

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шебзухова Татьяна Александровна
Должность: Директор Пятигорского института (филиал) Северо-Кавказского
федерального университета
Дата подписания: 25.07.2023 14:42:00
Уникальный программный ключ:
d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ
ПО ДИСЦИПЛИНЕ УЧЕБНЫЙ ПРОФЕССИОНАЛЬНЫЙ ПРОЕКТ

Направление подготовки	09.04.02
Направленность (профиль)	Информационные системы и технологии «Технологии работы с данными и знаниями, анализ информации»
Квалификация выпускника	Магистр

Пятигорск, 2023

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

СОДЕРЖАНИЕ

1. Наименование лабораторных работ	3
2. Содержание лабораторных работ	3
Тема 1. CASE-технологии проектирования информационных систем	3
Лабораторная работа №1. Разработка моделей информационной системы	4
Тема 2. Методы объектно-ориентированного и функционального моделирования информационных систем	10
Лабораторная работа №2. Разработка диаграмм бизнес-процессов	10
Тема 3. Предпроектное обследование предметной области. Согласование технического задания на разработку информационной системы	23
Лабораторная работа №3. Разработка технического задания на проект информационной системы	23
Тема 4. Разработка прототипа информационной системы. Тестирование взаимодействия программного интерфейса с базой данных	26
Лабораторная работа № 4. Разработка прототипа информационной системы	26
Тема 5. Спецификация функциональных и обеспечивающих подсистем информационной системы	35
Лабораторная работа № 5. Выбор архитектуры информационной системы	35
Тема 6. Контроль качества и надежности информационной системы	40
Лабораторная работа № 6. Оценка качества и надежности информационной системы	40
Тема 7. Оценка экономического эффекта внедрения информационной системы	52
Лабораторная работа № 7. Расчеты экономических показателей	52
3. КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ	58
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ	59
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	59

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

1. Наименование лабораторных работ

Раздел 1. Методология и инструментарий проектирования информационных систем

Лабораторная работа 1. Разработка моделей информационной системы.

Содержание: Применение методологии DFD, IDEF0, IDEF3

Лабораторная работа 2. Разработка диаграмм бизнес-процессов.

Содержание: Диаграммы RationalRose. Диаграммы бизнес-процессов.

Раздел 2. Реализация этапов жизненного цикла проекта информационной системы

Лабораторная работа 3. Разработка технического задания на проект информационной системы.

Содержание: Техничко-экономический анализ предметной области. Структура технического задания на разработку информационной системы.

Лабораторная работа 4. Разработка прототипа информационной системы.

Содержание: Разработка структуры базы данных. Разработка программного интерфейса информационной системы. Тестирование функциональности прототипа информационной системы.

Тема 5. Спецификация функциональных и обеспечивающих подсистем информационной системы

Лабораторная работа 5. Выбор архитектуры информационной системы.

Содержание: Архитектура информационной системы. Спецификация базы данных и программных модулей информационной системы. Спецификация аппаратно-программно-аппаратной инфраструктуры информационной системы.

2. Содержание лабораторных работ

Тема 1. CASE-технологии проектирования информационных систем

Лабораторная работа №1. Разработка моделей информационной системы.

Цель и содержание работы: изучить возможности программы BPWin и ее установку.

Организационная форма занятий: решение конкретных задач проектирования

Вопросы для обсуждения на лабораторном занятии: Возможности BPWin

Теоретическое обоснование

При создании информационной системы первым этапом работ является проведение предпроектного обследования предприятия. На этом этапе производится сбор информации о предприятии, его целях и задачах, структуре и финансово-хозяйственной деятельности. Также изучается внешнее окружение, взаимодействующее с предприятием, и в котором предприятие осуществляет свою деятельность. В результате предпроектного обследования должно быть сформировано описание организационной структуры предприятия, описание функционального взаимодействия подразделений предприятия и информационная схема, содержащая информационные потоки предприятия, места получения и накопления информации и прочие информационные аспекты деятельности предприятия. В целом в результате предпроектного обследования должно быть получено комплексное описание предприятия и его бизнеса.

Методология функционального моделирования IDEF0 – это технология описания системы в целом как множества взаимозависимых действий, или функций. Важно

отметить функциональную направленность IDEF0 – функции системы исследуются независимо от объектов, которые обеспечивают их выполнение. «Функциональная» точка зрения позволяет четко разделить аспекты назначения системы от аспектов ее физической реализации.

IDEF0 методология представляет собой формализованный подход к созданию функциональных моделей – структурных схем изучаемого процесса или системы. Схемы строятся по иерархическому принципу с необходимой степенью подробности и помогают разобраться в том, что происходит в изучаемой системе или процессе, какие функции выполняются и в какие отношения вступают между собой и с окружающей средой ее функциональные блоки. IDEF0-модель принципиально не может ответить на вопросы о том, как протекают процессы в системе во времени и в пространстве.

