

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**Л. Г. Высоцкий**

## **ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ПИ)**

Учебно-методическое пособие по выполнению лабораторных работ  
для студентов заочного отделения, обучающихся в бакалавриате по направлению  
подготовки 09.03.03 Прикладная информатика (заочное отделение)

Калининград  
Изд-во ФГБОУ ВО «КГТУ»  
2022

УДК 004.9(075)

Рецензент:

кандидат педагогических наук, доцент кафедры прикладной информатики  
ФГБОУ ВО «Калининградский государственный технический  
университет» Е. Ю. Заболотнова

**Высоцкий, Л. Г.**

**Программная инженерия (ПИ):** учебно-методическое пособие по выполнению лабораторных работ для студентов заочного отделения, обучающихся в бакалавриате по направлению подготовки 09.03.03 Прикладная информатика / **Л. Г. Высоцкий.** – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 60 с.

Учебно-методическое пособие представляет собой указания по выполнению лабораторного практикума по дисциплине «Программная инженерия» для студентов заочного отделения. Практикум включает четыре лабораторные работы, которые выполняются строго по индивидуальному заданию. Каждая работа предваряется теоретическим введением, в котором изложен ход выполнения работы. В пособие также включены подробные указания по оформлению отчетов по лабораторным работам и контрольные вопросы к ним.

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой прикладной информатики института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 19 сентября 2022 г., протокол № 3

Учебно-методическое пособие рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией ИЦТ 20 сентября 2022 г., протокол № 6

УДК 004.9(075)

© Федеральное государственное бюджетное  
образовательное учреждение  
высшего образования  
«Калининградский государственный  
технический университет», 2022 г.  
© Высоцкий Л. Г., 2022 г.

## **Оглавление**

ВВЕДЕНИЕ .....	4
Лабораторная работа № 1 Планирование разработки программной системы.....	5
Лабораторная работа № 2 Использование метода экспертных оценок при разработке программных систем .....	18
Лабораторная работа № 3 Формирование информационной модели предметной области .....	30
Лабораторная работа № 4 Изучение стандартов.....	42
ЗАКЛЮЧЕНИЕ .....	54
ЛИТЕРАТУРА:.....	55
Список условных сокращений .....	59

## ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов направления 09.03.03 Прикладная информатика (заочное отделение), изучающих дисциплину «Программная инженерия».

**Цель** лабораторного практикума по дисциплине:

- изучение отечественных и международных стандартов, регламентирующих процесс разработки программных систем в целом, последовательность отдельных его стадий, структуру и содержание проектной документации;
- освоение практических средств планирования и управления процессом разработки программных систем;
- освоение современных подходов к реализации отдельных стадий и задач процесса проектирования;
- изучение CASE-средств, которые стали неотъемлемым атрибутом современного процесса разработки программного обеспечения.

Лабораторный практикум содержит четыре лабораторные работы.

Лабораторные работы проводятся в компьютерных классах КГТУ.

В результате выполнения лабораторных работ ожидается, что студенты сформируют практические навыки разработки больших программных систем.

Все лабораторные работы снабжены обширным теоретическим материалом по изучаемым темам, что дает возможность использовать методические указания по данному практикуму в виде учебного пособия.

Каждая лабораторная работа предусматривает несколько десятков вариантов заданий, что позволяет строго индивидуально контролировать работу студентов во время их выполнения.

## Лабораторная работа № 1 Планирование разработки программной системы

### *Общие сведения*

*Цель работы:* изучение формальных методов планирования разработки программных системы (ПС) с целью оптимизации управления данным процессом.

Работа носит чисто расчетный характер, из CASE-средств используется только любой текстовый редактор для оформления отчета по лабораторной работе.

Для успешной сдачи лабораторной работы необходимо предоставить отчет и пройти защиту в соответствии с разделом «Требования к отчету и защите».

Аудиторное время выполнения (под руководством преподавателя): 3 часа.

Время самостоятельной подготовки: 3 часа.

### *Теоретическое введение*

Разработка ПС представляет собой сложный комплекс различных задач (этапов), решение которых характеризуется высокой трудоемкостью и большой длительностью. Сами этапы, их длительность и взаимосвязи определяются на стадии «Планирование» и оформляются в виде графика работ, который становится руководящим документом для всех участников проекта. Составление данного графика (рис. 1.5, 1.6) - весьма ответственный этап работы менеджеров проекта, поскольку на основании его оценивается длительность проекта, определяются ресурсы, устанавливается согласованность в выполнении этапов.

Все этапы разработки можно выполнять последовательно (рис. 1.1, задачи идентифицированы буквами), один за другим, что определяет общую длительность разработки как арифметическую сумму длительностей отдельных этапов (табл. 1.1).



Рис. 1.1

Таблица 1.1

Операция	A	B	C	D	E	F	G	H	I	J	K	L	Всего
Продолжительность, дни	30	7	10	14	10	10	21	7	12	15	30	15	178

В то же время анализ содержимого отдельных этапов разработки ПС показывает (рис. 1.2), что они находятся в отношении частичного, а не строгого порядка (одни задачи могут выполняться строго последовательно друг за другом, а другие такой

связью не характеризуются и могут выполняться в произвольном порядке по отношению друг к другу), что позволяет сократить общий срок разработки. Графически отношение частичного порядка на множестве отображается в виде сети [1]. Существуют два подхода к построению таких сетей.

Первый из них предполагает, что дуга сети идентифицирует некоторую работу (задачу), а вершины являются событиями, представляющими начало/завершение работ (рис. 1.2).

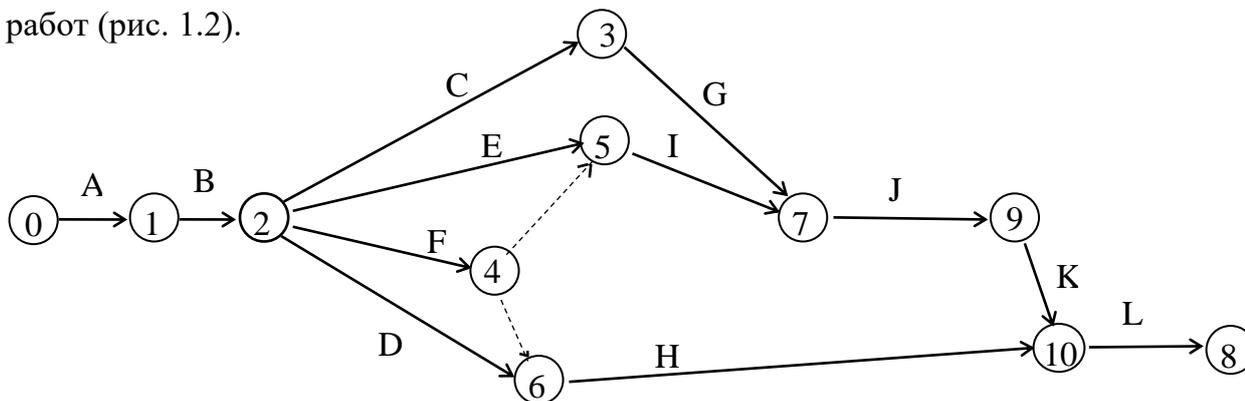


Рис. 1.2

Однако возможен и второй подход к построению сети, отображающей связи работ, при котором различные задачи отображаются вершинами, а дуги показывают связи между ними (отношение предшествования). Достоинства такого подхода:

1. Нет событий, все вычисления относятся непосредственно к работам;
2. Нет необходимости вводить фиктивные работы, так как любое отношение предшествования можно точно представить и без них (в сети «дуга-работа» каждая работа, предшествующая двум или более другим, требует введение фиктивных работ. На рис. 1.2 они представлены пунктирными дугами);
3. Для идентификации работы достаточно одного индекса.

При любом варианте построения модели плана работ методом сетевого планирования и управления (СПУ) можно определить:

- \* ранние и поздние сроки начала и завершения работ;
- \* критический путь, определяющий минимально возможную длительность выполнения всего проекта;
- \* резервы времени (полные, свободные, независимые, гарантированные) в выполнении работ, не лежащих на критическом пути.

В сети минимальная продолжительность выполнения проекта определяется длиной критического пути (выделен на рис. 1.3 жирной линией). Любая задержка на

этом пути приводит к увеличению длительности выполнения всего проекта, поэтому бригада «критического пути» формируется из наиболее опытных разработчиков. Задержка в выполнении этапов, не лежащих на критическом пути, не влияют на общую продолжительность до тех пор, пока длительность всех этапов по этому пути не превысит длительность этапов на критическом пути.

Приведенный ранее анализ обоих подходов к построению сетей делает второй вариант более перспективным при планировании проектов с использованием компьютеров. На рис. 1.3 представлена сеть "узел-работа" для того же проекта, что изображен на рис. 1.2. В неё включены условные работы "Начало" и "Окончание" с нулевой длительностью, чтобы получить единичный начальный и конечный узлы (хотя для данной сети они излишни). Для каждой  $i$ -й вершины водится понятие *предшествующего множества работ*  $S_i$ , как множества работ, непосредственно предшествующих  $i$ -й работе, и *следующего множества работ*  $F_i$  - как множества работ, непосредственно следующего за этой же работой. Далее определяется понятие наиболее раннего возможного срока начала  $i$ -й работы  $ES_i$ , наиболее позднего допустимого срока начала  $i$ -й работы  $LS_i$ , наиболее раннего возможного срока окончания  $i$ -й работы  $EF_i$  и наиболее позднего допустимого срока окончания  $i$ -й работы  $LF_i$ .

Полагаем, что для работы "Начало"  $ES_{\text{начало}} = EF_{\text{начало}} = 0$ . Отсюда следует, что для всех последующих работ  $ES_i = \max_{EF_x \in S_i} [EF_x]$  и  $EF_i = ES_i + d_i$ , где  $d_i$  - продолжительность  $i$ -й работы. Данные вычисления ведутся таким образом, чтобы каждая работа была рассмотрена до её появления в непосредственно предшествующем множестве другой работы.

Итак  $ES_{\text{начало}} = EF_{\text{начало}} = 0$ .

$ES_A = 0$ .  $EF_A = 0 + 30 = 30$ .

$ES_B = 30$ .  $EF_B = 30 + 7 = 37$ .

$ES_C = 37$ .  $EF_C = 37 + 10 = 47$ .

$ES_D = 37$ .  $EF_D = 37 + 14 = 51$ .

$ES_E = 37$ .  $EF_E = 37 + 10 = 47$ .

$ES_F = 37$ .  $EF_F = 37 + 10 = 47$ .

$ES_G = 47$ .  $EF_G = 47 + 21 = 68$ .

$ES_H = \max[47, 51] = 51$ .  $EF_H = 51 + 7 = 58$ .

$ES_I = \max[47, 47] = 47$ .  $EF_I = 47 + 12 = 59$ .

$ES_J = \max[68, 59] = 68$ .  $EF_J = 68 + 15 = 83$ .

$ES_K = 83$ .  $EF_K = 83 + 30 = 113$ .

$ES_L = \max[113, 58] = 113$ .  $EF_L = 113 + 15 = 128$ .

$ES_{\text{окончания}} = EF_{\text{окончания}} = 128$ .

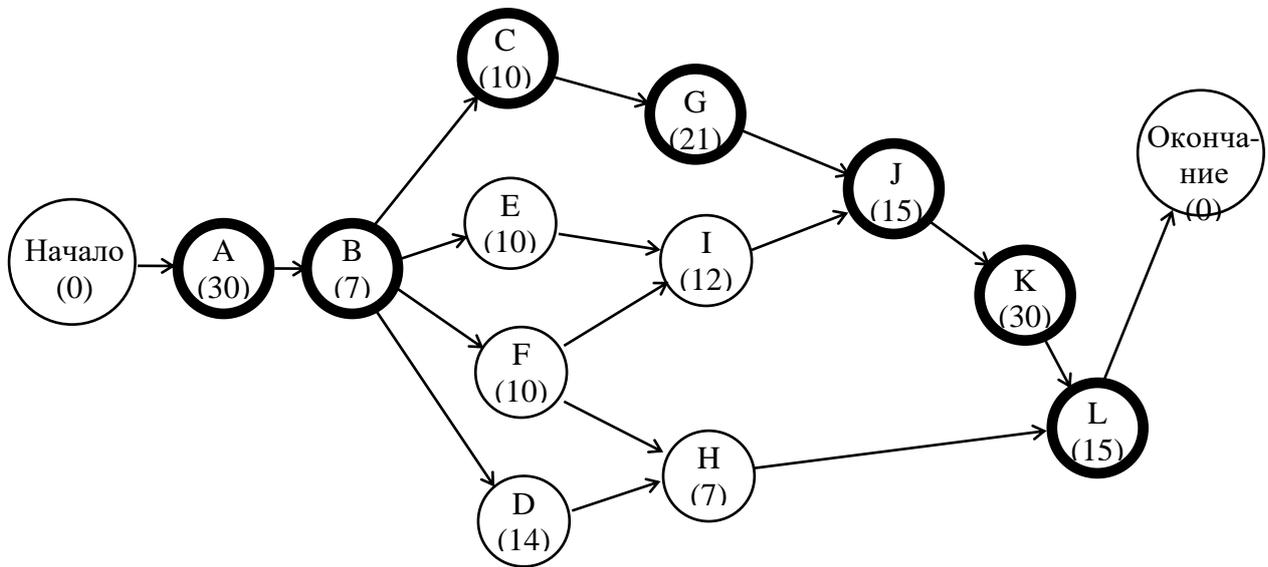


Рис. 1.3

Результаты вычисления наиболее ранних сроков начала/окончания работ для рис. 1.3 приведены в табл. 1.2.

Таблица 1.2

Работа $i$	Продолжительность $d_i$	Наиболее ранний возможный срок		Наиболее поздний допустимый срок		Запас
		Начала, $ES_i$	Окончания, $EF_i$	Начала, $LS_i$	Окончания, $LF_i$	
Начало	0	0	0	0	0	
A	30	0	30	0	30	0
B	7	30	37	30	37	0
C	10	37	47	37	47	0
D	14	37	51	92	106	55
E	10	37	47	46	56	9
F	10	37	47	46	56	9
G	21	47	68	47	68	0
H	7	51	58	106	113	55
I	12	47	59	56	68	9
J	15	68	83	68	83	0
K	30	83	113	83	113	0
L	15	113	128	113	128	0
Окончание	0	128	128	128	128	

Для вычисления наиболее позднего допустимого срока окончания работ необходимо знать плановый срок окончания проекта  $T$ , для которого должно выполняться условие  $T \geq EF_{\text{окончания}}$ . Обычно принимается  $T = EF_{\text{окончания}}$ .

Далее принимается  $LF_{\text{окончания}} = LS_{\text{окончания}} = T$  и для каждой  $i$ -й вершины определяется  $LF_i = \min_{LS_x \in F_i} [LS_x]$ ,  $LS_i = LF_i - d_i$ . Для рис. 1.3 примем  $LF_{\text{окончания}} = LS_{\text{окончания}} = 128$ . Тогда:

$$\begin{array}{ll}
 LF_L = 128. \quad LS_L = 128 - 15 = 113. & LF_K = 113. \quad LS_K = 113 - 30 = 83. \\
 LF_J = 83. \quad LS_J = 83 - 15 = 68. & LF_I = 68. \quad LS_I = 68 - 12 = 56. \\
 LF_H = 113. \quad LS_H = 113 - 7 = 106. & LF_G = 68. \quad LS_G = 68 - 21 = 47. \\
 LF_F = \min[56, 106] = 56. \quad LS_F = 56 - 10 = 46. & LF_E = 56. \quad LS_E = 56 - 10 = 46. \\
 LF_D = 106. \quad LS_D = 106 - 14 = 92. & LF_C = 47. \quad LS_C = 47 - 10 = 37. \\
 LF_B = \min[37, 46, 46, 92] = 37. \quad LS_B = 37 - 7 = 30. & LF_A = 30. \quad LS_A = 30 - 30 = 0. \\
 LF_{\text{начала}} = LS_{\text{начала}} = 0.
 \end{array}$$

Очевидно, что и в данном случае на критическом пути находятся вершины, у которых отсутствуют резервы времени ( $ES_i = EF_i$  и  $LS_i = LF_i$ ). Для рассматриваемого примера критический путь образован вершинами <"Начало", А, В, С, G, J, К, L, "Окончание"> (рис. 1.3).

Резервы времени для вершин сети вычисляются по следующим выражениям:

$$TF_i = LS_i - ES_i \text{ или } TF_i = LF_i - EF_i,$$

т. е. как разность между поздним и ранним началами или поздним и ранним окончаниями.

Далее строится диаграмма Гантта (рис. 1.4), которая визуально представляет все временные параметры будущих работ [2]. Для этого откладывается от раннего начала длительность работы  $d_i$ , а затем, если имеет место, ее резерв времени (запас).

