.14).

Функции: формирование производственных заданий по операциям для рабочих мест; проверка обеспеченности запущенных в производство заказов сырьем, материалами, комплектующими; определение дефицитных материалов, из-за которых могут произойти простои оборудования; оперативный учет изготовления деталей и готовой продукции; выдача информации по запросам в ходе работы цеха.

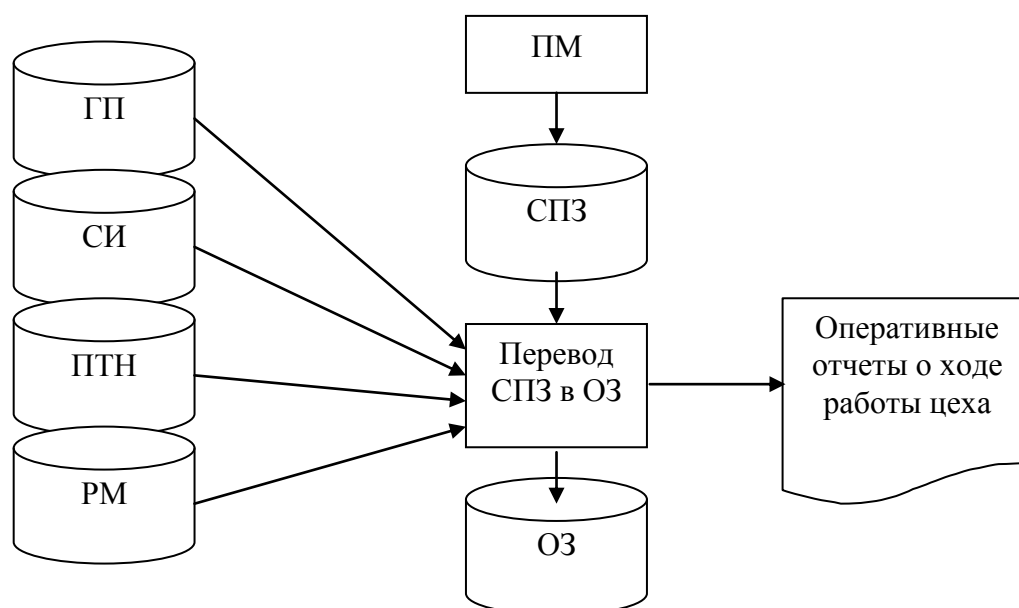


Рис. 6.14

Система интегрированной обработки данных представлена на рис. 6.15.