В этом случае рекомендуется переходить к другим моделям – математическим, имитационным, описывающим процессы в функциональных блоках IDEF0-модели. По терминологии, принятой в исследовании процессов, IDEF0-модели относятся к классу концептуальных. Именно концептуальные модели являются основой построения имитационных и математических моделей.

Наиболее часто IDEF0 применяется как технология исследования и проектирования систем на логическом уровне. По этой причине она, как правило, используется на ранних этапах разработки проекта, для сбора данных и моделирования процесса «как есть». Результаты IDEF0-анализа могут применяться при проведении проектирования с использованием моделей для описания потоков работ IDEF3 и потоков данных DFD.

IDEF0 сочетает в себе небольшую по объему графическую нотацию (она содержит только два обозначения: блоки и стрелки) со строгими и четко определенными рекомендациями, в совокупности предназначенными для построения качественной и понятной модели системы.

Первый шаг при построении модели IDEF0 заключается в определении назначения модели – набора вопросов, на которые должна отвечать модель. **Границы моделирования** предназначены для обозначения ширины охвата предметной области и глубины детализации и являются логическим продолжением уже определенного назначения модели. Следующим шагом указывается предполагаемая целевая аудитория, для нужд которой создается модель. Зачастую от выбора целевой аудитории зависит уровень детализации, с которым должна создаваться модель. Перед построением модели необходимо иметь представление о том, какие сведения о предмете моделирования уже известны, какие дополнительные материалы и (или) техническая документация для понимания модели могут быть необходимы целевой аудитории, какие язык и стиль изложения являются наиболее подходящими.

Под **точкой зрения** понимается перспектива, с которой наблюдалась система при построении модели. Точка зрения выбирается таким образом, чтобы учесть уже обозначенные границы моделирования и назначение модели. Однажды выбранная точка зрения остается неизменной для всех элементов модели. При необходимости могут быть созданы другие модели, отображающие систему с других точек зрения. Вот несколько примеров точек зрения при построении моделей: клиент, поставщик, владелец, редактор.

Действие, обычно в IDEF0 называемое функцией, обрабатывает или переводит входные параметры (сырье, информацию и т.п.) в выходные. Поскольку модели IDEF0 представляют систему

как множество в первую очередь должна функция, описывающая **контекстная функция**. на диаграммах как прямоугольники, или Имена функций в IDEF0

быть определена система в целом. Функции изображаются поименованные функциональные блоки.