При формировании оптимизационных графиков каждая работа вне критического пути может передвигать и/или растягиваться только в пределах своих раннего начала и позднего окончания. Именно этот диапазон и визуализируется диаграммой Гантта.

Первый вариант календарного плана строится на основе ранних сроков запуска работ на выполнение и ориентирован на максимальное число бригад (групп) разработчиков, которое определяется как максимальное число независимых путей в сети. Для рассматриваемого примера данный календарный план приведен на рис. 1.5. Он рассчитан на четыре группы разработчиков, так как исходная сеть содержит четыре разных пути, включающих только разные вершины (не входят в другие пути). При этом фиксированным по содержанию является только критический путь, в данном случае

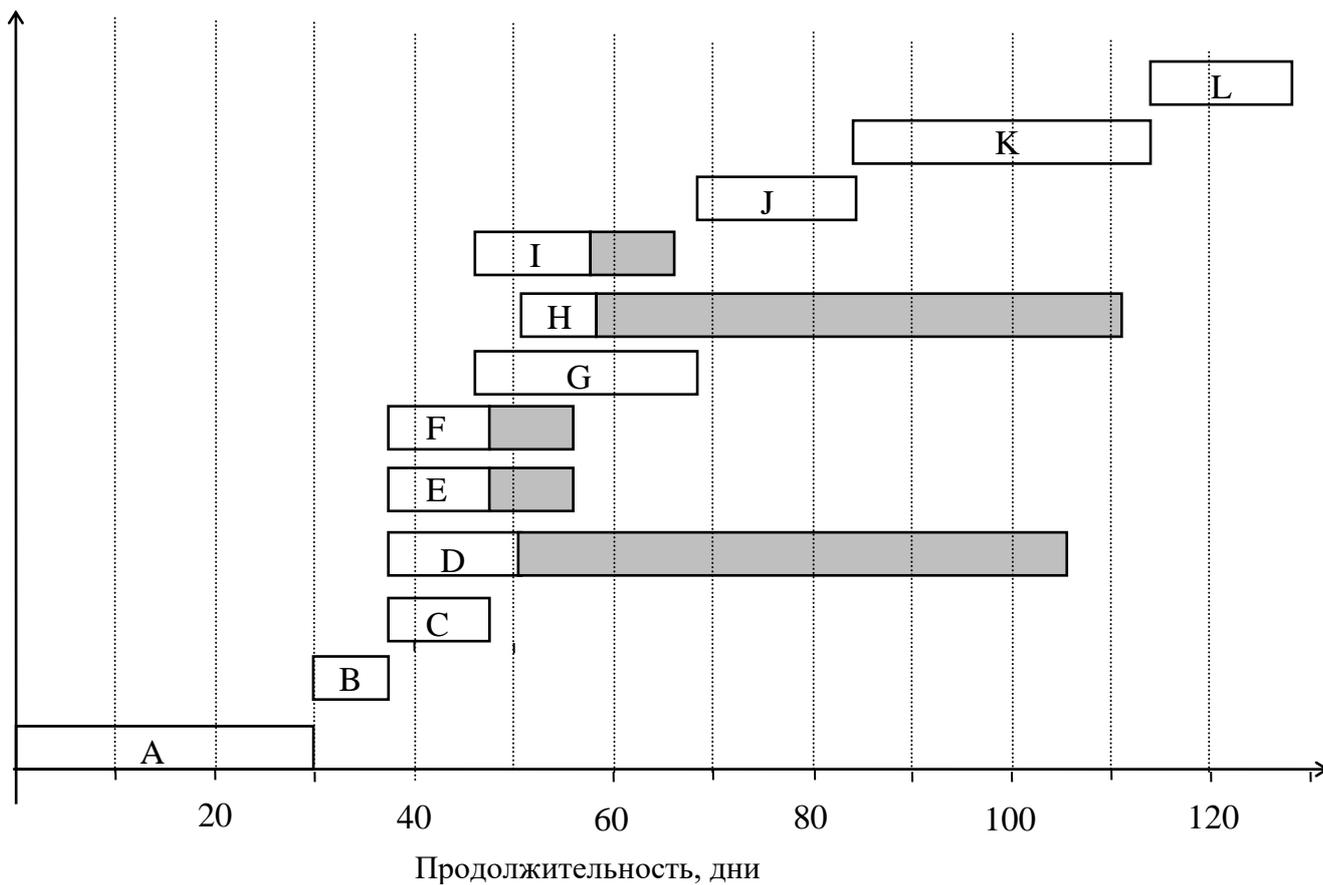


Рис. 1.4

1:  $A \rightarrow B \rightarrow C \rightarrow G \rightarrow J \rightarrow K \rightarrow L$ .

Другие пути могут выглядеть так:

2:  $E \rightarrow I$

3:  $F \rightarrow H$

4: D

или

2: E

3:  $F \rightarrow I$

4:  $D \rightarrow H$

или

2:  $E \rightarrow I$

3: F

4:  $D \rightarrow H$ .

В данном случае выбран вариант

2: E

3:  $F \rightarrow I$

4:  $D \rightarrow H$

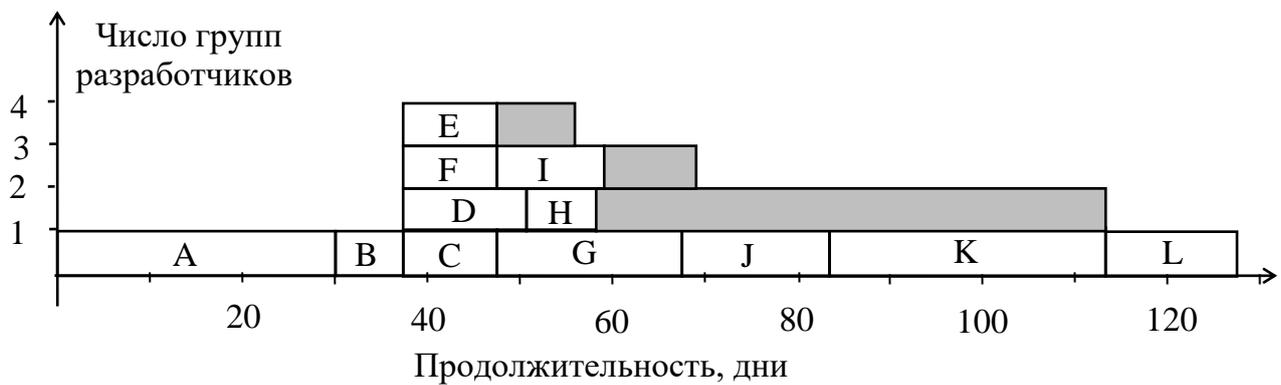


Рис. 1.5

При формировании графика первым откладывается критический путь, остальные пути строятся с учетом следования работ, а также ранних и поздних сроков, имея целью максимизацию возможных запасов (сдвигая работы максимально влево). Получаемая загрузка бригад неравномерна, но позволяет менеджменту маневрировать трудовыми ресурсами.

При минимизации трудовых ресурсов (количества бригад) планирование ведётся по наиболее поздним допустимым срокам окончания работ с целью максимальной загрузки бригад разработчиков и, таким образом, уменьшения их числа. Возможный вариант такого плана представлен на рис. 1.6. Он показывает, что разработка допустима на основе двух бригад. При этом каждую работу можно двигать только в пределах ее раннего и позднего сроков (см. рис. 1.4), стремясь максимизировать оставшиеся резервы.

В данном случае возможен только единственный вариант минимизации: вправо сдвигаются работы D и H, а на их место опускаются сначала E, а только потом F и I. При этом еще сохраняется запас в 23 дня.

Разумеется, будут временные промежутки, когда часть работников некоторых бригад будет не занята, поскольку невозможно составить такой график, при котором у всех задач разработка заканчивалась бы одновременно. В это время работники могут находиться в отпуске, проходить переобучение или работать над другими проектами. Отсюда и следует важность четкого соблюдения графика (например, специалиста ждут в другом проекте, а этап, в котором он занят сейчас, реализуется с задержкой).

#### **Литература:**

4. Планирование проекта с. 287 – 290

5. 4.2. Календарный график и штатная численность проекта с. 116 -118

12. Лабораторная работа 2. Разработка календарного плана проекта с. 12 - 19



Рис. 1.6

**Контрольные вопросы для самопроверки:**

1. В чём достоинство сети "узел-работа"?
2. Как определить вершины, лежащие на критическом пути, если плановый срок окончания проекта  $T > EF_{\text{окончания}}$ ?
3. Какого типа матрицы можно использовать для описания сети "узел-работа"?
4. Какой граф называется сетью?
5. Почему множество работ по созданию ИС является частично-упорядоченным?
6. Какие разработчики назначаются в бригаду «критического пути»?

**Задания к лабораторной работе**

Преобразовать сеть «дуга-работа» в сеть «узел-работа» и произвести два вида оптимизации календарного графика.