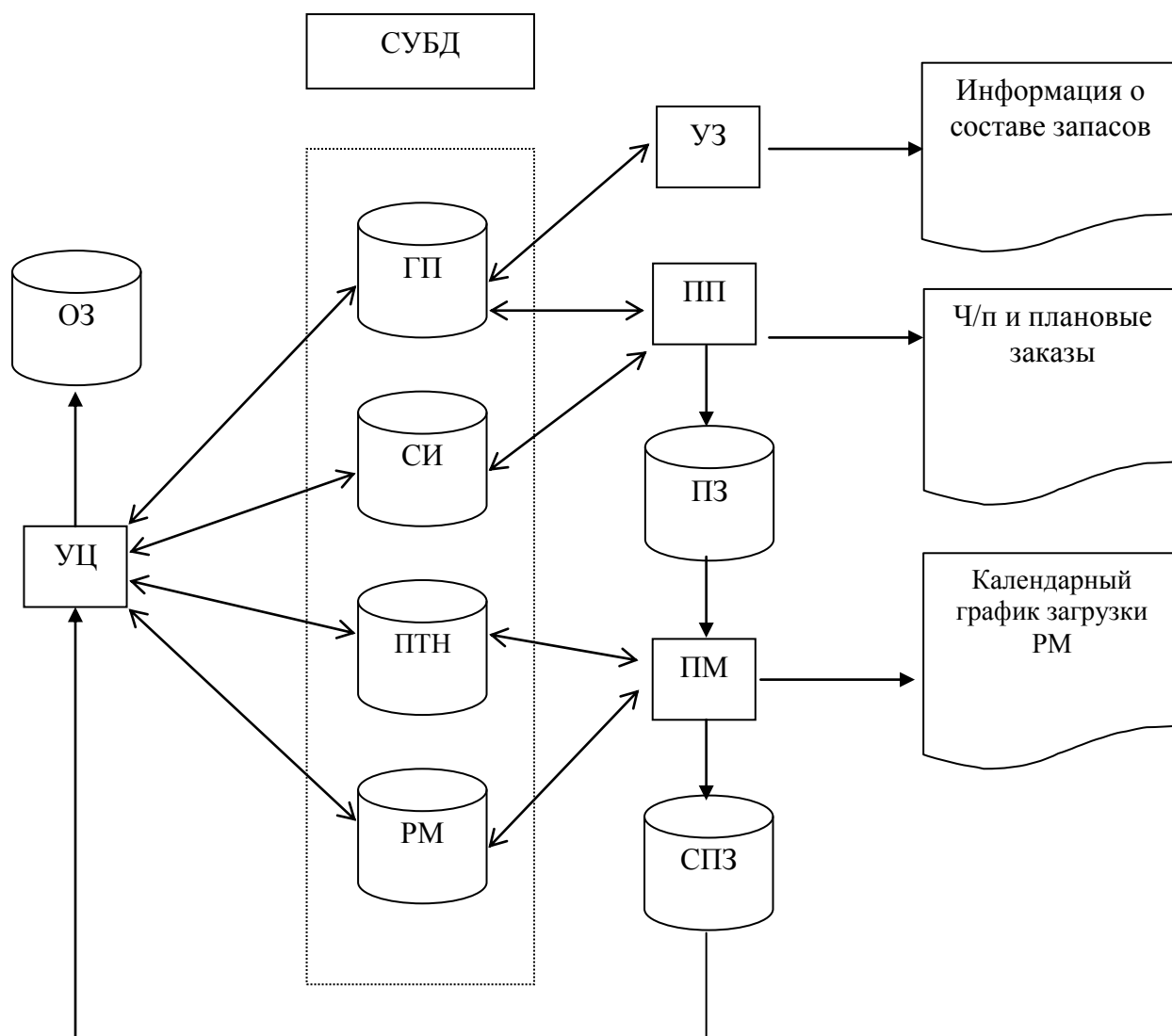


Рис. 6.15

## ПРИЛОЖЕНИЕ

### Тест

1. Записью называется:
  - а) элемент файла;
  - б) набор реквизитов;
  - в) элементарная единица информации.
2. Отношение  $R(A, B)$ , где  $A, B$  – реквизиты, имеет максимально:
  - а) первую нормальную форму;
  - б) вторую нормальную форму;
  - в) третью нормальную форму.
3. По результатам измерений среднее время выполнения одного запроса на ЭВМ равно 2 мин, интенсивность потока запросов – 20 запросов в час. Коэффициент использования ЭВМ равен:
  - а)  $4/5$ ;
  - б)  $3/4$ ;
  - в)  $2/3$ .
4. Доля выборки записей при одном запросе к файлу из 10 тысяч записей составляет в среднем 40 %. Файл должен быть организован:
  - а) прямым методом;
  - б) индексным методом;
  - в) последовательным методом.
5. Какой метод организации данных позволяет быстрее всего реализовать операцию корректировки записей:
  - а) древовидный;
  - б) цепной;
  - в) последовательный.
6. Какой метод организации данных использует меньший объем дополнительной памяти:
  - а) последовательный с индексом для каждой записи;
  - б) цепной;
  - в) древовидный.
7. Синонимами называются записи:
  - а) с одинаковым значением адресной функции;
  - б) одинаковым значением ключа;
  - в) одинаковыми значениями всех реквизитов.
8. Может ли иерархическая БД

корневое отношение – БРИГАДА;  
зависимое отношение – РАБОЧИЙ

содержать информацию о человеке, который трудится индивидуально, вне бригады:

- а) да;
- б) нет;
- в) да, если такой случай единственный в БД.