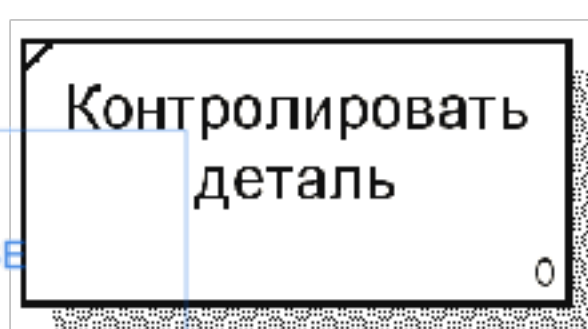


Рисунок 1.1 Функциональный блок IDEF0

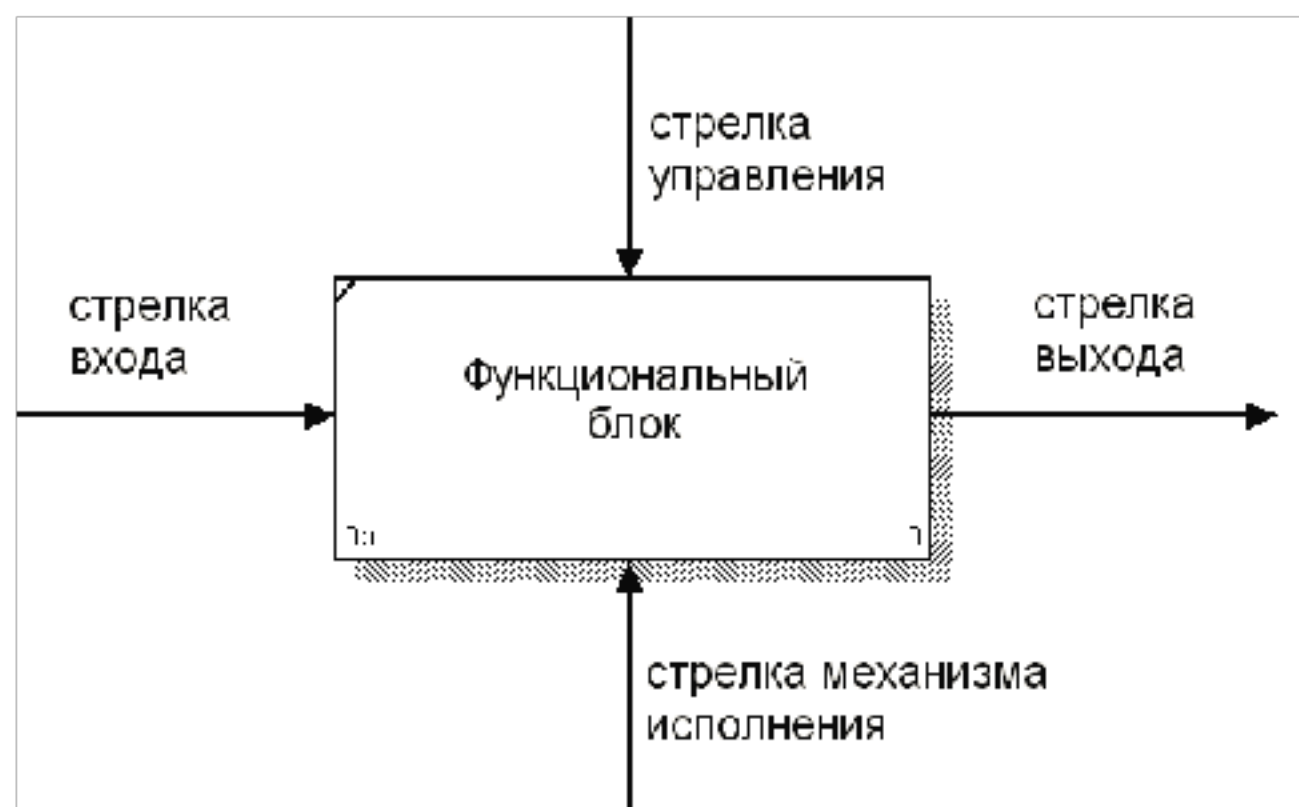


Рисунок 1.3 – Типы стрелок функционального блока

Для названия стрелок, как правило, употребляются имена существительные. Стрелки могут представлять собой людей, места, вещи, идеи или события. Как и в случае с функциональными блоками, присвоение имен всем стрелкам на диаграмме является только необходимым условием для понимания читателем сути изображенного. Отдельное описание каждой стрелки в текстовом виде может оказаться критическим фактором для построения точной и полезной модели.

Стрелки входа. Вход представляет собой сырье, или информацию, потребляемую или преобразуемую функциональным блоком для производства выхода. Стрелки входа всегда направлены в левую сторону прямоугольника, обозначающего в IDEF0 функциональный блок. Наличие входных стрелок на диаграмме не является обязательным, так как возможно, что некоторые блоки ничего не преобразуют и не изменяют. Примером блока, не имеющего входа, может служить «принятие решения руководством», где для принятия решения анализируется несколько факторов, но ни один из них непосредственно не преобразуется и не потребляется в результате принятия какого-либо решения.

Стрелки управления. Стрелки управления отвечают за регулирование того, как и когда выполняется функциональный блок, и, если он выполняется, какой выход получается в результате его выполнения. Так как управление контролирует поведение функционального блока для обеспечения создания желаемого выхода, каждый функциональный блок должен иметь, как минимум, одну стрелку управления. Стрелки управления всегда входят в функциональный блок сверху.

Управление часто существует в виде правил, инструкций, законов, политики, набора необходимых процедур или стандартов. Влияя на работу блока, оно непосредственно не потребляется и не трансформируется в результате. Может оказаться, что целью функционального блока является как раз изменение того или иного правила, инструкции, стандарта и т.п. В этом случае стрелка, содержащая соответствующую информацию, должна рассматриваться не как управление, а как вход функционального блока.

Управление можно рассматривать как специфический вид входа. В случаях, когда неясно, относить ли стрелку к входу или к управлению, предпочтительно относить ее к управлению до момента, пока неясность не будет разрешена.

Стрелки выхода. Выход – это продукция или информация, получаемая в результате работы функционального блока. Каждый блок должен иметь, как минимум, один выход. Действие, которое не производит никакого четко определяемого выхода, не должно моделироваться вообще (по меньшей мере, должно рассматриваться в качестве одного из первых кандидатов на исключение из модели).

При моделировании непродуцированных предметных областей выходами, как правило, являются данные, в каком-либо виде обрабатываемые функциональным блоком. В этом случае важно, чтобы названия стрелок входа и выхода были достаточно различимы

по своему смыслу. Например, блок «Прием пациентов» может иметь стрелку «Данные о пациенте» как на входе, так и на выходе. В такой ситуации входящую стрелку можно назвать «Предварительные данные о пациенте», а исходящую – «Подтвержденные данные о пациенте».

Стрелки механизма исполнения. Механизмы являются ресурсом, который непосредственно исполняет моделируемое действие. С помощью механизмов исполнения могут моделироваться: ключевой персонал, техника и (или) оборудование. Стрелки механизма исполнения могут отсутствовать в случае, если оказывается, что они не являются необходимыми для достижения поставленной цели моделирования.

Комбинированные стрелки. В IDEF0 существует пять основных видов комбинированных стрелок: выход – вход, выход – управление, выход – механизм исполнения, выход – обратная связь на управление и выход – обратная связь на вход.

Стрелка *выход – вход* применяется, когда один из блоков должен полностью завершить работу перед началом работы другого блока (рисунок 4).