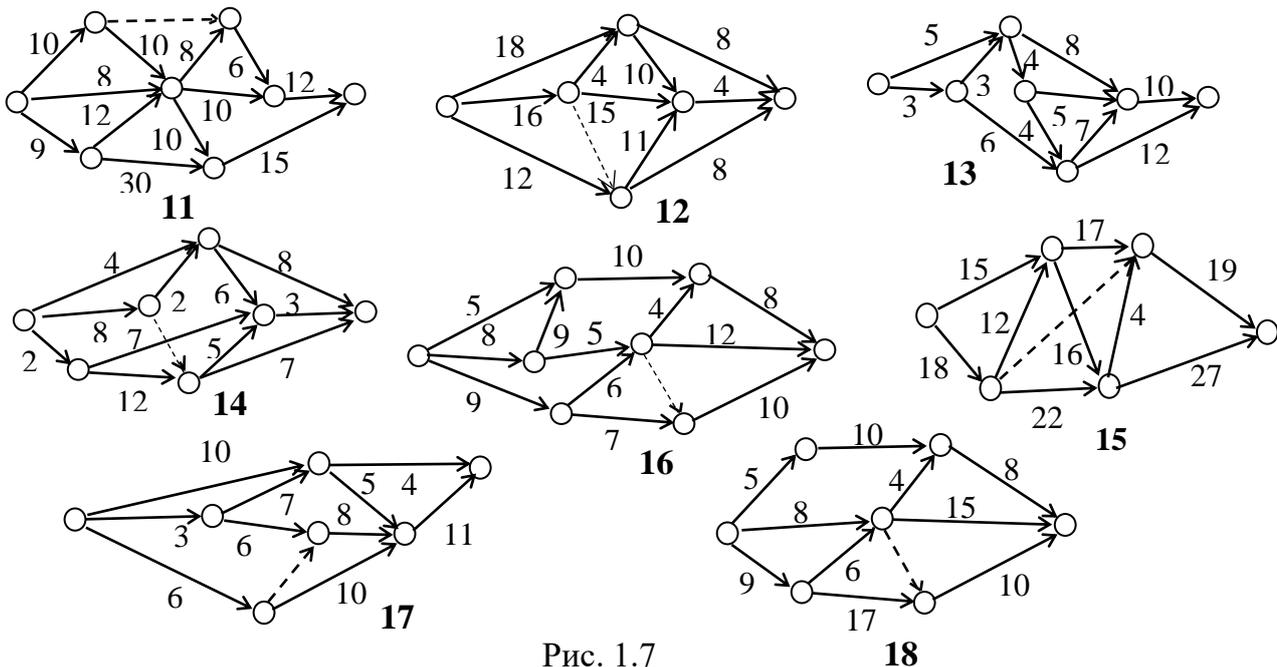


Рис. 1.7

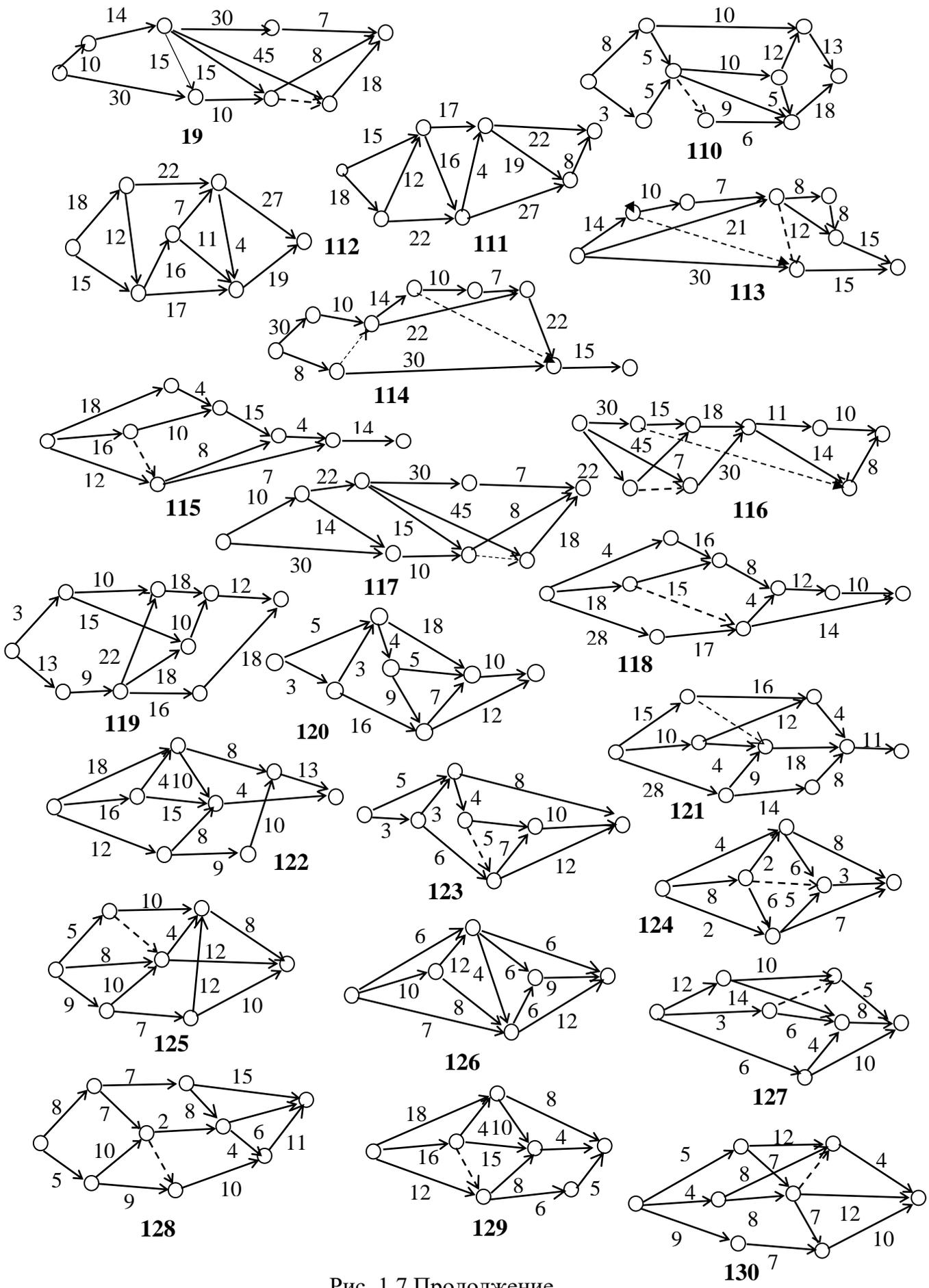


Рис. 1.7 Продолжение

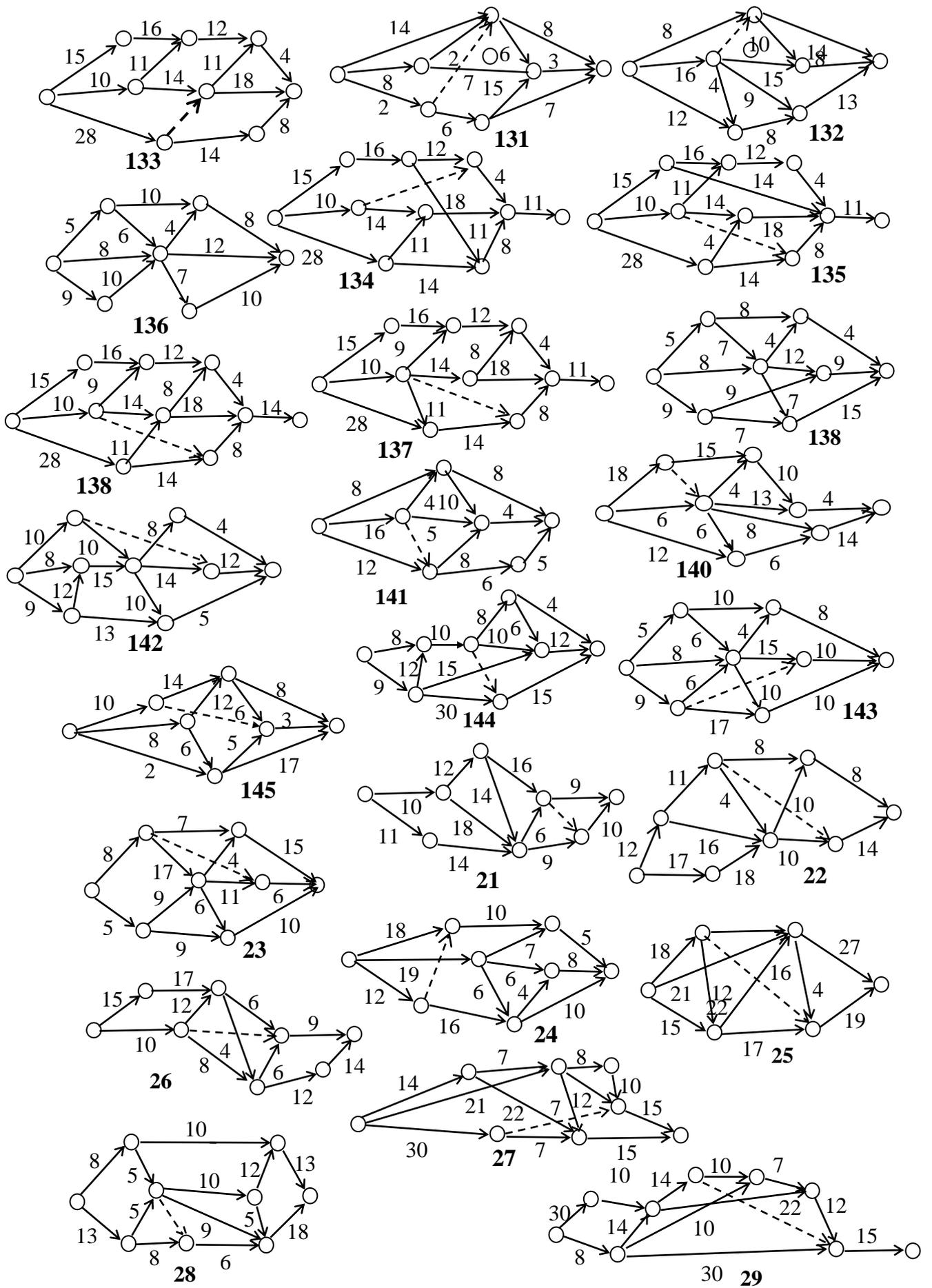


Рис. 1.7 Продолжение

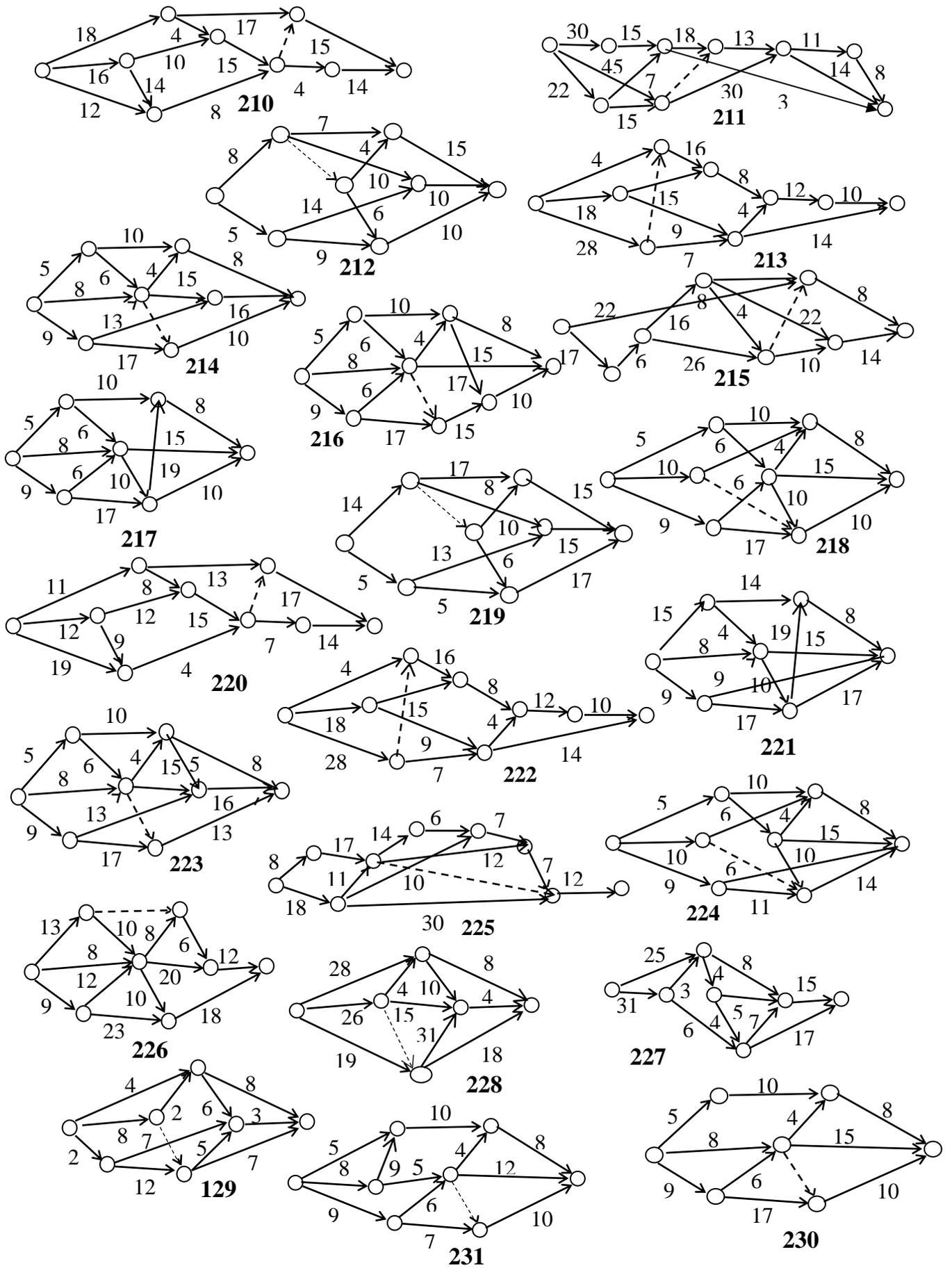


Рис. 1.7 Продолжение

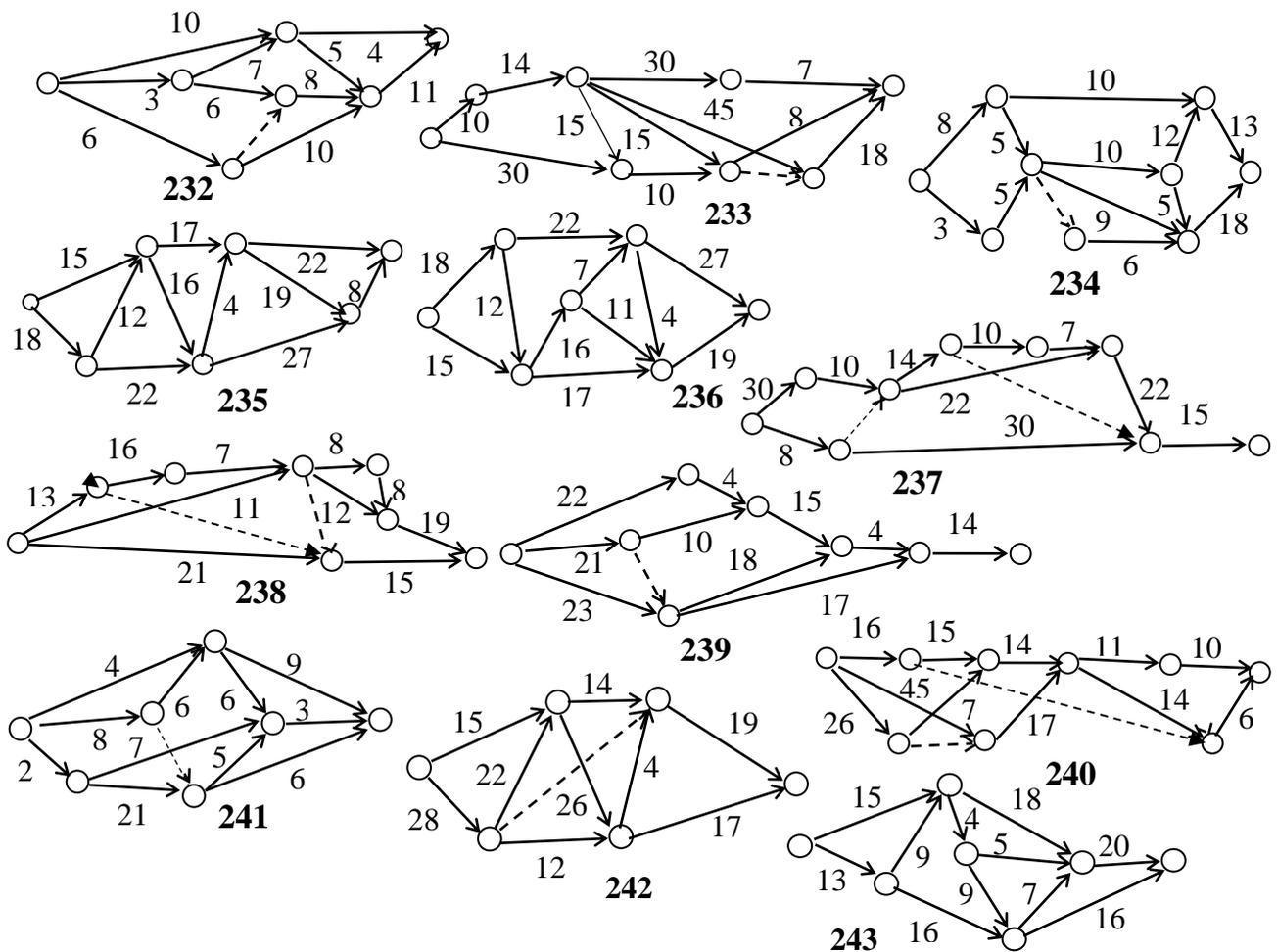


Рис. 1.7 Продолжение

***Методические указания и порядок выполнения работы***

1. В соответствии с вариантом выбрать на рис. 1.7 сетевой график "дуга-работа". Первая цифра в номере варианта – это номер группы, далее следует номер студента в списке группы.
2. Поименовать работы.
3. Преобразовать график в сеть "узел-работа".
4. Рассчитать все временные параметры работ.
5. Выделить в сети критический путь.
6. Построить в масштабе календарные планы выполнения работ в двух заданных вариантах оптимизации.

***Требования к отчету и защите***

Отчет по выполненной лабораторной работе должен включать:

1. Исходную сеть работ.

2. Преобразованную сеть работ.
3. Промежуточные и конечные результаты расчетов.
4. Вычерченные в масштабе календарные планы обеих оптимизаций.

Весь отчет выполняется шрифтом с размером не менее 12 пунктов.

Защита проводится во время лабораторных занятий и представляет собой перечень ответов на вопросы по теоретическим аспектам изучаемой темы и практическим результатам, полученным в ходе выполнения лабораторной работы. Примеры подобных вопросов приведены ранее, в разделе «Контрольные вопросы для самопроверки». Результатом каждого ответа является бинарная оценка «правильно/неправильно». Защита считается состоявшейся, если студент набрал первыми три правильных ответа. В противном случае защита не состоялась и может быть повторена не ранее, чем через день.

## Лабораторная работа № 2 **Использование метода экспертных оценок при разработке программных систем**

### ***Общие сведения***

*Цель работы:* изучение основных положений метода экспертных оценок, отработка методики проведения опроса и обработки его результатов применительно к планированию процесса разработки программной системы.

Работа носит чисто расчетный характер, из CASE-средств используется только любой текстовый редактор для оформления отчета по лабораторной работе.

Для успешной сдачи лабораторной работы необходимо предоставить отчет и пройти защиту в соответствии с разделом «Требования к отчету и защите».

Аудиторное время выполнения (под руководством преподавателя): 3 часа.

Время самостоятельной подготовки: 3 часа.

### ***Теоретическое введение***

В процессе разработки программных систем многие предварительные оценки (вероятность тех или иных рисков, стоимость проекта, очередность реализации задач и др.) не могут быть получены на основе строгих формализованных процедур в силу их отсутствия или высокой трудоемкости использования. В этом случае широко используется знания и опыт экспертов - специалистов в данной области.

В зависимости от того, принимается во внимание мнение одного эксперта или группы, различают индивидуальные и коллективные методы экспертных оценок. Коллективные методы предпочтительнее из-за бóльшей точности и конкретности результатов эксперимента. Они делятся на три группы:

- *метод комиссий* - проведение группой экспертов дискуссии с целью выработки общего мнения о характере исследуемого объекта;
- *метод отнесенной оценки* - многоэтапный процесс, заключающийся в чередовании свободных творческих дискуссий с индивидуальными критическими оценками каждым экспертом результатов работы комиссии;
- *дельфийский метод* - заочный опрос по специальным анкетам и обработка ответов методами математической статистики.

Наибольшее распространение на практике получил именно дельфийский метод.

Методика организации опроса и анализа его результатов по данному методу предполагает следующие этапы:

1. Определить количество и состав групп экспертов;
2. Сформулировать основной вопрос таким образом, чтобы эксперт интерпретировал его однозначно и мог дать ответ в численном виде;
3. Составить анкету, включающую основной и дополнительный вопросы, ответы на которые разъясняют существо основного вопроса;
4. Провести первый тур опроса;
5. Проанализировать ответы на согласованность мнений, выявить дополнительные факторы, которые необходимо учесть экспертам, определить экспертов, имеющих максимальное расхождение с общим мнением группы;
6. Познакомить экспертов с ответами других членов группы, сформулировать дополнительные вопросы;
7. Провести второй, третий и т.д. туры опросов до получения согласованного мнения экспертов;
8. Обобщить результаты опроса и выявить рекомендации по исследуемой проблеме.

В данной лабораторной работе рассматривается только процесс математической обработки результатов опроса (этап 5).

Предположим, что опрос посвящен определению очередности разработки задач создания программного проекта и результаты опроса сведены в таблицу 2.1.

Табл. 2.1

Номера задач	Задачи	Эксперты				
		1	2	3	4	5
1	Разработка структуры БД	2	3	1	2	3
2	Определение архитектуры системы	2	2	3	1	2
3	Разработка видеформ	3	3	4	2	3
4	Разработка плана аттестации	5	4	4	4	5
5	Определение рисков	1	1	1	3	1
6	Определение архитектуры технических средств	4	4	4	5	4
$Q_i$		1	2	2	1	1
$t_{1i}$		2	2	2	2	2
$t_{2i}$			2	3		
$T_i$		6	12	30	6	6

При выставлении оценок эксперты:

- могут нескольким задачам поставить одинаковые оценки (ранги);

- максимальное значение ранга соответствует числу учитываемых задач (в данном случае 6);
- минимальное значение ранга (1) соответствует наиболее важной (первоочередной) задаче.

На следующем шаге определяется количество групп одинаковых рангов  $Q_i$ , назначенных каждым экспертом, и количество рангов в каждой группе  $t_{ji}$  (см. табл. 2.1), а также для каждого эксперта значение показателя

$$T_i = \sum_{q_i=1}^{Q_i} (t_{q_i}^3 - t_{q_i}). \quad (2.1)$$

Если проанализировать суммы рангов, выставленных каждым экспертом, то можно увидеть, что, в общем случае, они не равны между собой (в табл. 2.1 суммы у четырех первых экспертов 17, а у пятого 18). Это значит, мнения разных экспертов будут иметь разный вес в определении итогового результата. С целью устранения данной некорректности проводится нормирование рангов у каждого эксперта (см. табл. 2.2)

Табл. 2.2

Номера задач	Эксперты					$S_j$	$\beta_j$	Очередность решения задач
	1	2	3	4	5			
1	2,5	3,5	1,5	2,5	3,5	13,5	4	3
2	2,5	2	3	1	2	10,5	7	2
3	4	3,5	5	2,5	3,5	18,5	-1	4
4	6	5,5	5	5	6	27,5	-10	6
5	1	1	1,5	4	1	8,5	9	1
6	5	5,5	5	6	5	26,5	-9	5

Алгоритм нормирования заключается в следующем:

- записать последовательность целых чисел: 1, 2, 3, ..., n, где n - количество рассматриваемых факторов (в данном случае задач);
- определить факторы, которым выставлены наименьшие ранги;
- на последовательности чисел выбрать столько наименьших чисел, сколько факторов имеет наименьший ранг. Сумму чисел разделить на количество факторов. Полученные значения записать в матрицу нормированных рангов вместо выбранных.
- пункты а., б., с. повторять до тех пор, пока не будут пронормированы все факторы.