9. Даны отношения

<i>R</i>	
<i>A</i>	<i>B</i>
8	6
1	2
2	2
1	1

<i>S</i>	
<i>C</i>	<i>D</i>
1	2
3	2
4	3
4	1

Сколько строк содержит результат соединения  $R$  и  $S$  по условию  $A <> C$ :

- а) 8;
- б) 10;
- в) 14.

10. Какой процесс обработки данных выполняется за время, пропорциональное  $M \cdot \log(M)$ :

- а) сортировка;
- б) поиск;
- в) корректировка.

11. Даны функциональные зависимости  $A \rightarrow B, C$ ;  $B \rightarrow D$ ;  $C \rightarrow E$ .

Какая из функциональных зависимостей является их следствием:

- а)  $A \rightarrow D$ ;
- б)  $B \rightarrow C$ ;
- в)  $B \rightarrow E$ .

12. Отношение  $R(A, B, C)$  имеет ключ  $A$ ,  $B$ , отношение  $S(B, D)$  – ключ  $B$ . Результат соединения отношений  $R$  и  $S$  по условию  $B = B$  имеет ключ:

- а)  $B$ ;
- б)  $A, B$ ;
- в)  $A, B, C, D$ .

13. Выберите правильные утверждения:

- а) двухуровневая сетевая структура базы данных может содержать циклические компоненты;
- б) любая функциональная зависимость в реляционной базе данных выводима из минимального покрытия этих функциональных зависимостей;
- в) если в отношении нет функциональных зависимостей, то в нем нет вероятного ключа.

14. Даны отношения в третьей нормальной форме:

$R(A, B, C)$  с ключом  $A$ ;

$S(D, E)$  с ключом  $D$ ;

$T(A, D, F, \text{ДАТА})$  с ключом  $A, D, F$ .

В отношение  $T$  добавлен реквизит ГОД. Третья нормальная форма отношений  $R, S, T$ :

- а) сохранится;
- б) исчезнет;
- в) ответ зависит от наличия функциональных зависимостей  $A \rightarrow \text{ГОД}$  и  $D \rightarrow \text{ГОД}$ .

15. Даны отношения  $R(A, B)$ ,  $T(B, C)$ ,  $Q(A, C)$ . Знаком «\*» обозначена операция соединения отношений по равенству реквизитов. Равенство  $(R * T) * Q = R * (T * Q)$ :

- а) всегда соблюдается;
- б) никогда не соблюдается;
- в) ответ зависит от смысла реквизитов  $A, B, C$ .

16. Какая модель данных разрешает участие приведенного отношения в качестве основного в одном всеерном отношении и одновременно в качестве зависимого – в другом:

- а) иерархическая;
- б) двухуровневая сетевая;
- в) реляционная.

17. Дано отношение с реквизитами ЗАВОД, ПРОДУКЦИЯ, ЦЕНА, ДАТА, ВЫПУСК. Первичным ключом будет набор реквизитов:

- а) ПРОДУКЦИЯ, ЦЕНА, ДАТА;
- б) ЗАВОД, ДАТА;
- в) ЗАВОД, ПРОДУКЦИЯ, ДАТА;
- г) ЗАВОД, ПРОДУКЦИЯ;
- д) ЗАВОД, ПРОДУКЦИЯ, ВЫПУСК.

18. Дано отношение с реквизитами СТУДЕНТ, ДИСЦИПЛИНА, ДАТА, ПРЕПОДАВАТЕЛЬ, ОЦЕНКА. Первичным ключом будет набор реквизитов:

- а) ДИСЦИПЛИНА, ДАТА;
- б) СТУДЕНТ, ДИСЦИПЛИНА, ДАТА;
- в) ДИСЦИПЛИНА, ПРЕПОДАВАТЕЛЬ;
- г) СТУДЕНТ, ПРЕПОДАВАТЕЛЬ, ОЦЕНКА;
- д) СТУДЕНТ, ДИСЦИПЛИНА, ОЦЕНКА.