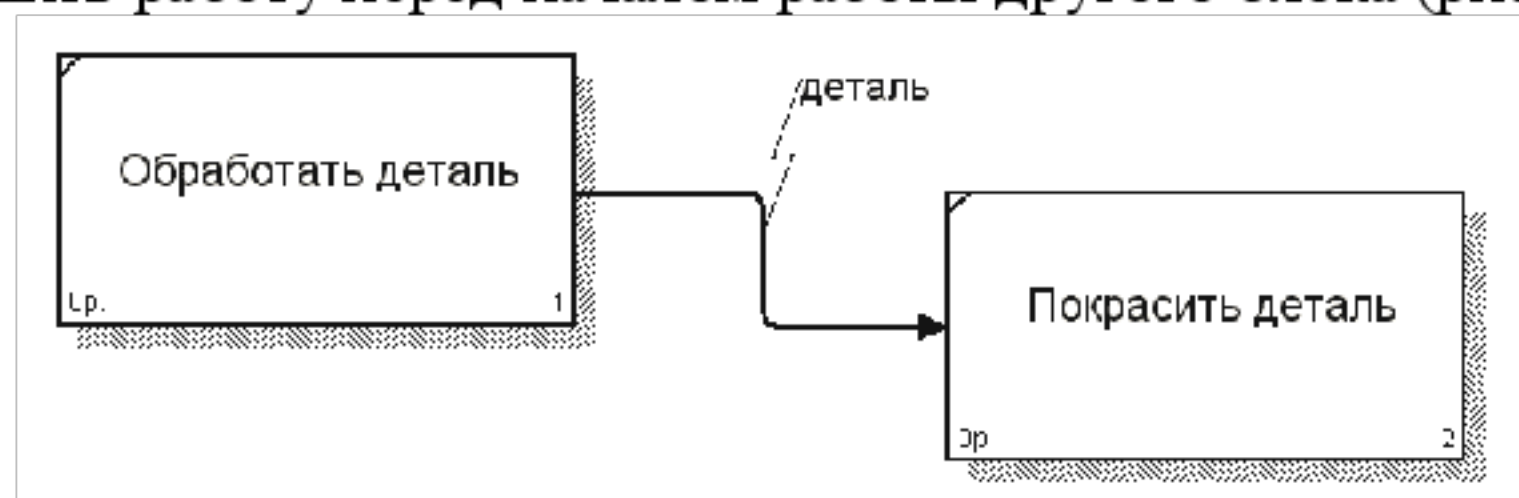


Рисунок 1.4 – Комбинированная стрелка *выход – вход*

Стрелка *выход - управление* отражает ситуацию преобладания одного блока над другим, когда один блок управляет работой другого. На рисунке 5 принципы формирования инвестиционного портфеля управляют поведением брокеров на бирже.