Для примера пронормируем ранги, выставленные 1-м экспертом.

Задаче 5 присвоен наименьший ранг, равный 1. Из последовательности 1, 2, 3, 4, 5, 6 выбираем первое число (т.е. 1). В нормированной матрице задаче 5 переписывается ранг 1.

Задачам 1 и 2 в исходной матрице присвоен ранг 2. Выбираем первые два числа из оставшейся последовательности 2, 3, 4, 5, 6 (т.е. 2 и 3). Тогда ранг задач 1 и 2 определится выражением  $\frac{2+3}{2} = 2,5$ .

Задаче 3 в исходной матрице присвоен ранг 3. Из оставшейся последовательности 4, 5, 6 выбираем первое число (т.е. 4) и заносим его в нормированную матрицу в качестве ранга задачи 3.

Задаче 6 в исходной матрице присвоен ранг 4. Из последовательности 5, 6 выбираем первое число (т.е. 5) и заносим его в нормированную матрицу в качестве ранга задачи 6.

В последовательности остается только число 6, которое станет нормированным рангом задачи 4

Аналогично нормируются ранги для остальных экспертов. Очевидно, что суммы столбцов в нормированной матрице должны быть равны между собой (т. е. мнения всех экспертов имеют одинаковый вес) и равняться значению

$$\sum_{j=1}^n d'_{ji} = \frac{n+1}{2} * n,$$

где  $d'_{ji}$  - нормированный ранг  $j$ -го фактора, назначенный  $i$ -м экспертом;

$n$  - число факторов.

В данном случае сумма будет равняться  $1 + 2 + 3 + 4 + 5 + 6 = 21$ .

Далее для нормированной матрицы определяется сумма рангов  $S_j$ , назначенных  $j$ -му фактору, по формуле  $S_j = \sum_{i=1}^m d'_{ji}$ , где  $m$  - число факторов. Очевидно, что в рассматриваемом примере  $S_1 = 2,5 + 3,5 + 1,5 + 2,5 + 3,5 = 13,5$ . Для остальных факторов значения  $S_j$  рассчитываются аналогично.

При этом предполагается, что чем меньше величина  $S_j$ , тем больше важность  $j$ -го фактора, т. е. в данном случае данный параметр определяет очередность выполнения задач (эксперты рекомендуют начать с определения рисков (задача 5), затем определить архитектуру системы (задача 2) и т. д.).

На следующем шаге определяется средний ранг совокупности факторов по формуле

$$S = \frac{\sum_{j=1}^n S_j}{n}.$$

Для рассматриваемого примера он составляет

$$S = \frac{13,5 + 10,5 + 18,5 + 27,5 + 8,5}{6} = 17,5.$$

Средний ранг позволяет найти отклонения  $\beta_j$  суммарных рангов  $S_j$  от среднего значения  $S$ , т. е.  $\beta_j = S - S_j$ . Результаты расчетов смотри в табл. 2.2.

Последним шагом обработки является нахождение коэффициента конкордации (согласованности мнения экспертов) по формуле

$$K = \frac{12 \sum_{j=1}^n \beta_j^2}{m^2 (n^3 - n) - m \sum_{i=1}^m T_i}. \quad (2.2)$$

Значение  $T_i$  вычисляется по выражению (2.1).

Максимальным значением коэффициента конкордации является 1 (все мнения экспертов максимально согласованны), а минимальным 0. Пороговым значением является 0,6: коэффициент конкордации выше этого значения предполагает, что эксперты достаточно единодушны в своих оценках и можно их мнение использовать как руководство к действию. При коэффициенте конкордации меньше 0,6 мнения экспертов существенно расходятся и требуется повторный тур опроса.

Для рассматриваемого примера коэффициент конкордации определится как

$$K = \frac{12(16 + 49 + 1 + 100 + 81 + 81)}{5^2(6^3 - 6) - 5(6 + 12 + 30 + 6 + 6)} = 0,8.$$

Поскольку  $K > 0,6$ , то дополнительный тур опроса можно не проводить.

До сих пор предполагалось, что все эксперты имеют одинаковую профессиональную квалификацию, что обуславливает одинаковую значимость их мнений. В действительности такая ситуация имеет место редко, т.е. к опросу привлекаются эксперты с разным уровнем опыта и профессиональных знаний. Для учета квалификации, компетенции, опыта работы каждому эксперту может присваиваться весовой ко-

эффицент. При этом сами эксперты, как правило, не знают присвоенные им коэффициенты.

Для учета веса эксперта при расчете суммарных рангов факторов  $S_j$  коэффициента конкордации  $K$  веса предварительно нормируются на основе выражения

$$\gamma'_i = \frac{\gamma_i}{\sum_{i=1}^m \gamma_i}, \quad (2.3)$$

где  $\gamma_i$  - первоначальный весовой коэффициент  $i$ -го эксперта;

$\gamma'_i$  - нормированный весовой коэффициент  $i$ -го эксперта.

Тогда суммарные ранги факторов определяются по формуле

$$S_j = \sum_{i=1}^m \gamma'_i d'_{ji},$$

а коэффициент конкордации - по формуле

$$K = \frac{12 \sum_{j=1}^n \beta_j^2}{(n^3 - n) - \frac{\sum_{i=1}^m T_i}{m}}.$$

Пороговое значение коэффициента конкордации остается прежним.

### ***Литература:***

11. 2.5.2 Прямой метод определения размеров программного продукта на основе опыта экспертов, с. 62 – 65.
13. 4.3.2. Модель функциональных зависимостей оценки перспективности концепции проекта, с. 95 -98.
17. 5.5.1 Оценка перспективности концепции на основе мнения экспертов, с. 89 – 90.

### ***Контрольные вопросы для самопроверки***

1. В каких случаях используется экспертный опрос?
2. Кто может быть экспертом?
3. Каким может быть максимальное значение коэффициента конкордации?
4. Для чего вводятся весовые коэффициенты при экспертном опросе?
5. Чему равна сумма весов нормированных коэффициентов?

### *Задания к лабораторной работе*

Произвести обработку экспертного опроса на основе данных, представленных ниже в таблицах.

Таблица 2.3 Строки таблицы 2.4 с номерами задач для каждого варианта

<b>Вариант</b>	<b>Задачи</b>
1-1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
1-2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13
1-3	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14
1-4	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15
1-5	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 16
1-6	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 17
1-7	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13
1-8	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14
1-9	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 15
1-10	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 16
1-11	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 17
1-12	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14
1-13	2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15
1-14	1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14
1-15	3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
1-16	1, 2, 5, 6, 7, 8, 9, 10, 11, 15, 16, 17
1-17	4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 17
1-18	1, 3, 5, 7, 8, 9, 10, 12, 14, 15, 16, 17
1-19	2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16
1-20	1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17
1-21	1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 15, 16
1-22	1, 2, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17
1-23	1, 3, 4, 5, 6, 9, 10, 11, 12, 14, 16, 17
1-24	2, 3, 4, 7, 8, 9, 10, 11, 12, 15, 16, 17
1-25	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
1-26	4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
1-27	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
1-28	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17
1-29	2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16
1-30	2, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 17
1-31	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
1-32	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17
1-33	2, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 17
1-34	4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
1-35	2, 3, 4, 5, 7, 8, 10, 11, 12, 14, 15, 16
1-31	3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16, 17
1-36	4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

Таблица 2.3. Продолжение

<b>Вариант</b>	<b>Задачи</b>
1-37	2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16
1-38	3, 5, 6, 7, 8, 10, 11, 12, 14, 15, 16, 17
1-39	4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
1-40	2, 3, 4, 5, 7, 8, 10, 11, 12, 13, 15, 16
1-41	3, 4, 6, 7, 8, 10, 11, 13, 14, 15, 16, 17
1-42	4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
1-43	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
1-44	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
1-45	2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16
2-1	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
2-2	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
2-3	2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14
2-4	2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14
2-5	2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14
2-6	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13
2-7	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 14
2-8	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 16
2-9	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 17
2-10	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13
2-11	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 14
2-12	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 15
2-13	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 16
2-14	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 17
2-15	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13
2-16	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14
2-17	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15
2-18	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 16
2-19	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 17
2-20	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13
2-21	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14
2-22	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 15
2-23	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 16
2-24	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 17
2-25	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13
2-26	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 14
2-27	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 15
2-28	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 16
2-29	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 17
2-30	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13
2-31	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 14
2-32	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 13, 14
2-33	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 16
2-34	1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 15, 17
2-35	1, 2, 3, 6, 7, 9, 10, 11, 12, 13, 14, 16

Таблица 2.3. Продолжение

<b>Вариант</b>	<b>Задачи</b>
2-36	2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 14, 15
2-37	1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 14, 15
2-38	3, 4, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16
2-39	1, 2, 3, 4, 7, 8, 9, 11, 12, 13, 15, 17
2-40	1, 3, 5, 6, 8, 9, 10, 11, 12, 13, 15, 17
2-41	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 15
2-42	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17

Таблица 2.4 Значения рангов для каждой задачи, выставленные экспертами

<b>№ задач</b>	<b>Эксперты</b>								
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
1	1	1	1	2	1	3	1	2	1
2	2	1	1	2	1	2	2	1	2
3	8	5	6	5	6	6	5	6	6
4	4	4	2	2	2	9	4	4	5
5	4	4	5	5	5	9	5	6	4
6	5	9	6	5	5	5	5	6	5
7	5	7	5	5	4	6	5	4	5
8	7	2	3	2	3	3	8	4	11
9	3	3	4	4	4	2	3	4	2
10	3	3	10	3	4	4	3	5	3
11	1	1	1	2	11	2	2	2	1
12	4	2	5	5	1	5	5	3	2
13	2	1	2	2	2	2	1	10	1
14	2	9	5	5	5	2	5	6	2
15	2	2	9	2	2	3	3	2	2
16	2	4	5	5	5	2	2	5	3
17	2	1	1	2	1	2	2	2	2

Таблица 2.5 Весовые коэффициенты экспертов

<b>Вариант</b>	<b>Коэффициенты</b>
1-1	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-2	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-3	0.5, 0.1, 0.2, 0.2, 0.3, 0.1, 0.1, 0.4, 0.2
1-4	0.2, 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2
1-5	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.4
1-6	0.1, 0.5, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-7	0.1, 0.3, 0.2, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
1-8	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
1-9	0.6, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.4, 0.2
1-10	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-11	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2

Таблица 2.5. Продолжение

<b>Вариант</b>	<b>Коэффициенты</b>
1-12	0.5, 0.1, 0.2, 0.2, 0.3, 0.1, 0.1, 0.4, 0.2
1-13	0.2, 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2
1-14	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.4
1-15	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-16	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-17	0.5, 0.1, 0.2, 0.2, 0.3, 0.1, 0.1, 0.4, 0.2
1-18	0.2, 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2
1-19	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.4
1-20	0.1, 0.5, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-21	0.1, 0.3, 0.2, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
1-22	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
1-23	0.6, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.4, 0.2
1-24	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-25	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
1-26	0.5, 0.1, 0.2, 0.2, 0.3, 0.1, 0.1, 0.4, 0.2
1-27	0.2, 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2
1-28	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.4
1-29	0.1, 0.3, 0.2, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
1-30	0.6, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.4, 0.2
1-31	0.1, 0.5, 0.3, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
1-32	0.4, 0.3, 0.2, 0.2, 0.2, 0.3, 0.5, 0.4, 0.2
1-33	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.4, 0.1, 0.3
1-34	0.6, 0.4, 0.1, 0.1, 0.2, 0.1, 0.5, 0.4, 0.2
1-35	0.3, 0.5, 0.2, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
1-36	0.1, 0.4, 0.1, 0.2, 0.2, 0.1, 0.4, 0.4, 0.3
1-37	0.1, 0.4, 0.2, 0.2, 0.3, 0.1, 0.3, 0.4, 0.2
1-38	0.2, 0.1, 0.2, 0.3, 0.2, 0.1, 0.1, 0.2, 0.2
1-39	0.5, 0.1, 0.2, 0.2, 0.2, 0.2, 0.1, 0.4, 0.4
1-40	0.2, 0.3, 0.2, 0.2, 0.3, 0.3, 0.1, 0.4, 0.2
1-41	0.6, 0.3, 0.1, 0.3, 0.2, 0.1, 0.1, 0.4, 0.2
1-42	0.2, 0.3, 0.1, 0.2, 0.2, 0.4, 0.1, 0.1, 0.2
1-43	0.1, 0.3, 0.2, 0.1, 0.2, 0.3, 0.1, 0.2, 0.4
1-44	0.4, 0.3, 0.2, 0.2, 0.3, 0.3, 0.1, 0.4, 0.2
1-45	0.3, 0.3, 0.2, 0.2, 0.3, 0.4, 0.1, 0.4, 0.1
2-1	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-2	0.2, 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2
2-3	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.4
2-4	0.1, 0.5, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-5	0.1, 0.3, 0.2, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
2-6	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
2-7	0.6, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.4, 0.2
2-8	0.6, 0.1, 0.3, 0.1, 0.2, 0.1, 0.3, 0.3, 0.3
2-9	0.1, 0.5, 0.2, 0.3, 0.2, 0.1, 0.1, 0.4, 0.2
2-10	0.1, 0.3, 0.2, 0.1, 0.2, 0.1, 0.1, 0.2, 0.4

Таблица 2.5. Продолжение

<b>Вариант</b>	<b>Коэффициенты</b>
2-11	0.4, 0.4, 0.2, 0.2, 0.3, 0.3, 0.1, 0.4, 0.2
2-12	0.3, 0.3, 0.2, 0.2, 0.3, 0.4, 0.1, 0.4, 0.1
2-13	0.2, 0.2, 0.3, 0.4, 0.3, 0.1, 0.1, 0.3, 0.2
2-14	0.5, 0.4, 0.3, 0.2, 0.1, 0.1, 0.2, 0.3, 0.4
2-15	0.1, 0.5, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-16	0.1, 0.3, 0.2, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
2-17	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
2-18	0.6, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.4, 0.2
2-19	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-20	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-21	0.5, 0.1, 0.2, 0.2, 0.3, 0.1, 0.1, 0.4, 0.2
2-22	0.2, 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2
2-23	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.4
2-24	0.1, 0.5, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-25	0.1, 0.3, 0.2, 0.2, 0.2, 0.3, 0.1, 0.4, 0.2
2-26	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
2-26	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
2-27	0.6, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.4, 0.2
2-28	0.6, 0.1, 0.3, 0.1, 0.2, 0.1, 0.3, 0.3, 0.3
2-29	0.1, 0.5, 0.2, 0.3, 0.2, 0.1, 0.1, 0.4, 0.2
2-30	0.1, 0.3, 0.2, 0.1, 0.2, 0.1, 0.1, 0.2, 0.4
2-31	0.1, 0.4, 0.2, 0.2, 0.4, 0.1, 0.3, 0.4, 0.2
2-32	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-33	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-34	0.2, 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.2, 0.2
2-35	0.1, 0.5, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-36	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
2-37	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-38	0.5, 0.1, 0.2, 0.2, 0.3, 0.1, 0.1, 0.4, 0.2
2-39	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.4
2-40	0.2, 0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2
2-41	0.3, 0.3, 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0.2
2-42	0.1, 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.4, 0.2

***Методические указания и порядок выполнения работы***

1. В соответствии с вариантом из табл. 2.3 выбрать номера строк табл. 2.4, представляющие номера задач, очередность выполнения которых определяется в результате экспертного контроля.
2. Из табл. 2.4 выбрать значения рангов для данных задач и сформировать таблицу - результат опроса, аналогичную табл. 2.1.

3. Провести нормализацию матрицы рангов в соответствии с методикой, изложенной в разделе «Теоретическое введение», и получить таблицу, аналогичную табл. 2.2.
4. Определить по табл. 2.2 очередность реализации задач.
5. На следующем шаге определяется количество групп одинаковых рангов  $Q_i$ , назначенных каждым экспертом, и количество рангов в каждой группе  $t_{ji}$  (см. табл. 2.1), а также на основании выражения (2.1) для каждого эксперта значение показателя  $T_i$ .
6. На основании выражения (2.2) вычислить коэффициент конкордации и принять окончательное решение о достаточности проведенного опроса или его продолжении.
7. Из табл. 2.5 выбрать в соответствии с вариантом массив весовых коэффициентов экспертов.
4. В соответствии с выражением (2.3) найти нормированные весовые коэффициенты экспертов для своего варианта.
5. Определить очередность реализации задач на основе учета весовых коэффициентов экспертов.
6. Определить коэффициент конкордации в данных условиях.
7. Сформулировать окончательный вывод о дальнейшем ходе опроса.

### ***Требования к отчету и защите***

Отчет по выполненной лабораторной работе должен включать:

1. Исходные данные для всех расчетов;
2. Этапы расчета со всеми сформированными таблицами и расчетными выражениями;
3. Выводы.
4. Весь отчет выполняется шрифтом с размером не менее 12 пунктов.