19. Дано отношение с реквизитами РАБОЧИЙ, ДЕТАЛЬ, ВЕС ДЕТАЛИ. ДАТА, КОЛИЧЕСТВО. Первичным ключом будет набор реквизитов:

- а) ДЕТАЛЬ, ДАТА, КОЛИЧЕСТВО;
- б) РАБОЧИЙ, ДЕТАЛЬ;
- в) РАБОЧИЙ, ДЕТАЛЬ, ДАТА;
- г) ДЕТАЛЬ, ДАТА;
- д) РАБОЧИЙ, ДЕТАЛЬ, КОЛИЧЕСТВО.

20. Отрицанием запроса к БД  
ДОЛЖНОСТЬ = «ИНЖЕНЕР» И ЗАРПЛАТА > 6000 ИЛИ СТАЖ > 20  
является запрос:

- а) ДОЛЖНОСТЬ  $\neq$  «инженер» или ЗАРПЛАТА  $\leq 6000$  и СТАЖ  $\leq 20$ ;
- б) ДОЛЖНОСТЬ  $\neq$  «инженер» и ЗАРПЛАТА  $\leq 6000$  или СТАЖ  $\leq 20$ ;
- в) ДОЛЖНОСТЬ  $\neq$  «инженер» или ЗАРПЛАТА  $\leq 6000$  и СТАЖ  $\leq 20$ ;
- г) ДОЛЖНОСТЬ  $\neq$  «инженер» или ЗАРПЛАТА  $\leq 6000$  и СТАЖ  $\leq 20$ ;
- д) ДОЛЖНОСТЬ  $\neq$  «инженер» и ЗАРПЛАТА  $\leq 6000$  и СТАЖ  $\leq 20$ .

21. Переместившееся на максимальное расстояние значение в столбце «Вес» в БД

Номер записи	ФИО	Вес
1	Иванов	67
2	Ликин	75
4	Сидоров	74
4	Асеев	74
5	Сизов	68
6	Сытин	66

после сортировки таблицы по реквизиту ФИО в порядке возрастания равно:

- а) 66;
- б) 67;
- в) 74;
- г) 75;
- д) 68.



22. Отрицанием запроса к БД (МАРКА = «Ауди» ИЛИ ЦЕНА > 19 000) И ЦВЕТ = «черный» является запрос:

- а) МАРКА ≠ «Ауди» или ЦЕНА ≤ 19 000 и ЦВЕТ ≠ «черный»;
- б) МАРКА ≠ «Ауди» и ЦЕНА ≤ 19 000 или ЦВЕТ ≠ «черный»;
- в) МАРКА ≠ «Ауди» и ЦЕНА > 19 000 или ЦВЕТ = «черный»;
- г) МАРКА ≠ «Ауди» или ЦЕНА ≤ 19 000 и ЦВЕТ = «черный»;
- д) МАРКА ≠ «Ауди» и ЦЕНА > 19 000 и ЦВЕТ ≠ «черный».

23. Операциями над структурой документа являются:

- а) нормализация;
- б) декомпозиция;
- в) инверсия;
- г) свертка;
- д) композиция.

24. Сколько показателей в базе данных с реквизитами: ДАТА, НАИМЕНОВАНИЕ СУДНА, СКОРОСТЬ ХОДА, ГРУЗО-ПОДЪЕМНОСТЬ, ДАТА НАЧАЛА ЗАДАНИЯ, ПОРТ ОКОНЧАНИЯ ЗАДАНИЯ, ВИД ВАЛЮТЫ, СУММА, ДОХОД ЗА ЗАДАНИЕ, РАСХОД ЗА ЗАДАНИЕ, ИМЯ АРЕНДАТОРА, ДАТА ПРИБЫТИЯ В ПОРТ, ДАТА ОТПРАВЛЕНИЯ ИЗ ПОРТА, НАИМЕНОВАНИЕ ПОРТА ЗАХОДА, КОД ГРУЗА, ОПЕРАЦИЯ НАД ГРУЗОМ, ОПЕРАЦИЯ В ПОРТУ, ВЕС ГРУЗА, ИМЯ ГРУЗОВЛАДЕЛЬЦА, ДАТА ГОТОВНОСТИ ГРУЗА К ПЕРЕВОЗКЕ, ДАТА ДОСТАВКИ ГРУЗА ПОЛУЧАТЕЛЮ, ПОРТ ОТПРАВЛЕНИЯ ГРУЗА. Порт назначения груза.