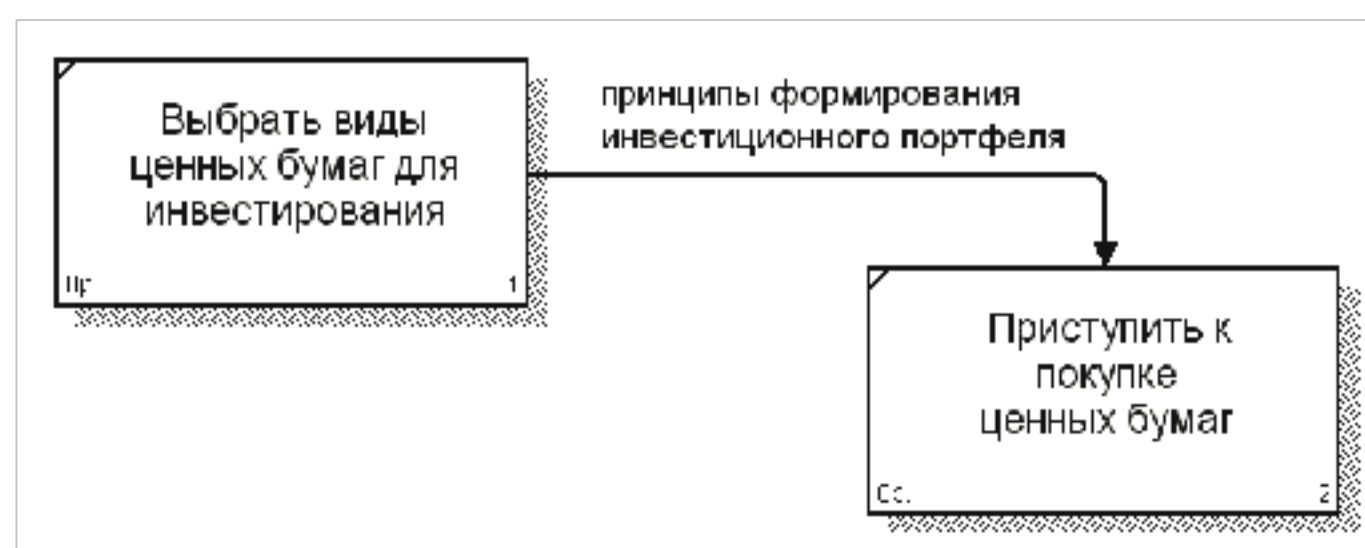
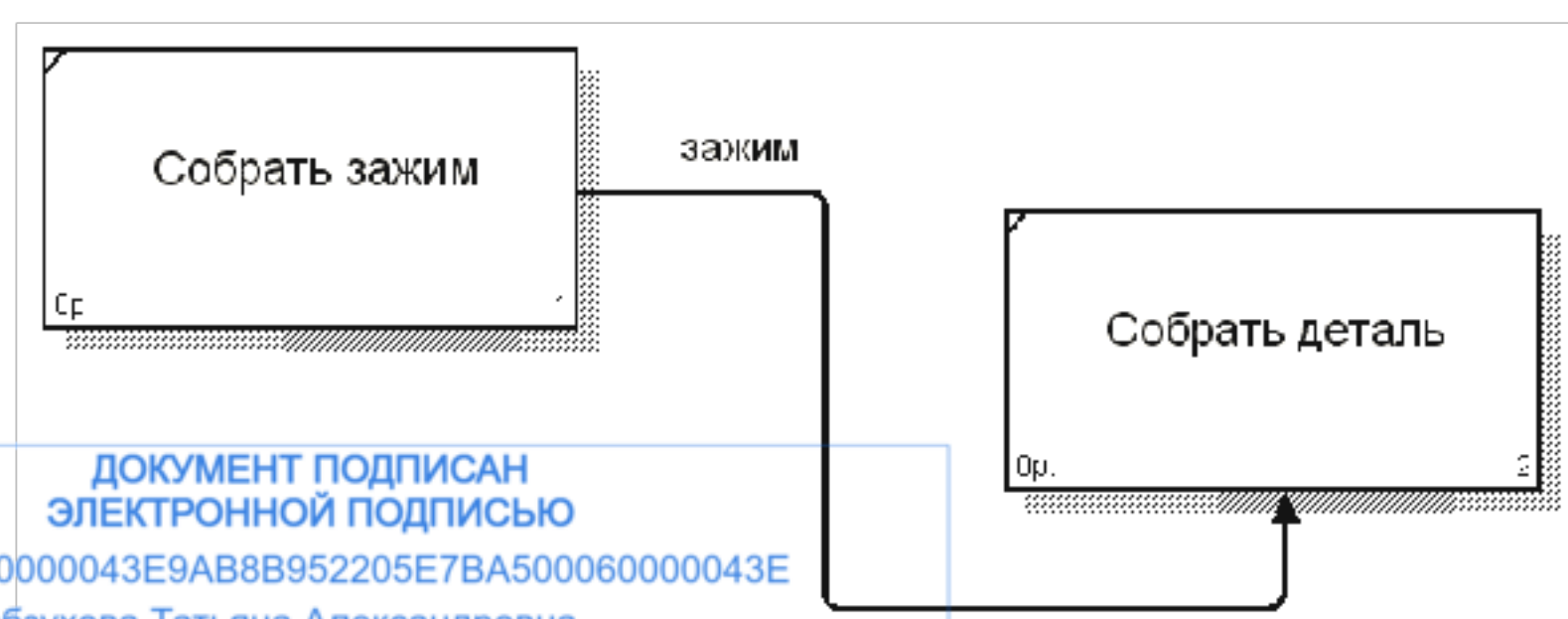


Рисунок 1.5 – Комбинированная стрелка *выход – управление*

Стрелки *выход - механизм исполнения* встречаются реже и отражают ситуацию, когда выход одного функционального блока применяется в качестве оборудования для работы другого блока. На рисунке 6 зажим - устройство, используемое для закрепления детали во время ее сборки, должен быть собран для того, чтобы выполнить сборку детали.



ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Рисунок 1.6 – Комбинированная стрелка *выход - механизм исполнения*

Действителен: с 19.08.2022 по 19.08.2023

Обратные связи на вход и на управление применяются в случаях, когда зависимые блоки формируют обратные связи для управляющих ими блоков. На рисунке 7 получаемая от брокеров информация о текущих биржевых курсах применяется для корректировки стратегии игры на бирже.