Защита проводится во время лабораторных занятий и представляет собой перечень ответов на вопросы по теоретическим аспектам изучаемой темы и практическим результатам, полученным в ходе выполнения лабораторной работы. Примеры подобных вопросов приведены ранее, в разделе «Контрольные вопросы для самопроверки». Результатом каждого ответа является бинарная оценка «правильно/неправильно». Защита считается состоявшейся, если студент набрал первыми три правильных ответа. В противном случае защита не состоялась и может быть повторена не ранее, чем через день.

## Лабораторная работа № 3 **Формирование информационной модели предметной области**

### ***Общие сведения***

*Цель работы:* изучение средств формирования информационной модели предметной области проектируемой программной системы с последующим закреплением на практике опыта использования данных средств.

Работа не требует поддержки специализируемым оборудованием, из CASE-средств используется только любой текстовый редактор для оформления отчета по лабораторной работе.

Для успешной сдачи лабораторной работы необходимо предоставить отчет и пройти защиту в соответствии с разделом «Требования к отчету и защите».

Аудиторное время выполнения (под руководством преподавателя): 4 часа.

Время самостоятельной подготовки: 3 часа.

### ***Теоретическое введение***

Цель стадии «Формирование и анализ требований» жизненного цикла (ЖЦ) проектируемой программной системы (ПС) – преобразование общих нечетких знаний о предназначении будущей ПС в сравнительно точные требования к ней, т.е. четкое формирование требующей решения проблемы. С этой целью проводится системный анализ внешнего окружения (метасистемы) будущей системы и на его основе строятся несколько моделей, отображающих различные аспекты решаемой проблемы на начальном этапе разработки: модель обработки данных, композиционная модель, архитектурная модель, классификационная модель, модель «стимул-ответ» и т. д. При этом они могут создаваться в двух версиях: модели “AS-IS”, отображающие текущую ситуацию, и “TO-BE”, представляющие конечную ситуацию как результат разработки ПС. В конечном итоге проектирование ПС будет сводиться к переходу от моделей “AS-IS” к моделям “TO-BE”.

Чаще всего выделяют на данной стадии *информационную, функциональную и поведенческую* модели. Информационная модель описывает информацию, которую, по мнению заказчика, должна обрабатывать будущая ПС. Она базируется на анализе свойств и природы объектов ПрО и информационных потребностей будущих пользователей разрабатываемой ПС и называется *концептуальной* или *инфологической*. Мо-

дель обобщенно представляет информационные потребности пользователей создаваемой системы в части использования данных и по существу является средством коммуникации разработчиков и будущих пользователей на разных стадиях ЖЦ ПС. Требования к инфологическим моделям:

- формализованность, позволяющая использовать автоматизированные средства обработки;
- наглядность за счет использования графических средств отображения.

Одним из наиболее широко используемых средств формализованного представления информационных моделей ПрО являются диаграммы «сущность-связь» (ERD - Entity-Relationship Diagrams, Чен, 1976 г.) в виде набора множества объектов и их характеристик, а также взаимосвязей между ними, нужных для выявления данных, которыми в дальнейшем будет оперировать проектируемая ПС. Существует несколько нотаций ERD: автора данных диаграмм Чена, Баркера, Мартина, IDEF1 и др. Но, независимо от нотации, любую ERD образуют три вида основных компонентов:

- **сущность** – реальный или абстрактный объект, имеющий определяющее значение для рассматриваемой ПрО (очевидно, что сущность аналогична типу данных «запись» или объекту в современных языках программирования). У каждой сущности обязательно должно быть имя и один или несколько атрибутов (рис. 3.1). Сущности делятся на *независимые* и *зависимые*. Существование экземпляров первого вида сущностей не зависит от наличия других сущностей, а вот экземпляры зависимой сущности должны быть удалены вместе с удалением экземпляра породившей их независимой сущности;

- **атрибут (свойство)** – некоторая характеристика экземпляра сущности, значимая в контексте рассматриваемой ПрО и предназначенная для квалификации, идентификации, классификации, количественной характеристики или отображения состояния сущности. Свойство может быть *простым*, т.е. не подлежащим дальнейшему делению с точки зрения решаемой проблемы, или *составным* (например, адрес) – состоящим из значений простых свойств. Свойство бывает базовым или производным

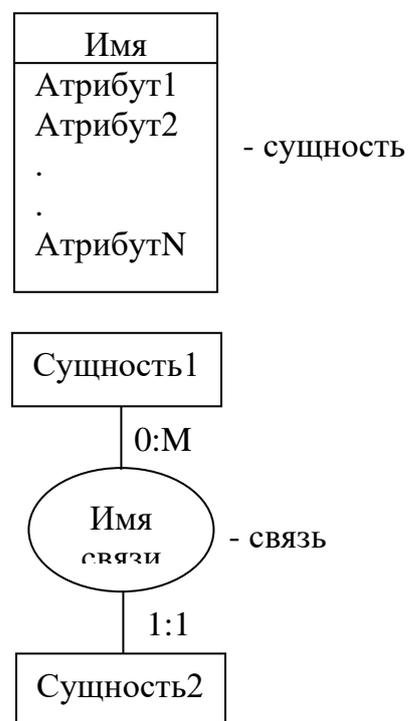


Рис. 3.1

(вычисляемым через базовые и/или другие производные). Различают также *статические* (или условно-постоянные, поскольку, если и меняются, то очень редко, например, адрес) и *динамические* (например, выработка за день) свойства. Если требуется уникальность значения свойства, чтобы по нему можно было однозначно идентифицировать сущность в определенном контексте, то свойство называется *ключевым*. Свойство может быть *необязательным*, если допускается в определенных ситуациях отсутствие его значения;

- **связь** – поименованная ассоциация между двумя или более сущностями, значимая для рассматриваемой ПрО. Количественный характер участия экземпляров сущностей в конкретной связи задается типом (мощностью) связи: «один к одному» (1:1), «один к многим» (1:M), «многие к одному» (M:1), «многие к многим» (M:N).

На рис. 3.1 представлены элементы нотации, используемой в CASE-средстве Silverrun, а на рис. 3.2 фрагмент ERD в данной нотации.

Каждая сущность обозначается прямоугольником, в верхней части которого находится ее имя, а под чертой список ее атрибутов. В начале списка идет идентификатор – один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности. На рис. 3.2 идентификаторы выделены жирным шрифтом. Для идентификации сотрудника достаточно одного атрибут **Табельный\_номер**.

В данной ПрО предполагается, что сотруднику разрешается совмещение, т.е. занятие нескольких штатных должностей (связь 1:M) с определенными окладами и величинам ставок, которые в этом варианте нотации указываются как свойство связи. Каждый отдел характеризуется своим набором штатных единиц, т.е. ликвидация отдела автоматически приводит к удалению соответствующих ему штатов, следовательно, сущность Штатная единица является *зависимой*, что отображается соответствующим обозначением мощности связи 1:1 с сущностью Отдел.

Конкретные штатные вакансии могут заниматься несколькими сотрудниками или быть в данный момент свободными (мощность 0:M).

Каждый сотрудник может входить в несколько рабочих групп как работник или не входить ни в одну (связь 0:M). В то же время Рабочая группа должна включать минимум одного работника (связь 1:M). Каждый работник может возглавлять несколько рабочих групп или ни одной (связь 0:M), но у каждой группы может быть один и только один руководитель (связь 1:1).

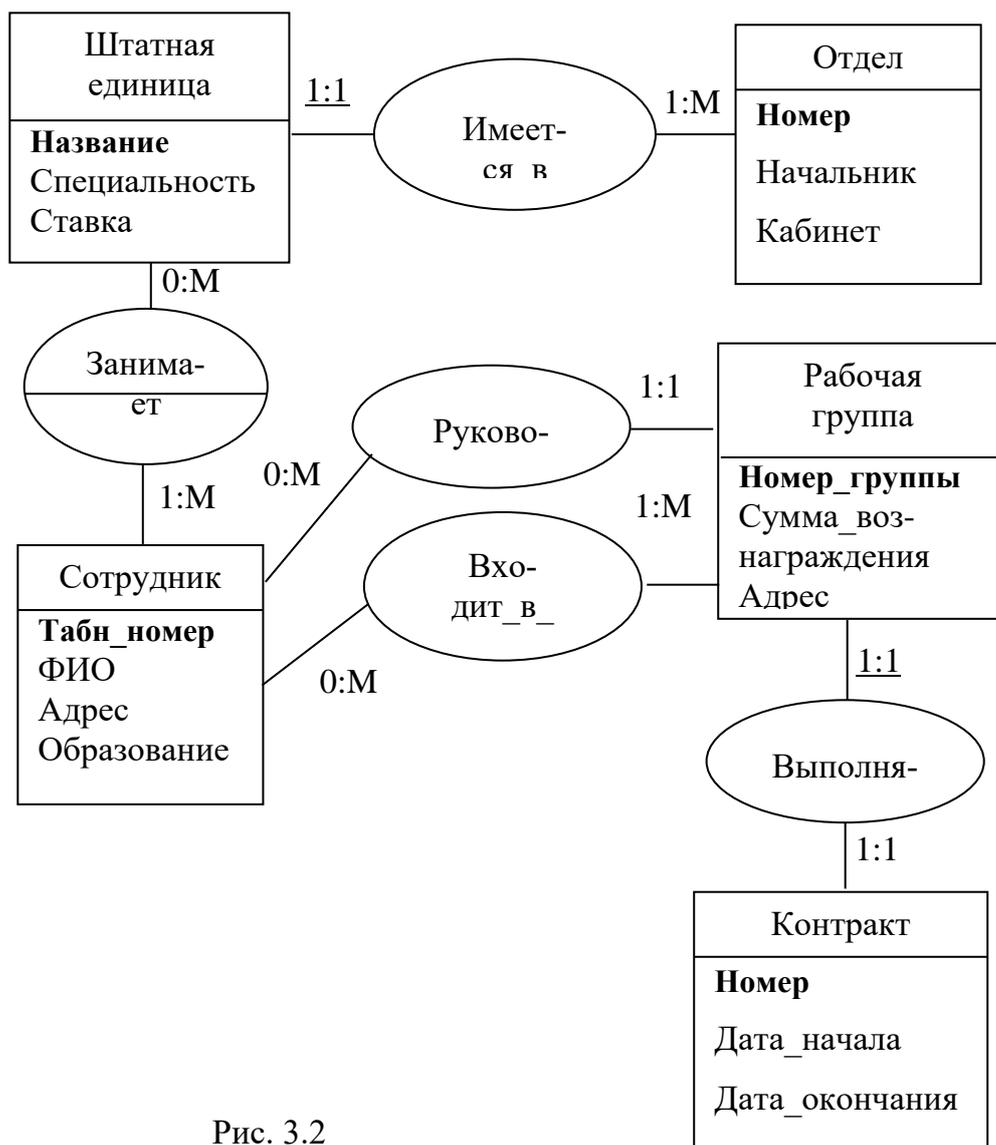


Рис. 3.2

Каждый контракт выполняется только одной рабочей группой (связь 1:1). Его окончание или прекращение автоматически ведет к прекращению соответствующей рабочей группы, т.е. сущность Рабочая группа также является зависимой.

Существует, как указывалось ранее, целый ряд других нотаций для формирования информационных моделей на стадии анализа ПрО:

Нотация Баркера (фирмы Oracle)

В этом случае сущность также отображается прямоугольником, обычно с прямыми, но используются иногда и прямоугольники со скругленными углами (рис. 3.3). Визуально зависимые и независимые изображаются одинаково. Внутри пиктограммы указываются имя сущности и соответствующий ей набор атрибутов: обязательный атрибут поме-

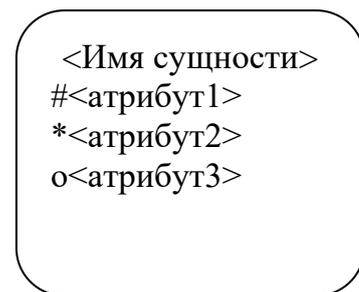


Рис. 3.3

чен звездочкой, необязательный – кружком. Решеткой идентифицируется атрибут, входящий в ключ.

Сущности, в данной нотации, делятся на *родительские* и *сущности-потомки*. Каждый экземпляр родительской сущности ассоциирован с произвольным (в том числе и нулевым) сущностей-потомков. Каждый же экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя. т. е. он может существовать только при наличии сущности-родителя.

Связи между сущностями (рис. 3.4) графически отображают степень и обязательность (кардинальность). На рис. 3.5 представлен пример связи между читателем библиотеки и книгами – один читатель может взять несколько книг или ни одной.

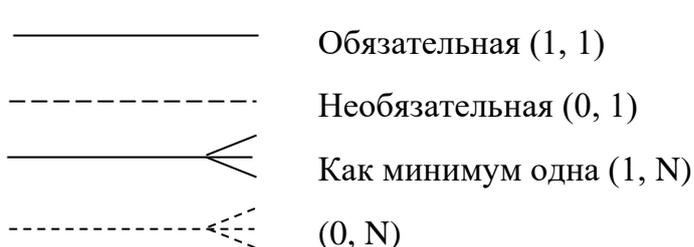


Рис. 3.4

Связь может именоваться. Имя в этом случае представляет глагол, помещаемый возле линии связи. В модели допускается дублирование имен связей.

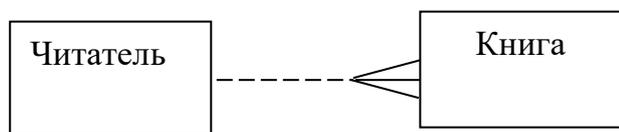


Рис. 3.5

Сущность может идентифицироваться полностью собственными ключевыми атрибутами (рис. 3.6,а) или с участием атрибутов сущности-родителя (рис. 3.6,б).

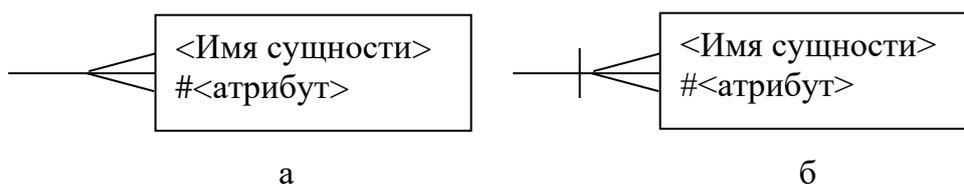


Рис. 3.6

В данной нотации также выделяют следующие конструкции:

- супертипы и подтипы – одна сущность является обобщающим понятием для группы более частных понятий (рис. 3.7). Иногда для указания категоризации используются дуги (скобки), которые охватывают подтипы;
- взаимно исключающие связи – каждый экземпляр сущности участвует только в одной связи из группы взаимно исключающих связей (рис. 3.8);
- рекурсивная связь – сущность может быть связана сама с собой (рис. 3.9).

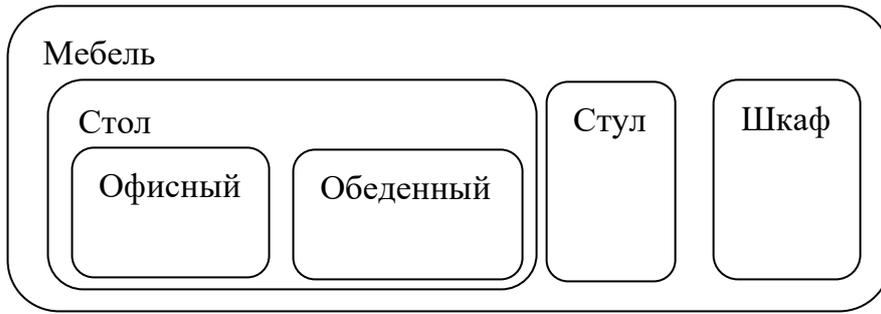


Рис. 3.7

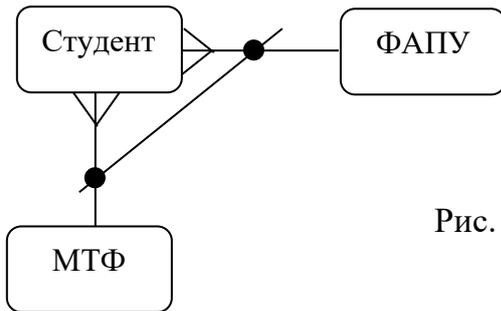


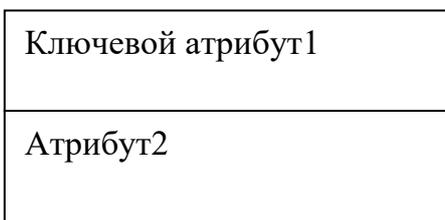
Рис. 3.8



Рис. 3.9

### Нотация метода IDEF1

Имя сущности/Но-  
мер сущности



Имя сущности/Но-  
мер сущности

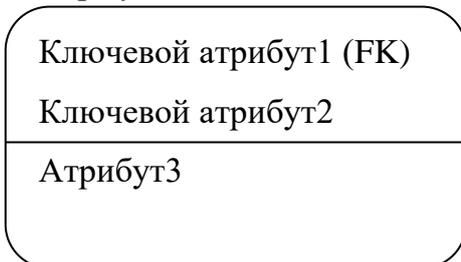


Рис. 3.10

В данной нотации независимая сущность обозначается прямоугольником с прямыми углами, а зависимая – со скругленными углами (рис. 3.10).