- а) 3;
- б) 4;
- в) 5;
- г) 6.

25. Декомпозиция показателя на несколько показателей:

- 1) всегда возможна;
- 2) невозможна;
- 3) ответ зависит от смысла атрибутов, входящих в показатель.

26. Выберите правильные утверждения.

- а) структуру базы данных экономической ИС можно представить как структуру одной составной единицы информации;
- б) атрибут не имеет структуры, потому что его значение не разделяется на поименованные части;
- в) ролевые атрибуты могут иметь разные области определения.

27. Даны три БД:

БД1 (ОТДЕЛ, ТЕЛЕФОН, ФИО СОТРУДНИКА, ТЕМА РАБОТЫ);  
БД2 (АПТЕКА, ТЕЛЕФОН, ЛЕКАРСТВО, СТРАНА-ИЗГОТОВИТЕЛЬ);

БДЗ (СУДНО, ГРУЗОПОДЪЕМНОСТЬ, ПОРТ ПРИПИСКИ, СУДОВЛАДЕЛЕЦ).

Не содержат атрибутов-оснований:

- а) БД1;
- б) БД2;
- в) БД3;
- г) БД1 и БД2;
- д) БД1 и БД3;
- е) БД2иБД3;
- ж) все БД.

28. Суммирование итогов является корректным:

а) по атрибуту, полученному в результате выборки по всей базе данных;

- б) атрибуту-основанию, расположенному в файле;
- в) атрибуту-основанию, расположенному в показателе.

29. Не имеет экономического смысла БД, содержащая только:

- а) атрибуты-признаки;
- б) атрибуты-основания;
- 3) атрибуты с логическими значениями.

30. Отдельное сообщение:

- а) обязательно должно иметь форму экономического показателя;
- б) не обязательно должно иметь форму экономического показателя;
- в) обязательно должно быть форматированным.

31. Операциями над значениями СЕИ являются:

- а) сортировка;
- б) выборка;
- в) транспонирование;
- г) корректировка.

32. Сколько реквизитов не входят ни в один показатель в БД с реквизитами: НАЗВАНИЕ ОБЪЕКТА, ГОД ИЗГОТОВЛЕНИЯ, ИНВЕНТАРНЫЙ НОМЕР, ДАТА ЛИКВИДАЦИИ, ПЕРВОНАЧАЛЬНАЯ СТОИМОСТЬ, СУММА ИЗНОСА, ОСТАТОЧНАЯ СТОИМОСТЬ:

- а) 0;
- б) 1;
- в) 2;
- г) 3.

33. Выберите правильные утверждения:

а) существует эквивалентное преобразование двухуровневой сети в реляционное представление;

б) существует эквивалентное преобразование двухуровневой сети в одну иерархическую базу данных;

в) существует эквивалентное преобразование многоуровневой сети в одну иерархическую базу данных.

34. Отношения  $T*S$  и  $S*T$ :

1) равны как множества;

2) не равны как множества;

3) содержат разное количество строк.

35. Пары связей (ИЗДЕЛИЕ, КОЛЕСО), (ИЗДЕЛИЕ, АВТОМОБИЛЬ), (ТРАНСПОРТ, АВТОМОБИЛЬ), (ТРАНСПОРТ, САМОЛЕТ) представляют собой структуру:

а) сущностей и связей;

б) событий;

в) понятий.

36. Какая модель представления знаний реализует наследование свойств:

а) фреймовая;

б) семантической сети;

в) реляционная.

37. Какая модель представления знаний использует присоединенные процедуры:

а) продукционная;

б) фреймовая;

в) семантической сети.

38. Связь между понятиями «инженер» и «специалист» имеет вид:

а) есть некоторый;

б) есть часть;

в) другой вид.

39. Связь между понятиями «отдел» и «учреждение» имеет вид:

а) есть некоторый;

б) есть часть;

в) другой вид.