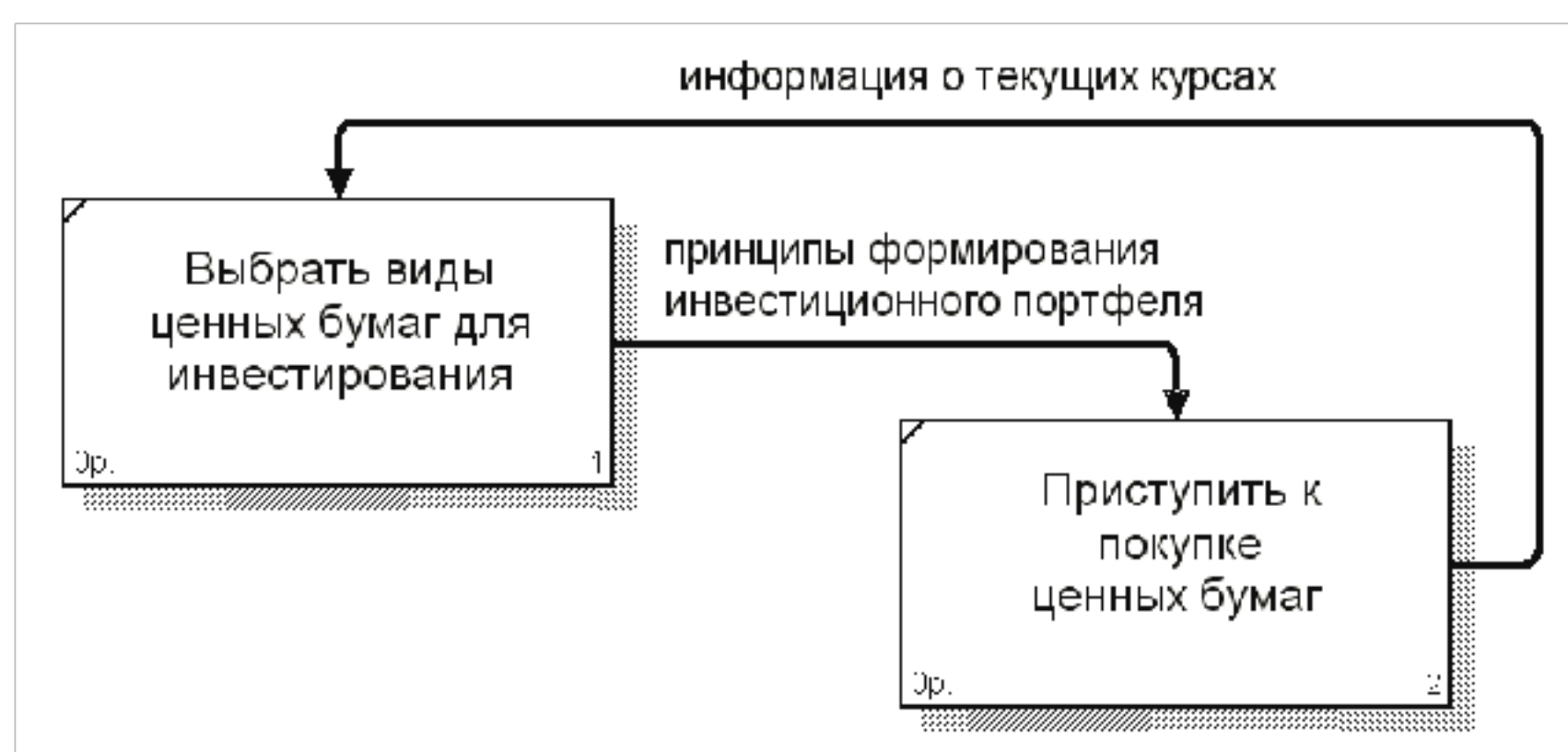


Рисунок 1.7 – Комбинированная стрелка *выход – обратная связь на управление*

Стрелка *выход - обратная связь на вход* обычно применяется для описания циклов повторной обработки чего-либо. Рисунок 8 может служить примером применения стрелки такого типа. Кроме того, связи *выход – обратная связь на вход* могут применяться в случае, если бракованная продукция может заново использоваться в качестве сырья, как это происходит, например, при производстве оконного стекла, когда разбитое в процессе производства стекло перемалывается и переплавляется заново вместе с обыкновенным сырьем.