Каждой сущности присваивается уникальное имя и номер, разделяемые прямым слешем и размещаемые над пиктограммой. Атрибуты, формирующие первичный ключ, размещаются наверху списка и отделяются горизонтальной чертой. Если атрибут является внешним ключом, то после его имени добавляются буквы FK (Foreign Key).

Связь отображается линией без точки или с точкой на конце у сущности-потомка (табл. 3.1). Допускаются следующие значения мощности:

- каждый экземпляр сущности-родителя имеет только один экземпляр сущности-потомка (без точки);
- каждый экземпляр сущности-родителя может иметь ноль, один или более одного экземпляра связанного с ним экземпляра сущности-потомка (с точкой);

- каждый экземпляр сущности-родителя должен иметь не более одного экземпляра связанного с ним экземпляра сущности-потомка (Z);
- каждый экземпляр сущности-родителя должен иметь не менее одного экземпляра связанного с ним экземпляра сущности-потомка (P);
- каждый экземпляр сущности-родителя связан с некоторым фиксированным числом N экземпляров сущности-потомка.

Таблица 3.1

Обозначение	Кардинальность
_____	1,1
_____●	0,M
_____● Z	0,1
_____● P	1,M
_____● N	Точно N

Если экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем, то связь называется *идентифицирующей* (сплошная линия), в противном случае – *неидентифицирующей* (пунктирная линия).

Также в этой нотации вводится понятие «отношение категоризации», аналогичное понятию супертипа на рис. 3.8,

т.е. иллюстрирующее наличие иерархических связей в диаграмме. На рис. 3.11 представлен вариант полной категоризации работников вуза.

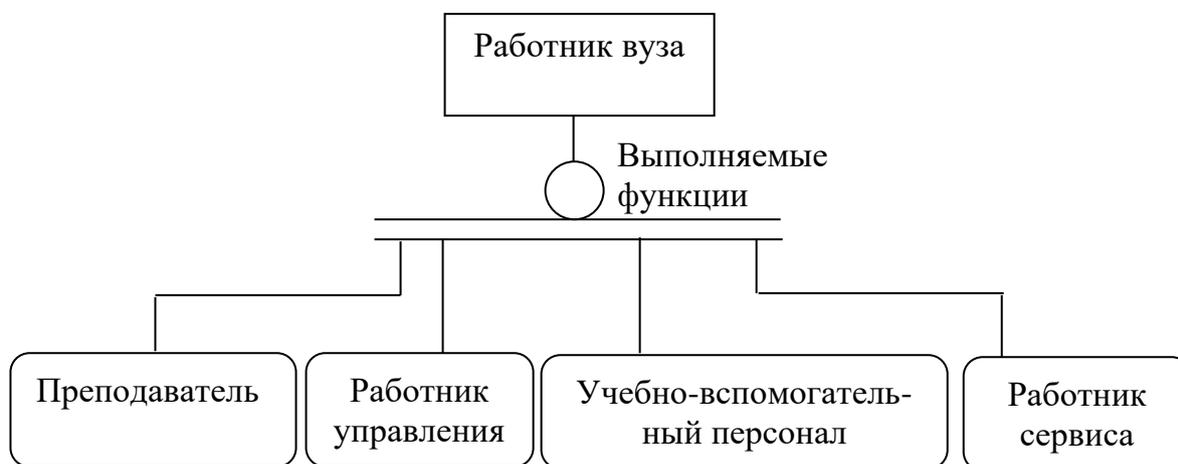


Рис. 3.11

При неполной категоризации ниже дескриптора изображается только одна горизонтальная линия.

### Нотация Чена

В этой нотации сущности также отображаются в виде прямоугольников (рис. 3.12), но зависимая сущность имеет двойной контур. Также выделяется родительская

сущность в иерархической связи, являющаяся аналогом супертипа на рис. 3.7 или 3.11. Отношения между сущностями отображаются в виде ромбов, в которые вписываются названия отношений (Рис. 3.13). Рядом с зависимой сущностью проставляется мощность отношения. Если отношение необязательно, то используется пунктирная линия. Атрибуты изображаются в виде овалов и связываются линиями с одним отношением или одной сущностью (рис. 3.14).

На рис. 3.15 представлен фрагмент описания некоторой ПрО в нотации Чена. Из диаграммы следует, что атрибутами сущности Сотрудник являются Табельный номер (первичный ключ), ФИО, Год рождения, Специальность (многозначный атрибут, поскольку у сотрудника может быть несколько специальностей), Иностранный язык (тоже многозначный атрибут, поскольку сотрудника может знать несколько языков, к тому же необязательный) и сложный атрибут Адрес, состоящий из простых атрибутов Город, Улица, Дом.

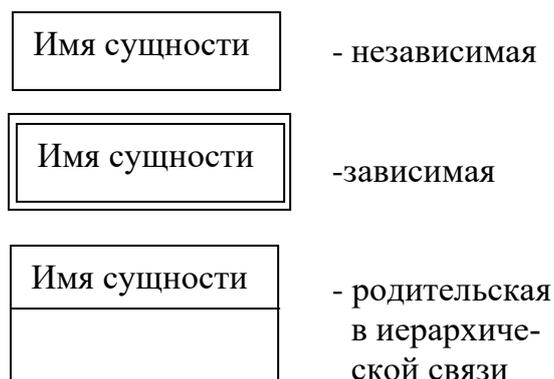


Рис. 3.12

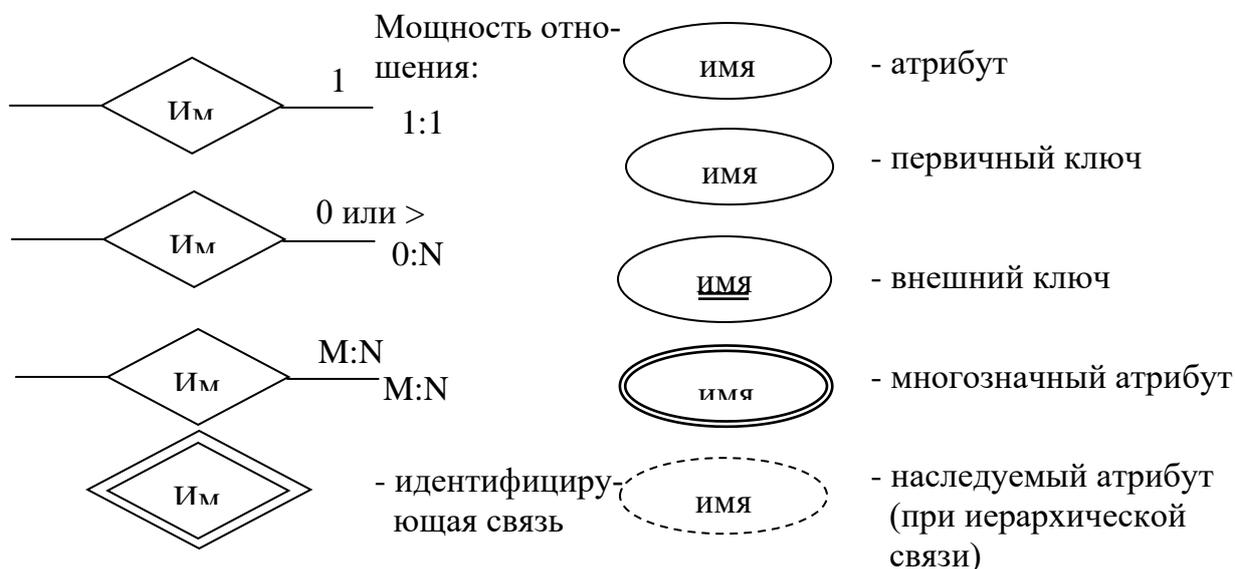


Рис. 3.13

Рис. 3.14

С каждым сотрудником обязательно соотносится определенный Отдел, в котором число работающих сотрудников не фиксировано.

Сущность Подчиненный является зависимой, так как удаление сотрудника, который является начальником, автоматически влечет удаление всех его подчиненных.

С другой стороны, эту связь можно сделать рекурсивной (см. рис. 3.9), так как каждый подчиненный и начальник являются сотрудниками

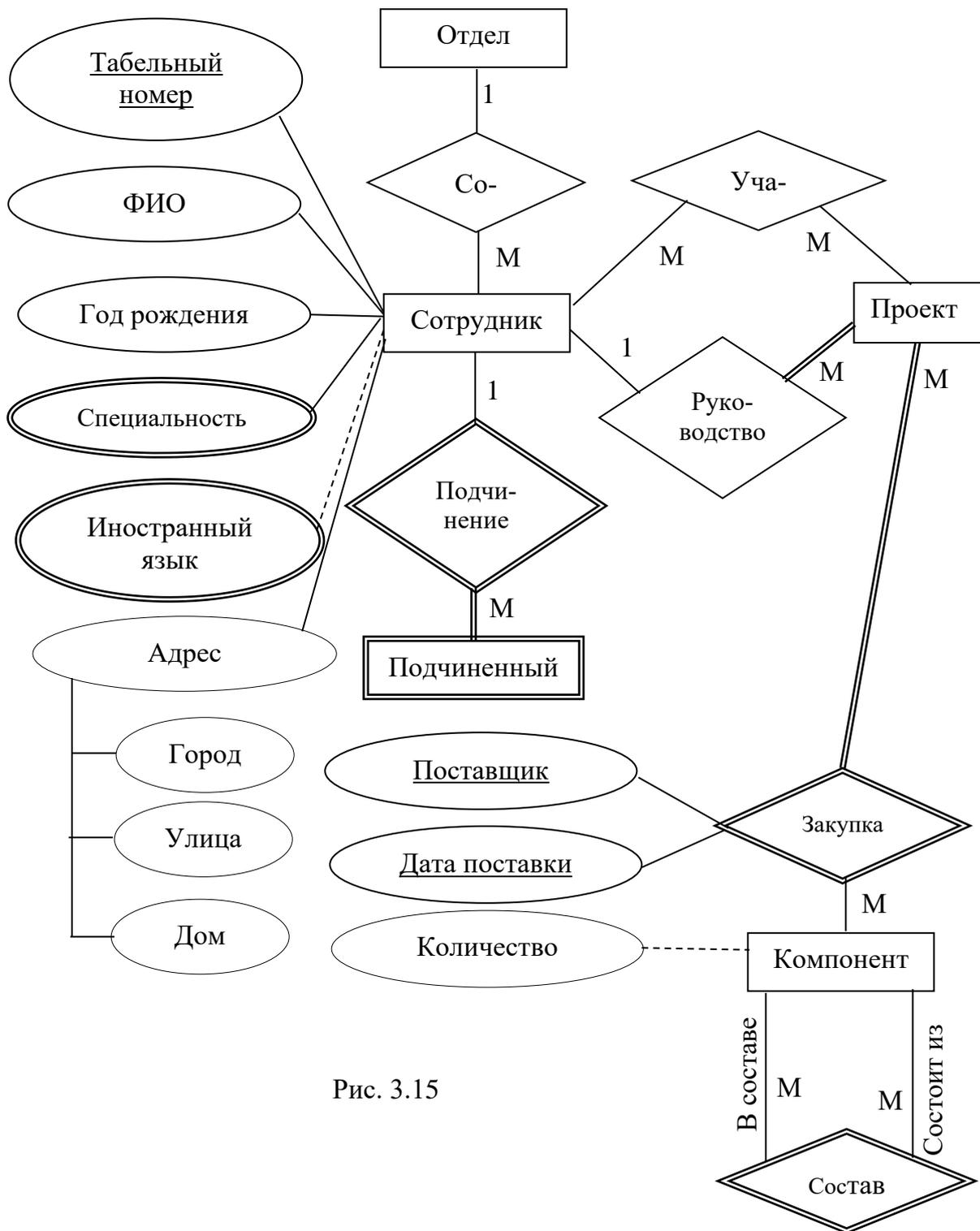


Рис. 3.15

Каждый сотрудник может участвовать в реализации неограниченного числа проектов как рядовой соисполнитель или руководитель, но у каждого Проекта может быть один и только один руководитель. При этом связь Руководство является идентифицирующей.

Для формирования проекта закупаются Компоненты, которые, в свою очередь, могут состоять из более мелких компонентов, т. е. имеет место рекурсивная связь. У связи Закупка также имеют место атрибуты Поставщик и Дата поставки, которые вместе образуют составной ключ.

### Нотация Мартина

В этом случае сущности также отображаются прямоугольниками трех видов (рис. 3.16). Ключевые атрибуты подчеркиваются. Виды связей и их кардинальность отображены в табл. 3.2. Имя связи указывается на линии, ее обозначающей.

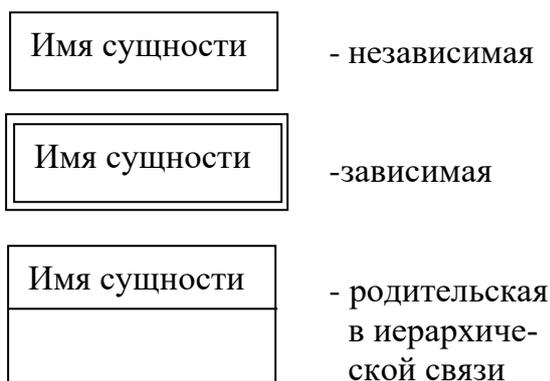


Рис. 3.16

Таблица 3.2

Обозначение	Кардинальность
_____	нет
_____	1,1
_____○	0,1
_____<	M,N
_____○<	0,N
_____+<	1,N

### ***Литература:***

22. Информационное моделирование и анализ требований : учебное пособие, 146 с.
1. 5. Определение требований к программному обеспечению, с. 71 -81.

### ***Контрольные вопросы для самопроверки***

1. На какой стадии жизненного цикла программной системы формируется информационная модель предметной области?
2. Какая связь называется идентифицирующей?
3. Может ли быть сущность без атрибутов?
4. Что означает связь M:M?
5. Может ли быть сущностью абстрактный объект?

### ***Задания к лабораторной работе***

В табл. 3.3 представлен перечень ПрО. В табл. 3.4 указаны для каждого студента в соответствии с его вариантом номер предметной области и вид нотации (1 – Sil-

verrun, 2 – Баркера, 3 - IDEF1, 4 - Чена, 5 - Мартина). Необходимо построить информационную модель в соответствии с заданием.

Таблица 3.3

№	Предметная область	№	Предметная область
1	Студенческая сессия	10	Информационная система заказов авиабилетов
2	Футбольный чемпионат	11	Драмтеатр
3	Приемная комиссия	12	Междугородний автовокзал
4	Сотрудники вуза	13	Автобаза, занимающаяся грузовыми перевозками
5	Картотека библиотеки	14	Учреждение общепита
6	Соревнования по фигурному катанию	15	Больница
7	Товарный магазин	16	Поликлиника
8	Оплаты жилищно-коммунальных услуг	17	Фильмотека
9	Туристической фирмы	18	Киностудия

Таблица 3.4

Вариант	Задание	Вариант	Задание	Вариант	Задание	Вариант	Задание
1-1	1-1	1-24	4-2	2-1	7-3	2-22	10-4
1-2	1-2	1-25	4-3	2-2	7-4	2-23	10-5
1-3	1-3	1-26	4-4	2-3	7-5	2-24	11-1
1-4	1-4	1-27	4-5	2-4	8-1	2-25	11-2
1-5	1-5	1-28	5-1	2-5	8-2	2-26	11-3
1-6	2-1	1-29	5-2	2-6	8-3	2-27	11-4
1-7	2-2	1-30	5-3	2-7	8-4	2-28	11-5
1-8	2-3	1-31	5-4	2-8	8-5	2-29	12-1
1-9	2-4	1-32	5-5	2-9	9-1	2-30	12-2
1-10	2-5	1-33	6-1	2-10	9-2	2-31	12-3
1-11	3-1	1-34	6-2	2-11	9-3	2-32	12-4
1-12	3-2	1-35	6-3	2-12	9-4	2-33	12-5
1-13	3-3	1-36	6-4	2-13	9-5	2-34	13-1
1-14	3-4	1-37	6-5	2-14	10-1	2-35	13-2
1-15	3-5	1-38	7-1	2-15	10-2	2-36	13-3
1-16	15-1	1-39	16-4	2-16	10-3	2-37	13-4
1-17	15-2	1-40	16-5	2-17	4-1	2-38	14-2
1-18	15-3	1-41	17-1	2-18	7-2	2-39	14-3
1-19	15-4	1-42	18-2	2-19	13-5	2-40	14-4
1-20	15-5	1-43	17-3	2-20	14-1	2-41	14-5
1-21	16-1	1-44	17-4	2-21	18-1	2-42	18-2
1-22	16-2	1-45	17-5				
1-23	16-3						

### ***Методические указания и порядок выполнения работы***

1. Из табл. 3.1 выбирается номера ПрО и вид модели ERD.
2. Формирует информационную модель на базе заданной нотации. Модель должна содержать не менее пяти сущностей и не менее двадцати атрибутов. Приветствуется объединение студентов с одной предметной областью в бригаду для совместной разработки информационной модели с описанием разными нотациями.
3. Каждый студент оформляет свой отчет в виде информационной модели и разрабатывает свои предложения по дополнению и совершенствованию используемой нотации.