40. В БД с реквизитами:

Регистрационный номер экспоната. Название экспоната. Автор экспоната. Дата поступления. Дата выбытия. Дата реставрации, ФИО реставратора. Оценочная стоимость экспоната. Название выставки. Период проведения выставки, ФИО владельца экспоната, реквизит «Автор экспоната» представляет:

а) объект;

- б) свойство объекта;
- в) взаимодействие объектов;
- г) свойство взаимодействия.

41. В БД с реквизитами НАИМЕНОВАНИЕ ТОВАРА, КОД ТОВАРА, ЕДИНИЦА ИЗМЕРЕНИЯ, ЦЕНА, НОРМА ЗАПАСА, ФИО ТОРГОВОГО АГЕНТА, НОМЕР ТОРГОВОГО АГЕНТА, НОМЕР ЗАКАЗА, ДАТА ОФОРМЛЕНИЯ ЗАКАЗА, ДАТА ИСПОЛНЕНИЯ ЗАКАЗА, КОЛИЧЕСТВО ЗАКАЗАННОГО ТОВАРА, ФИО КЛИЕНТА, АДРЕС И ТЕЛЕФОН КЛИЕНТА, СУММА ПО ЗАКАЗУ, СУММА ПРЕДОПЛАТЫ реквизит СУММА ПРЕДОПЛАТЫ представляет:

- а) объект;
- б) свойство объекта;
- в) взаимодействие объектов;
- г) свойство взаимодействия.

42. В модели сущностей и связей дуга может соединять:

- а) сущность и сущность;
- б) связь и связь;
- в) сущность и связь.

43. Существует БД с реквизитами ФИО АБИТУРИЕНТА, КОД АБИТУРИЕНТА, ГОД РОЖДЕНИЯ, ПОЛ, СУММАРНЫЙ БАЛЛ ЭКЗАМЕНОВ, КОД ФАКУЛЬТЕТА, НАЗВАНИЕ ФАКУЛЬТЕТА, ФИО ДЕКАНА, КОЛИЧЕСТВО МЕСТ. ПРОХОДНОЙ БАЛЛ, НАЗВАНИЕ ЭКЗАМЕНАЦИОННОЙ ДИСЦИПЛИНЫ, ДАТА СДАЧИ ЭКЗАМЕНА, ОЦЕНКА ЭКЗАМЕНА, КОД ГРУППЫ, ФИО ЗАЧИСЛЕННОГО АБИТУРИЕНТА.

Сколько реквизитов представляют взаимодействие объектов:

- а) 0;
- б) 2;
- в) 3;
- г) 4.

44. Существует БД с реквизитами НАИМЕНОВАНИЕ ТОВАРА, КОД ТОВАРА, ЕДИНИЦА ИЗМЕРЕНИЯ, ЦЕНА, НОРМА ЗАПАСА, ФИО ТОРГОВОГО АГЕНТА, НОМЕР ТОРГОВОГО АГЕНТА, НОМЕР ЗАКАЗА, ДАТА ОФОРМЛЕНИЯ ЗАКАЗА, ДАТА ИСПОЛНЕНИЯ ЗАКАЗА, КОЛИЧЕСТВО ЗАКАЗАННОГО ТОВАРА, ФИО КЛИЕНТА, АДРЕС И ТЕЛЕФОН КЛИЕНТА, СУММА ПО ЗАКАЗУ, СУММА ПРЕДОПЛАТЫ.

Сколько реквизитов представляют свойство взаимодействия:

- а) 0;
- б) 5;
- в) 3;
- г) 4.

45. Правильным утверждением является:

- а) в дереве каждый элемент имеет только один предшествующий элемент;
- б) дерево соответствует связному графу, не содержащему циклов;
- в) неверно, что число записей в дереве всегда меньше, чем произведение порядка дерева на его ранг.

46. В бинарном дереве на рис. 4.13 для записи  $D$  соседней записью является:

- а)  $F$ ;
- б)  $G$ ;
- в)  $J$ .