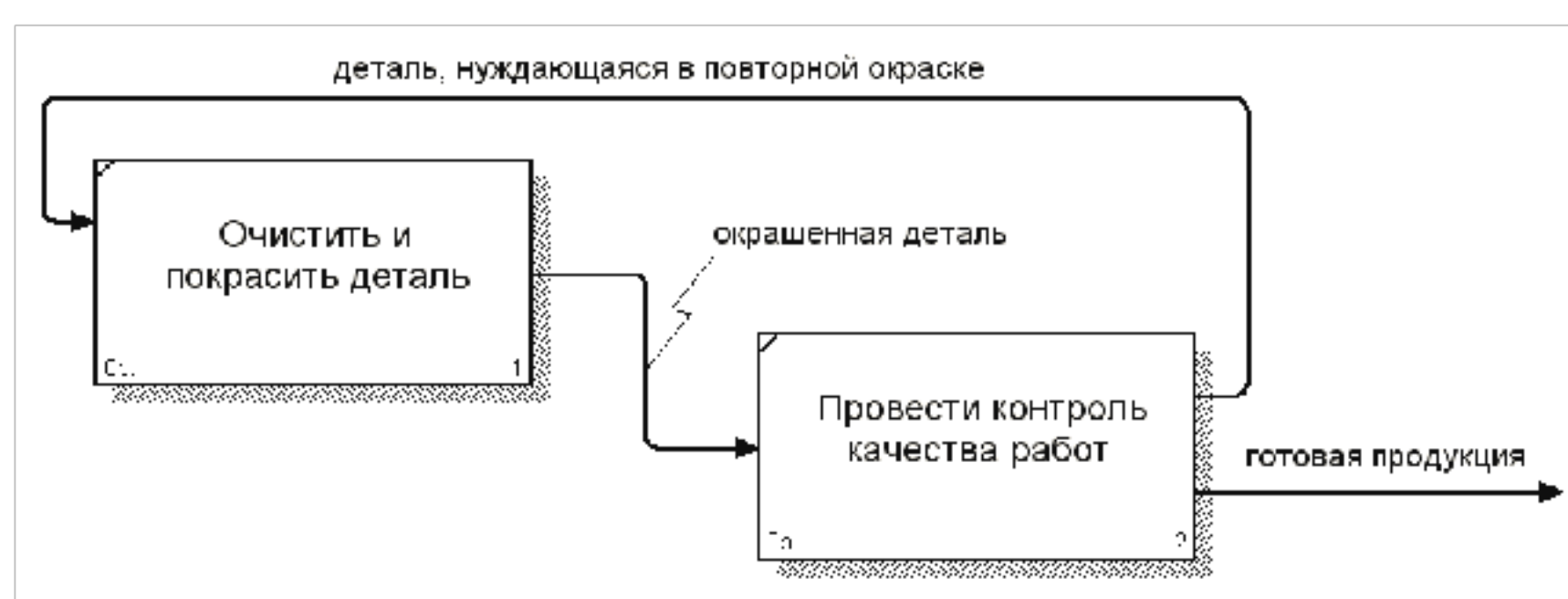


Рисунок 1.8 – Комбинированная стрелка *выход – обратная связь на вход*

Ветвление и слияние стрелок. Выход функционального блока может использоваться в нескольких других блоках. Фактически чуть ли не главная ценность IDEF0 заключается в том, что эта методология помогает выявить взаимозависимости между блоками системы. Соответственно IDEF0 предусматривает как ветвление, так и слияние стрелок на диаграмме. Разбитые на несколько частей стрелки могут иметь наименования, отличающиеся от наименования исходной стрелки. Исходная и разбитые (или объединенные) стрелки в совокупности называются связанными. Такая техника обычно применяется для того, чтобы отразить использование в процессе только части сырья или информации, обозначаемых исходной стрелкой.

1. Разработка проекта информационной системы «Служба занятости».
2. Разработка проекта информационной системы «Фондовый рынок».
3. Разработка проекта информационной системы «Кредитный эксперт».
4. Разработка программы учета клиентов для проекта информационной системы класса CRM.
5. Разработка программы оперативного учета поступления товара на склад.
6. Разработка проекта информационной системы «Недвижимость».
7. Разработка проекта информационной системы «Каталог товаров».
8. Разработка проекта информационной системы «Зарплата».
9. Разработка проекта информационной системы «Поставщики предприятия».
10. Разработка проекта информационной системы инвентаризации аппаратных средств.
11. Разработка проекта информационной системы учета программных средств предприятия.
12. Разработка проекта информационной системы тестирования знаний.
13. Разработка проекта информационной системы «Абитуриент».
14. Разработка проекта информационной системы статистической обработки данных.
15. Разработка проекта информационной системы «Библиотека».
16. Разработка программы учета данных для проекта информационной системы «Реклама».
17. Разработка проекта информационной системы учета успеваемости студентов.
18. Разработка проекта информационной системы «Выпускник вуза».
19. Разработка проекта информационной системы «Отдел кадров».
20. Разработка программы оперативного учета продаж торгового предприятия.

Вопросы для обсуждения:

1. Назначение функциональной модели IDEF0 при разработке проекта.
2. Анализ функции в рассматриваемой предметной области.
3. Модель потоков данных.
4. Анализ данных в предметной области.
5. Взаимосвязь модели потоков данных и функциональной модели.
6. Модель процессов IDEF3 назначение и взаимосвязь с моделями DFD и IDEF0.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: отчет к лабораторной работе.

Тема 2. Методы объектно-ориентированного и функционального моделирования информационных систем

Лабораторная работа №2. Разработка диаграмм бизнес-процессов.

Цель и содержание работы: изучить возможности программы ArgoUML и ее установку.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Возможности ARGUML

Теоретическое обоснование

ДОКУМЕНТ ПОДПИСАН
Электронно-цифровой подписью

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Класс (class) - категория вещей, которые имеют общие атрибуты и операции.

Классы - это строительные блоки любой объектно-ориентированной системы. Они представляют собой описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. При проектировании объектно-ориентированных систем диаграммы классов обязательны.

Классы используются в процессе анализа предметной области для составления *словаря предметной области* разрабатываемой системы. Это могут быть как абстрактные понятия предметной области, так и классы, на которые опирается разработка и которые описывают программные или аппаратные сущности.