### ***Требования к отчету и защите***

Отчет по выполненной лабораторной работе должен включать:

1. Вариант задания (название ПрО и вид нотации ERD);
2. Сформированную модель ПрО на основе заданной нотации;
3. Выводы.
4. Весь отчет выполняется шрифтом с размером не менее 12 пунктов.

Защита проводится во время лабораторных занятий и представляет собой перечень ответов на вопросы по теоретическим аспектам изучаемой темы и практическим результатам, полученным в ходе выполнения лабораторной работы. Примеры подобных вопросов приведены ранее, в разделе «Контрольные вопросы для самопроверки». Результатом каждого ответа является бинарная оценка «правильно/неправильно». Защита считается состоявшейся, если студент набрал первыми три правильных ответа. В противном случае защита не состоялась и может быть повторена не ранее, чем через день.

## Лабораторная работа № 4 **Изучение стандартов**

### **Общие сведения**

*Цель работы:* изучение российских и международных стандартов по разработке программного обеспечения.

Работа носит чисто расчетный характер, из CASE-средств используется только любой текстовый редактор для оформления отчета по лабораторной работе.

Для успешной сдачи лабораторной работы необходимо предоставить отчет и пройти защиту в соответствии с разделом «Требования к отчету и защите».

Аудиторное время выполнения (под руководством преподавателя): 3 часа.

Время самостоятельной подготовки: 3 часа.

### **Теоретическое введение**

Один из факторов преодоления кризиса программирования – широкое внедрение во все стадии, на всех этапах жизненного цикла программного продукта стандартизации, охватывающей деятельность всех разработчиков ПС.

**Стандартизация** - деятельность, направленная на разработку и установление требований, норм, правил, характеристик, как обязательных для исполнения, так и рекомендуемых, обеспечивающих право потребителя на приобретение товаров надлежащего качества, а также право на безопасность и комфортабельность труда.

**Цель стандартизации** - достижение оптимальной степени упорядоченности в той или иной области посредством широкого и многократного использования установленных положений, требований, норм для решения реально существующих, планируемых или потенциальных задач.

**Основные результаты стандартизации** - повышение степени соответствия продукта (услуги), процессов их функциональному назначению, устранение технических барьеров в международном товарообмене, содействие научно-техническому прогрессу и сотрудничеству в различных областях.

**Объект стандартизации** - продукция, процесс, услуга, для которых разрабатывают те или иные требования, характеристики, параметры, правила и т. п. Стандартизации подвергается или весь объект в целом, или отдельные его составляющие (характеристики).

**Область стандартизации** - совокупность взаимосвязанных объектов стандартизации.

Есть следующие уровни стандартизации:

- ❖ международный (участие в стандартизации открыто для соответствующих органов любой страны);
- ❖ региональный (деятельность открыта только для соответствующих органов государств одного географического, политического или экономического региона);
- ❖ национальный - стандартизация в одном конкретном государстве. Этот уровень, в свою очередь, делится на:
  1. государственный;
  2. отраслевой;
  3. административно-территориальный;
  4. внутрифирменный.

Стандарты, согласно специфике объекта стандартизации, делятся на следующие виды:

- основополагающий стандарт;
- стандарт на продукцию, услугу;
- стандарт на работы (процессы);
- стандарт на методы испытаний.

*Основополагающие* стандарты устанавливают общие организационно-методические положения для определенной области деятельности, а также общетехнические требования, нормы и правила, обеспечивающие взаимопонимание, техническое единство и взаимосвязь различных областей науки, техники и производства в процессах создания и использования продукции, охрану окружающей среды, безопасность продукции, процессов и услуг для жизни, здоровья людей и имущества и другие общетехнические требования. *Основополагающие* стандарты имеют широкую область распространения и содержат общие положения для определенной области деятельности. Они могут служить основой для разработки других стандартов или иных нормативных или технических документов.

Стандарты *на продукцию* устанавливают требования, которым должна удовлетворять группа однородной продукции или конкретная продукция, с тем, чтобы обеспечить ее соответствие своему назначению. Стандарты на продукцию кроме требова-

ний соответствия назначению могут включать термины и определения, классификации, требования безопасности, экологичности, порядок приемки, методы контроля, требования к маркировке, упаковке, транспортированию и хранению, а иногда технологические или эксплуатационные требования.

Стандарты *на процессы* устанавливают основные требования, которым должен удовлетворять процесс, с тем, чтобы обеспечить соответствие процесса его назначению.

Стандарты *на методы контроля* устанавливают методы, способы, приемы, методики выполнения испытаний, измерений и анализа продукции в процессе ее производства, контроля качества и подтверждения соответствия.

Приведенные выше классификации и определения показывают всю трудность создания однозначного и, в то же время, лаконичного определения понятия **стандарт**. Поэтому ниже приводятся несколько дополняющих друг друга таких определений.

**Стандарт** – образец, эталон, модель, принимаемые за исходные для сопоставления с ними других подобных объектов. Как документ стандарт устанавливает комплекс норм, правил, требований к объекту стандартизации. Стандарт может быть разработан как на материальные предметы, так и на нормы, правила, требования в различных областях.

**Стандарт** - документ, разработанный на основе консенсуса и утвержденный признанным органом, в котором устанавливаются для всеобщего и многократного использования правила, общие принципы и характеристики, касающиеся различных видов деятельности или их результатов, и который направлен на достижение оптимальной степени упорядочения в определенной области.

В соответствии с определением, принятым Международной организацией по стандартизации (ИСО) и Европейской экономической комиссией ООН (ЕЭК ООН), **стандарт** определяется как документ технических условий или другой документ, доступный для общественности, разработанный при сотрудничестве и на основе консенсуса или утвержденный всеми заинтересованными в нем сторонами, основанный на обобщенных результатах науки, техники и практического опыта, направленный на достижение оптимальной пользы для общества и утвержденный органом, признанным на национальном, региональном или международном уровнях. Данное определе-

ние было принято почти всеми национальными организациями по стандартизации и является общепризнанным.

В российском законе "О техническом регулировании" понятие *стандарта* сформулировано так: это документ, в котором в целях добровольного многократного использования устанавливаются характеристики продукции, правила осуществления и характеристики процесса производства, эксплуатации, хранения, перевозки, реализации и утилизации, выполнения работ или услуг. Стандарт также может содержать требования к терминологии, символике, методам испытаний, эксплуатации оборудования, упаковке, маркировке или этикетке и правилам их нанесения.

На основе этих определений можно выделить несколько основных характеристик стандарта, который представляет собой документ:

- утвержденный признанным и компетентным органом;
- направленный на достижение оптимальной упорядоченности в определенной области;
- разработанный на основе сотрудничества и консенсуса, доступный для общественности;
- разработанный для пользы общества с целью технического прогресса и утвержденный всеми заинтересованными сторонами;
- для многократного или постоянного применения;
- для добровольного использования.

Стандарты должны периодически пересматриваться для внесения изменений, так как стандарты могут и тормозить использование современных технологий и средств. Это особенно важно в сфере ИТ.

#### *Стандарты в области программного обеспечения*

**Стандартизация** - принятие соглашения по спецификации, производству и использованию аппаратных и программных средств ВТ: установление и применение стандартов норм, правил и т. п. Такой подход позволяет в процессе разработки использовать данные и программы других разработчиков, осуществлять экспорт/импорт данных. Пример, OLE (Object Linking and Embedding - связывание и встраивание объектов) - реализовано на основе стандарта. Другой пример - GUI (графический интерфейс пользователя).

В соответствии с изложенным выше, ИТ-отрасль располагает набором общепризнанных стандартов на процессы жизненного цикла ПО, из которых можно выбирать подходящие для конкретной ситуации. Стандарты международные (ISO), профессиональные (IEEE), национальные (скажем, британские или австралийские) и относящиеся к конкретным сферам профессиональной деятельности (например, Международного агентства по атомной энергии) определяют 18 основных процессов жизненного цикла ПО. Они описывают наилучшие практические решения для создания и поддержки программного обеспечения на основе существующих технологий и накопленного опыта.

С другой стороны, в стандарте ISO/IEC 12207 написано "Стратегия разработки ПО требует перехода от множества разнообразных стандартов, процедур, методов, инструментальных средств к общему порядку, который позволит специалистам в области ПО "говорить на одном языке" при разработке и управлении ПО. Этот международный стандарт обеспечивает такой общий порядок".

Более 250 подкомитетов и организаций работают над стандартами в области информационных технологий. Более 1000 стандартов уже ими принято или разрабатывается. И этот процесс вряд ли когда-либо закончится.

Основным международным органом, который руководит разработкой стандартов, является Международная организация ИСО (ISO - International Organization for Standardization). Она была создана в 1946 г. двадцатью пятью национальными организациями. ИСО охватывает все сферы человеческой деятельности, кроме электротехники, электроники, радиолокации и приборостроения, относящиеся к компетенции Международной электротехнической комиссии (МЭК/IEC - International Electrical Commission). Поэтому имеют место и совместные стандарты ISO/IEC, прежде всего, в вопросах информационных технологий и микропроцессорной техники.

Сейчас в ИСО 120 стран. Россию представляет Госстандарт РФ в качестве комитета-члена ИСО.

С ИСО поддерживают контакты более 500 международных организаций.

МЭК была создана в 1881 г. на международной конференции, в которой приняли участие 13 стран.

В 1987 г. ИСО и МЭК объединили свою деятельность, создав единый орган JTC1 (Joint Committee 1 - Объединенный технический комитет 1), предназначенный

для стандартизации в области ИТ. ИТС1 имеет 17 подкомиссий: от техники программного обеспечения до языков программирования, компьютерной графики и обработки изображений.

В области информационных технологий стандартизацией занимается также международный союз по телекоммуникации ИТУ-Т (International Telecommunication Union-Telecommunications).

Существует также и Европейский комитет по стандартизации (СЕН).

При этом важно отметить, что международные стандарты не имеют статуса обязательности для стран-участниц.

В России головным органом в данной области является Государственный комитет РФ по стандартизации и метрологии (Госстандарт России). Он же занимается и сертификацией организаций. Глава Госстандарта является главным государственным инспектором РФ по надзору за государственными стандартами и обеспечением единства измерений, т.е. Госстандарт одновременно является службой по надзору за государственными стандартами и обеспечением единства измерений. Для этого по всей стране существуют подчиненные Госстандарту центры стандартизации, метрологии и сертификации.

По области стандартизации в программной инженерии выделяют:

**Общие стандарты** – они регламентируют вспомогательные процессы деятельности фирмы: правила общения с клиентами компании, сотрудников внутри фирмы, перечень общего ПО, не связанного с особенностями работы подразделения (ОС, ТП, ЭТ, архиваторы, антивирусы и т.п.). Эти же стандарты регламентируют и *рабочее пространство* - набор носителей для хранения различного рода информации со структурой каталогов, а также правила работы с хранимой информацией. Обычно выделяют рабочие пространства отдельно для:

- аналитиков;
- программистов;
- тестеров;
- специалистов отдела внедрения;
- технической поддержки.

**Стандарты на анализ и проектирование** – они обычно предназначены для аналитического отдела организации и регламентируют:

- применение методик структурного анализа или методов объектно-ориентированного анализа;
- описание бизнес-процессов программными средствами;
- правила хранения проектно-аналитической документации, правила кодирования имен файлов.

**Стандарты на разработку** – они ориентированы на повышение читабельности кода программы, сокращение времени разработки. Здесь выделяют стандарты на:

- формирование наименований: образование наименований, использование больших букв, формирование сложных наименований, имен процедур;
- структуру директорий разработки: сборки, исходных текстов, документации, БД;
- документирование исходного кода;
- регламент отладки программы;
- регламент использования конструкций языка программирования (например, запрет на использование выхода из цикла по оператору Exit, запрет на использование имен глобальных переменных в процедурах);
- визуальный интерфейс: что можно использовать, взаимное расположение, выравнивание на экране;
- сообщения, выдаваемые программой: виды сообщений, формирование текста сообщений (например, сами сообщения запрещается включать в текст программы, для этого они сводятся в текстовый файл, что упрощает национальную локализацию);
- регламент проектирования БД;
- регламент работы с ПО разработки (среда разработки, компиляторы и т. п.);
- ведение версий разрабатываемого ПО.

**Стандарты на тестирование** – они могут включать:

- стандарт на разработку методики тестирования;
- регламент на проведение испытаний и т. п.

Первым стандартом в области разработки ПО в России (тогда СССР) явился ГОСТ ЕСПД (Единая система программной документации - серия ГОСТ 19.XXX). Этот ГОСТ ориентирован на класс относительно простых программ небольшого объема, создаваемых отдельными программистами. Рядом находится ГОСТ группы 34 по ЖЦ Автоматизированных систем (АС) обработки информации и управления. Но сей-

час эти стандарты тоже устарели, так как в них практически не отражены вопросы создания распределенных АС. Поэтому проектным организациям приходится для больших АС создавать свои внутренние стандарты или использовать международные. В настоящее время ГОСТы группы 19 носят рекомендательный характер на всем пространстве СНГ и обязательны только при заключении контрактов на разработку ПО.

Из всего множества стандартов можно выделить четыре основных стандарта на процессы программной инженерии:

- ISO/IEC 12207, Software Lifecycle Processes, 1995;
- ISO/IEC 15504, Software Process Assessment (multipart), 1998;
- ISO 9001, Quality Management Systems-Requirements, 2000;
- ISO/IEC 12207:2008 System and software engineering — Software life cycle processes (ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»).

Эти стандарты позволяют разработчикам определять модели и процессы создания и поддержки программного обеспечения. С их помощью сторонние организации могут установить, использует ли разработчик данные процессы должным образом.

Перечисленные спецификации поддерживаются другими международными стандартами (такими как ISO 10007, Quality Management Systems — Guidelines for Configuration Management, 2003), профессиональными стандартами (например, IEEE 1042, Software Configuration Management, 1987) и стандартами конкретной сферы бизнеса (скажем, Европейское космическое агентство, ESA PSS-05-09, Guide to Software Configuration Management, 1992).

В ISO/IEC 12207 и ISO/IEC 15939, Software Measurement Process (2002) определены 18 основных процессов, объединяющих в себе зрелые практические решения программной инженерии. Для каждого процесса из этого набора разрабатываются, соответственно, один либо несколько уточняющих его международных, профессиональных или бизнес-стандартов.

### *Литература:*

19. Профессиональные стандарты специалистов в области информационных технологий, с. 26, 27.

13. 2. СТАНДАРТИЗАЦИЯ ПРОЦЕССОВ СОЗДАНИЯ ПРОГРАММНОГО ПРОДУКТА, с. 25 – 41.

17. 2. Стандартизация основных процессов жизненного цикла создания программных продуктов, с. 20 – 45.

### ***Контрольные вопросы для самопроверки***

1. Для чего нужна стандартизация?
2. Какие международные организации занимаются разработкой стандартов в области создания ПО?
3. Как распределяются стандарты по областям применения?
4. В каком случае применение стандартов является обязательным?
5. Что может явиться объектом стандартизации?

### ***Задания к лабораторной работе***

В табл. 4.1 указаны конкретные стандарты, группы стандартов, а также вопросы по теме стандартизации.