47. Выберите ошибочные утверждения:

- а) перед включением новой записи необходим поиск ее местоположения относительно существующих записей;
- б) все методы сортировки основаны на сравнении пар ключевых признаков записей;
- в) в упорядоченном бинарном дереве, построенном на основе массива, упорядоченного по невозрастанию ключевых признаков, заполнены только левые адреса связи.

48. Вероятность того, что массив из  $M$  записей случайно окажется упорядоченным, равна:

- а)  $1/M$ ;
- б)  $1/(M!)$ ;
- в)  $2/(M!)$ .

49. Выберите правильные утверждения:

- а) двухуровневая сетевая структура базы данных может содержать циклические компоненты;
- б) любая функциональная зависимость для базы данных является следствием из минимального покрытия этих функциональных зависимостей;
- в) если в отношении нет функциональных зависимостей, то в нем нет вероятного ключа.

50. Для отношения  $R(A, B, C)$  дан запрос « $A = 10$  или  $A <> 10$  и  $B <> 2$ ». Эквивалентным для него запросом является:

- а) «не ( $A = 10$  или  $B = 2$ )»;
- б) «не ( $A = 10$  и  $B = 2$ )»;
- в) « $A = 10$  или  $B <> 2$ »;
- г) « $B <> 2$ ».

51. Сколько вероятных ключей в отношении

*T1*

<i>KOS</i>	<i>KOGR</i>	<i>WK</i>	<i>DUBL</i>	<i>PRIN</i>
<i>AK</i>	61	<i>WA</i>	570	<i>WS</i>
<i>AT</i>	44	<i>PR</i>	751	<i>WA</i>
<i>AK</i>	20	<i>AI</i>	856	<i>RI</i>
<i>AU</i>	74	<i>WA</i>	546	<i>TX</i>
<i>AT</i>	15	<i>PR</i>	751	<i>AX</i>
<i>AK</i>	94	<i>FY</i>	456	<i>PI</i>
<i>AU</i>	40	<i>CM</i>	846	<i>CI</i>
<i>AT</i>	52	<i>VD</i>	751	<i>WS</i>

- а) 2;
- б) 3;
- в) 4;
- г) 1.

52. Составной единицей информации не является:

- а) отношение;
- б) байт;
- в) показатель.

53. Операцией реляционной алгебры не является:

- а) проекция;
- б) соединение;
- в) инверсия;
- г) выборка.

54. Дано отношение

*R*

<i>A</i>	<i>B</i>	<i>C</i>	<i>X</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>x1</i>
<i>a1</i>	<i>b2</i>	<i>c1</i>	<i>x1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>	<i>x1</i>
<i>a2</i>	<i>b1</i>	<i>c1</i>	<i>x1</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>	<i>x2</i>
<i>a2</i>	<i>b2</i>	<i>c2</i>	<i>x2</i>
<i>a3</i>	<i>b2</i>	<i>c1</i>	<i>x3</i>
<i>a3</i>	<i>b2</i>	<i>c2</i>	<i>x3</i>

В отношении *R* соблюдается функциональная зависимость  $A, B, C \rightarrow X$ .  
В проекции  $T = R[B, C, X]$  соблюдается функциональная зависимость:

- а)  $B, C \rightarrow X$ ;
- б)  $B \rightarrow X$ ;
- в) никакая.

55. Видовым по отношению к родовому понятию «Рабочий» является понятие:

- а) «слесарь»;
- б) «технолог»;
- в) «конструктор»;
- г) «литейщик».

56. Даны отношения:

T1(ЦЕХ, КОД ИЗДЕЛИЯ, ПЛАН 2004 ГОДА);

T2(КОД МАТЕРИАЛА, ЦЕНА МАТЕРИАЛА);

T3(КОД ИЗДЕЛИЯ, ЦЕНА ИЗДЕЛИЯ);

T4(КОД МАТЕРИАЛА. КОД ИЗДЕЛИЯ. НОРМА РАСХОДА).

Количество веерных отношений, которые будут установлены при формировании двухуровневой сетевой БД, равно:

- а) 1;
- б) 2;
- в) 3;
- г) 4.