Таблица 4.1

<b>№ п/п</b>	<b>Вопрос</b>
1-1	ГОСТ Р ИСО/МЭК 9294-93: назначение, структура
1-2	ГОСТ Р ИСО/МЭК 9126-93: назначение, структура
1-3	ГОСТ Р ИСО/МЭК 9127-94: назначение, структура
1-4	ГОСТ Р ИСО/МЭК 8631-94: назначение, структура
1-5	ГОСТ Р 56713-2015 Системная и программная инженерия. Содержание информационных продуктов процесса жизненного цикла систем и программного обеспечения (документация)
1-6	Международные стандарты в области программирования, разработанные ISO/IEC (не менее 3). Их назначение, структура.
1-7	Группы организации ITU-T, ориентированные на стандартизацию информационных технологий.
1-8	Назначение, структура и краткое содержание стандарта ГОСТ Р ИСО/МЭК 12207-2010
1-9	Структура объединенного технического комитета JTC1: группы и подгруппы
1-10	Организация CEN: назначение, структура, результаты деятельности
1-11	Структура и краткое содержание стандарта IEEE 1074-1995
1-12	Стандарты IEEE в области информационных технологий
1-13	Стандарт ISO/IEC/IEEE 24765-2010: назначение, краткое содержание

№ п/п	Вопрос
1-14	Виды нормативных документов, рекомендуемых ИСО/МЭК, их назначение, отличия
1-15	Структура ИСО
1-16	Госстандарт России: функции, структура
1-17	Стандарт ИСО/МЭК 9126-2001: назначение, краткое содержание
1-18	Стандарт ISO/IEC 14102-1995: назначение, краткое содержание
1-19	Структура ЕСПД
1-20	Назначение, структура и краткое содержание стандарта ГОСТ Р ИСО/МЭК 12207-1995
1-21	Назначение, структура и краткое содержание стандарта ISO/IEC CD 12207-2:1995
1-22	Международные стандарты, регламентирующие процесс тестирования программного продукта: перечень, краткое содержание
1-23	Международные стандарты, регламентирующие структуру документации программной системы: перечень, краткое содержание
1-24	Система стандартов ISO 9001: назначение структура
1-25	ГОСТ Р 56920-2016 Системная и программная инженерия. Тестирование программного обеспечения. Часть 1. Понятия и определения
1-26	ГОСТ Р 56921-2016 Системная и программная инженерия. Тестирование программного обеспечения. Часть 2. Процессы тестирования
1-27	Процессы ЖЦ в соответствии со стандартом IEEE 1074-1995
1-28	ГОСТ Р 56922-2016 Системная и программная инженерия. Тестирование программного обеспечения. Часть 3. Документация тестирования
1-29	Характеристики качества ПО в соответствии со стандартом ISO 9126:1991
1-30	Документирование процесса разработки ПО в соответствии со стандартом ГОСТ Р ИСО/МЭК 9294-93
2-1	Назначение, структура и краткое содержание стандарта ГОСТ Р ИСО/МЭК 27001:2013
2-2	Назначение, структура и краткое содержание стандарта ГОСТ Р ИСО/МЭК 17799
2-3	Показатели надежности ПО в соответствии со стандартом ISO 9126:1991
2-4	ГОСТ Р ИСО/МЭК 12119-2000: назначение, структура, основные определения
2-5	Международные организации, занимающиеся стандартизацией в области создания программного обеспечения
2-6	Система стандартов ISO 9000: назначение структура
2-7	Стандарт ИСО/МЭК ТО 9294:1990: назначение краткое содержание
2-8	Назначение, структура и краткое содержание стандарта ГОСТ Р ИСО/МЭК 9126-93
2-9	ГОСТ Р ИСО/МЭК 8631-94: назначение, структура
2-10	Основные работы по приобретению ПО в соответствии со стандартом ISO 12207:1995
2-11	Характеристика качества ПО в соответствии со стандартом ISO 12207:1995

№ п/п	Вопрос
2-12	Документирование процесса разработки ПО в соответствии со стандартом ISO 12207:1995
2-13	Рекомендации по обеспечению качества ПО в соответствии со стандартом ISO 12207:1995
2-14	Управление процессом разработки ПО в соответствии со стандартом ISO 12207:1995
2-15	Назначение, структура и краткое содержание стандарта ГОСТ Р ИСО/МЭК 9127-94
2-16	Назначение, структура и краткое содержание стандарта ГОСТ Р ИСО/МЭК 8631-94
2-17	Стандарт Х.800: назначение, структура, краткое содержание
2-18	Стандарт Х.509: назначение, структура, краткое содержание
2-19	Решение вопросов информационной безопасности на основе международного стандарта ISO/IEC 7498-2
2-20	Основные задачи, решаемые международным стандартом ИСО/МЭК 15408-99
2-21	Задачи, решаемые ГОСТ Р 51583-2000
2-22	Задачи, решаемые ГОСТ Р 51275-99
2-23	Задачи, решаемые ГОСТ Р 50739-95
2-24	Отечественные стандарты по тестированию ПО: перечень, названия, краткая характеристика
2-25	Структура тестирования ПО в соответствии со стандартами
2-26	Документация процесса тестирования ПО в соответствии со стандартами
2-27	Внутрифирменные (внутрикорпоративные) стандарты: назначение, классификация
2-28	ГОСТ Р 57100-2016 Системная и программная инженерия. Описание архитектуры
2-29	ГОСТ Р 57101-2016 Системная и программная инженерия. Процессы жизненного цикла. Управление проектом
2-30	ГОСТ Р 56939-2016 Защита информации. Разработка безопасного программного обеспечения. Общие требования
2-31	ГОСТ Р 51188-98 Защита информации. Испытания программных средств на наличие компьютерных вирусов. Типовое руководство.
2-32	ISO/IEC/IEEE 29148:2011 «Recommended Practice for Software Requirements Specifications»
2-33	Стандарты на оформление требований к программному обеспечению
2-34	ГОСТ Р 51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию
2-35	ГОСТ Р 54593-2011 Информационные технологии. Свободное программное обеспечение. Общие положения
2-36	<a href="#">ГОСТ Р ИСО 9127-94</a> Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов
2-37	<a href="#">ГОСТ Р ИСО/МЭК 12119-2000</a> Информационная технология. Пакеты программ. Требования к качеству и тестирование

№ п/п	Вопрос
2-38	ГОСТ Р ИСО/МЭК 9126-93 Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению
2-39	ГОСТ Р ИСО/МЭК 14764-2002 Информационная технология. Сопровождение программных средств
2-40	ГОСТ Р ИСО/МЭК 15910-2002 Информационная технология. Процесс создания документации пользователя программного средства

### ***Методические указания и порядок выполнения работы***

1. В соответствии с вариантом выбрать из табл. 4.1 номер ГОСТа, название организации или тему анализа.
2. В соответствии с выбором создать отчет объемом 3, 4 страницы, содержащий:
  - для названия конкретного ГОСТа – краткое описание стандарта и его назначение по всем рассмотренным в разделе «**Теоретическое введение**» классификациям;
  - для группы стандартов – перечень стандартов, их краткие характеристики, годы издания.
  - для названия организации – назначение организации, дата образования, структура.

### ***Требования к отчету и защите***

Отчет по выполненной лабораторной работе должен включать:

1. Исходное задание.
2. Текст реферата по заданной теме.

Защита проводится во время лабораторных занятий и представляет собой перечень ответов на вопросы по теоретическим аспектам изучаемой темы и практическим результатам, полученным в ходе выполнения лабораторной работы. Примеры подобных вопросов приведены ранее, в разделе «Контрольные вопросы для самопроверки». Результатом каждого ответа является бинарная оценка «правильно/неправильно». Защита считается состоявшейся, если студент набрал первыми три правильных ответа. В противном случае защита не состоялась и может быть повторена не ранее, чем через день.

## **ЗАКЛЮЧЕНИЕ**

В данном учебно-методическом пособии представлены материалы для полноценного проведения девяти лабораторных работ по дисциплине «Программная инженерия». Каждая лабораторная работа предваряется обширным теоретическим материалом, позволяющим понять контекст выполняемого задания, его место в повышении практических навыков и умений студента в изучаемой области будущей профессии, а также осмысленно и творчески выполнить требуемое задание. Все задания индивидуализированы. Обширный список литературы позволяет студентам расширять и углублять знания по рассматриваемым в лабораторных работах темам.

## ЛИТЕРАТУРА:

1. Программная инженерия: учебное пособие / сост. Т. В. Киселева; Северо-Кавказский федеральный университет. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2017. – Часть 1. – 137 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=467203> (дата обращения: 16.05.2022). – Библиогр. в кн. – Текст: электронный.
2. Программная инженерия: учебное пособие / сост. Т. В. Киселева; Северо-Кавказский федеральный университет. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2017. – Часть 2. – 100 с.: схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=494790> (дата обращения: 16.05.2022). – Текст: электронный.
3. Программная инженерия: [16+] / сост. Т. В. Киселева. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2018. – Часть 3. – 130 с.: ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=563341> (дата обращения: 16.05.2022). – Библиогр.: с. 128. – Текст : электронный.
4. Романов, Е. Л. Программная инженерия: учебное пособие: [16+] / Е. Л. Романов; Новосибирский государственный технический университет. – Новосибирск: Новосибирский государственный технический университет, 2017. – 395 с. : табл., схем., ил. – (Учебники НГТУ). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=573945> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-7782-3455-0. – Текст: электронный.
5. Абдулаев, В. И. Программная инженерия: учебное пособие: [16+] / В. И. Абдулаев. – Йошкар-Ола : Поволжский государственный технологический университет, 2016. – Часть 1. Проектирование систем. – 168 с.: схем., табл., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=459449> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-8158- 1767-8 (ч. 1); ISBN 978-5-8158- 1766-1. – Текст: электронный.
6. Антамошкин, О. А. Программная инженерия. Теория и практика: учебник/ О. А. Антамошкин; Сибирский федеральный университет. – Красноярск: Сибирский федеральный университет (СФУ), 2012. – 247 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=363975> (дата обращения: 16.05.2022). – Библиогр.: с. 240. – ISBN 978-5-7638-2511-4. – Текст : электронный.
7. Перл, И. А. Введение в методологию программной инженерии: учебное пособие: [16+] / И. А. Перл, О. В. Калёнова. – Санкт-Петербург: Университет ИТМО, 2019. – 53 с.: ил., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=566776> (дата обращения: 16.05.2022). – Библиогр. в кн. – Текст : электронный.
8. Мейер, Б. Объектно-ориентированное программирование и программная инженерия: учебное пособие: [16+] / Б. Мейер. – 2-е изд., испр. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 286 с. : ил. – Режим доступа: по под-

- писке. – URL: <https://biblioclub.ru/index.php?page=book&id=429034> (дата обращения: 16.05.2022). – Текст: электронный.
9. Соловьев, Н. А. Введение в программную инженерию: учебное пособие / Н. А. Соловьев, Л. А. Юркевская; Оренбургский государственный университет. – Оренбург: Оренбургский государственный университет, 2017. – 112 с.: схем., табл., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=481815> (дата обращения: 16.05.2022). – Библиогр.: с. 83. – ISBN 978-5-7410-1685-5. – Текст: электронный.
10. Ехлаков, Ю. П. Введение в программную инженерию: учеб. пособие / Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Томский государственный университет систем управления и радиоэлектроники, 2011. – 148 с. : табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=209001> (дата обращения: 16.05.2022). – ISBN 978-5-4332-0018-0. – Текст: электронный.
11. Ехлаков, Ю. П. Экономика программной инженерии: учеб. пособие / Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Эль Контент, 2013. – 132 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=480604> (дата обращения: 16.05.2022). – Библиогр.: с. 124-125. – ISBN 978-5-4332-0126-2. – Текст: электронный.
12. Программная инженерия: лабораторный практикум: практикум: [16+] / Д. Г. Лагерев, Д. А. Коростелев, А. А. Азарченков, Е. В. Коптенюк. – Москва; Берлин: Директ-Медиа, 2021. – 157 с.: табл., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=602232> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-4499-2105-5. – Текст: электронный.
13. Ехлаков, Ю. П. Управление программными проектами: учебник / Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Томский государственный университет систем управления и радиоэлектроники, 2015. – 217 с. : схем., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=480634> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-86889-723-8. – Текст: электронный.
14. Мякишев, Д. В. Принципы и методы создания надежного программного обеспечения АСУТП: учебное пособие: [16+] / Д. В. Мякишев. – 2-е изд. – Москва; Вологда: Инфра-Инженерия, 2021. – 116 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=617225> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-9729-0674-1. – Текст: электронный.
15. Ехлаков, Ю. П. Планирование и организация вывода программного продукта на рынок: учебное пособие: [16+] / Ю. П. Ехлаков. – Томск: ТУСУР, 2017. – 121 с.: ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=481009> (дата обращения:

- 16.05.2022). – Библиогр.: с. 115-117. – ISBN 978-5-4332-0258-0. – Текст: электронный.
16. Петрухин, В. А. Методы и средства инженерии программного обеспечения: курс: учебное пособие / В. А. Петрухин, Е. М. Лаврищева; Национальный Открытый Университет "ИНТУИТ". – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2008. – 424 с. : табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=234553> (дата обращения: 16.05.2022). – Текст: электронный.
17. Ехлаков, Ю. П. Управление программными проектами: учебное пособие/ Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Эль Контент, 2014. – 140 с.: схем., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=480462> (дата обращения: 16.05.2022). – Библиогр.: с. 128-130. – ISBN 978-5-4332-0163-7. – Текст: электронный.
18. Москвитин, А. А. Решение задач на компьютерах: учебное пособие:/ А. А. Москвитин. – Москва; Берлин : Директ-Медиа, 2015. – Часть 2. Разработка программных средств. – 429 с.: ил., схем., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=273667> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-4475-3646-6. – DOI 10.23681/273667. – Текст: электронный.
19. Соловьев, Н. Системы автоматизации разработки программного обеспечения: учебное пособие / Н. Соловьев, Е. Чернопрудова; Оренбургский государственный университет. – Оренбург: Оренбургский государственный университет, 2012. – 191 с.: ил., схем., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=270302> (дата обращения: 16.05.2022). – Библиогр.: с. 182-183. – Текст: электронный.
20. Кайгородцев, Г. И. Введение в курс метрической теории и метрологии программ: учебник/ Г. И. Кайгородцев. – Новосибирск: Новосибирский государственный технический университет, 2011. – 190 с. : табл., схем., ил. – (Учебники НГТУ). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=435984> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-7782-1648-8. – Текст: электронный.
21. Дроздов, С. Н. Структуры и алгоритмы обработки данных: учебное пособие: [16+] / С. Н. Дроздов. – Таганрог: Южный федеральный университет, 2016. – 228 с.: схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=493032> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-9275-2242-2. – Текст: электронный.
22. Нехорошкова, Л. Г. Информационное моделирование и анализ требований : учебное пособие: [16+] / Л. Г. Нехорошкова ; Поволжский государственный технологический университет. – Йошкар-Ола: Поволжский государственный технологический университет, 2020. – 146 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=615678> (дата обращения: 16.05.2022). – Библиогр.: с. 113-114. – ISBN 978-5-8158-2209-2. – Текст : электронный.

23. Зыкина, А. В. Методы принятия оптимальных решений: учебное пособие: [16+] / А. В. Зыкина, О. Н. Канева, Т. Ю. Финк; Омский государственный технический университет. – Омск : Омский государственный технический университет (ОмГТУ), 2020. – 178 с.: ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=683053> (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-8149-3175-7. – Текст : электронный.
24. Программирование и основы алгоритмизации: учебное пособие / В. К. Зольников, П. Р. Машевич, В. И. Анциферова, Н. Н. Литвинов; Федеральное агентство по образованию, Воронежская государственная лесотехническая академия. – Воронеж: Воронежская государственная лесотехническая академия, 2011. – 341 с.: ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=142309> (дата обращения: 29.05.2022). – Текст : электронный.
25. Князьков, В. С. Введение в теорию автоматов / В. С. Князьков, Т. В. Волченская. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2008. – 78 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=234134> (дата обращения: 29.05.2022). – Текст : электронный.

## Список условных сокращений

АС – автоматизированная система.

БД – база данных.

ГОСТ – государственный стандарт.

ДПС - диаграмма переходов состояний

ЕСПД - единая система программной документации

ЖЦ – жизненный цикл.

КГТУ – Калининградский государственный технический университет.

ИТ – информационные технологии.

КУВ – коэффициент уменьшения вычислений

ЛВС – локальная вычислительная сеть.

НАМ - начальный адрес массива.

НСД - несанкционированный доступ.

ПО программное обеспечение

ПрО – предметная область.

ПС – программная система.

СНГ – содружество независимых государств.

СПУ – сетевое планирование и управление.

ЧПУ – числовое программное управление

CASE – Computer-Aided Software Engineering.

ERD - Entity-Relationship Diagrams

FK - Foreign Key

GUI – graphic user interface.

IEC - International Electrical Commission.

ITU-T - International Telecommunication Union-Telecommunications.

ISO - International Organization for Standardization.

JTC1 - Joint Committee 1.

OLE - object Linking and Embedding.

Локальный электронный методический материал

Леонид Григорьевич Высоцкий

**ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ПИ)**

Редактор Г.А. Смирнова

Уч.-изд. л. 2,7. Печ. л. 3,75.

Издательство федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Калининградский государственный технический университет».  
236022, Калининград, Советский проспект, 1