57. Даны отношения:

T1 (МУЗЕЙ, ГОРОД);

T2(МУЗЕЙ, ЭКСПОНАТ, ГОД ПОСТУПЛЕНИЯ);

T3(ЭКСПОНАТ, РЕСТАВРАТОР, ГОД РЕСТАВРАЦИИ);

T4(ЭКСПОНАТ, ОЦЕНОЧНАЯ СТОИМОСТЬ).

Количество веерных отношений, которые будут установлены при формировании двухуровневой сетевой БД, равно:

- а) 1;
- б) 2;
- в) 3;
- г) 4.

58. Выберите правильные утверждения:

а) если в отношении продублировать часть строк, то оно будет представлять то же множество сообщений, что и до преобразования;

б) если в отношении продублировать часть столбцов, то оно будет представлять то же множество сообщений, что и до преобразования;

в) если в отношении заменить текстовые значения объектов их цифровыми кодами, то оно будет представлять то же множество сообщений, что и до преобразования.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Палюх, Б.В. Применение современных информационных технологий для разработки информационных систем: учеб. пособие для студентов вузов по специальности 080801 «Прикладная информатика» и других экономических специальностей / Б.В. Палюх, В.В. Алексеев, А.Ю. Ключин, С.В. Котлинский. Тверь: ТГТУ, 2010. 175 с.
2. Палюх, Б.В. Информационные технологии в управлении экономикой: учеб. пособие / Б.В. Палюх, С.Л. Федченко, ЮГ. Козлова, А.Н. Прохныч. Тверь: ТГТУ, 2007. 127 с.
3. Котлинский, С.В. Разработка моделей систем на базе современных методов и сред моделирования: учеб. пособие. Ч. 1 / С.В. Котлинский, А.Ю. Ключин. 1-е изд. Тверь: ТвГТУ, 2011. 124 с.
4. Палюх, Б.В. Применение современных языков и инструментов для моделирования предметной области автоматизации: учеб. пособие. Ч. 1 / Б.В. Палюх, С.В. Котлинский, А.Ю. Ключин. Тверь: ТвГТУ, 2013. 187 с.
5. Палюх, Б.В. Применение современных языков и инструментов для моделирования предметной области автоматизации: учеб. пособие. Ч. 2 / Б.В. Палюх, С.В. Котлинский, А.Ю. Ключин. Тверь: ТвГТУ, 2013. 172 с.
6. Берзин, Е.А. Элементарные решения неэлементарных задач на графах: учеб. пособие / Е.А. Берзин, А.Н. Кудинов; под ред А.Н. Кудинова. Тверь: ТвГТУ, 2005. 136 с.
7. Котов, С.Л. Методы оценки, тестирования и выбора рациональных характеристик корпоративных информационных систем: учеб. пособие / С.Л. Котов, Б.В. Палюх, С.Л. Федченко. 1-е изд. Тверь: ТвГТУ, 2008. 204 с.
8. Мишенин, А.И. Теория экономических информационных систем. Практикум: учеб. пособие / А.И. Мишенин, С.П. Салмин. М.: Финансы и статистика, 2005. 192 с.
9. Мишенин, А.И. Теория экономических информационных систем: учебник / А.И. Мишенин. М.: Финансы и статистика, 2002. 240 с.
10. Смирнова, Г.Н. Проектирование экономических информационных систем: учебник для студентов экономических вузов / Г.Н. Смирнова, А.А. Сорокин, Ю.Ф. Тельнов; под ред. Ю.Ф. Тельнова. М.: Финансы и статистика, 2005. 510 с.
11. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: учебник для экономических вузов по специальности «Прикладная информатика» и «Прикладная математика и информатика» / А.М. Вендров. М.: Финансы и статистика, 2002. 352 с.
12. Вендров, А.М. Практикум по проектированию программного обеспечения экономических информационных систем: учеб. пособие для вузов по специальностям «Прикладная информатика в экономике», «Математическое обеспечение и администрирование информационных систем» / А.М. Вендров. М.: Финансы и статистика, 2006.