Диаграмма классов - это набор статических, декларативных элементов модели. Диаграммы классов могут применяться и при прямом проектировании, то есть в процессе разработки новой системы, и при *обратном проектировании* - описании существующих и используемых систем. Информация с диаграммы классов напрямую отображается в исходный код приложения - в большинстве существующих инструментов UML-моделирования возможна *кодогенерация* для определенного языка программирования (обычно Java или C++). Таким образом, диаграмма классов - конечный результат проектирования и отправная точка процесса разработки.

Но мы опять заговорились, а, как известно, лучше один раз увидеть, чем сто раз услышать. Мы уже знаем, как классы обозначаются в UML, но пока еще не видели ни одной диаграммы "с их участием". Итак, посмотрим на примеры диаграмм классов.

Первый пример (рис. 2.1) весьма прост. Как видим, он, хоть и немного однобоко, иллюстрирует с помощью операции наследования или генерализации "генеалогическое древо" бытовой техники. Думаем, мы бы поняли смысл этой диаграммы, даже если бы ничего не знали о классах и не занимались программированием вообще.

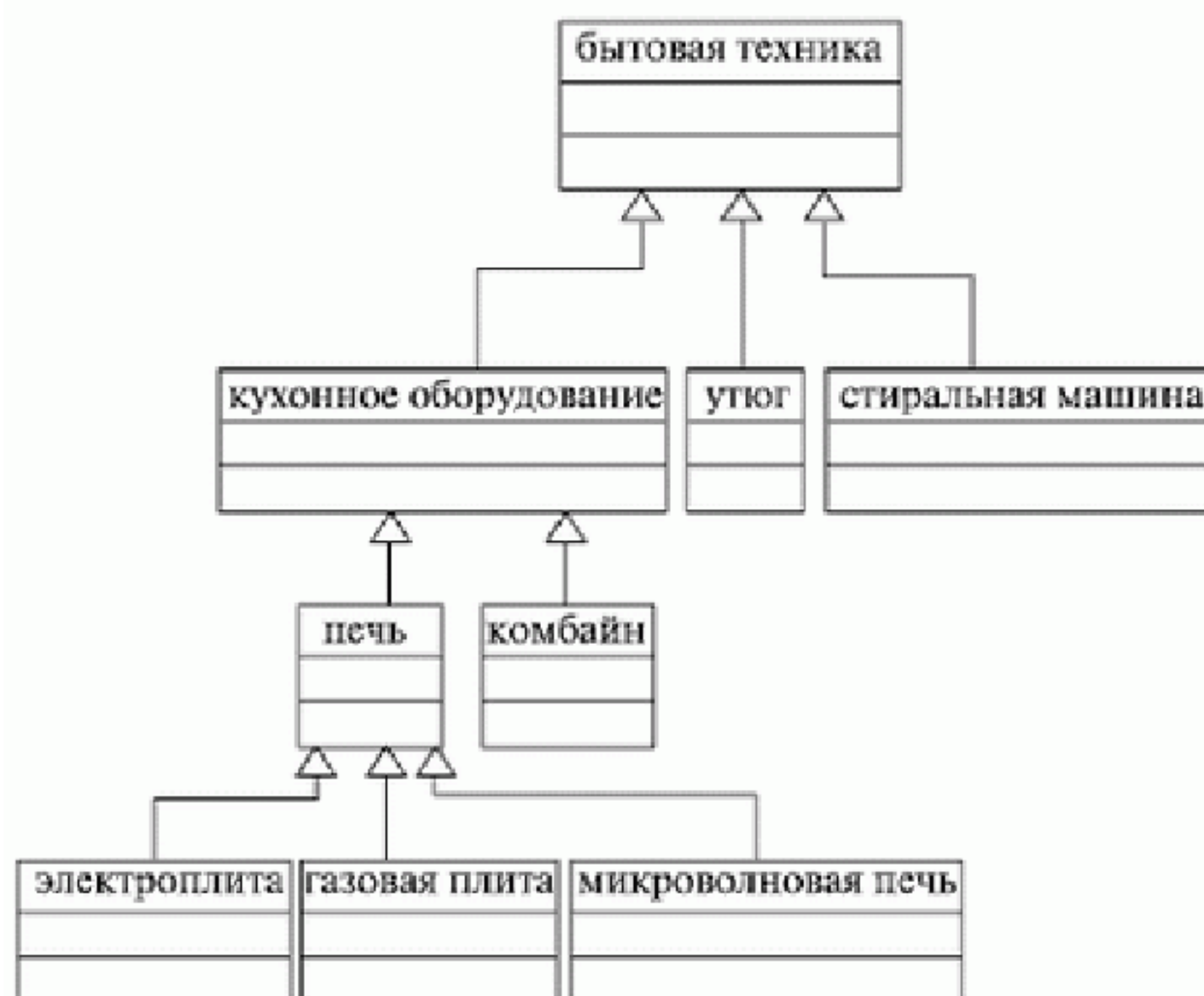


Рис. 2.1.

Рассмотрим еще пример:

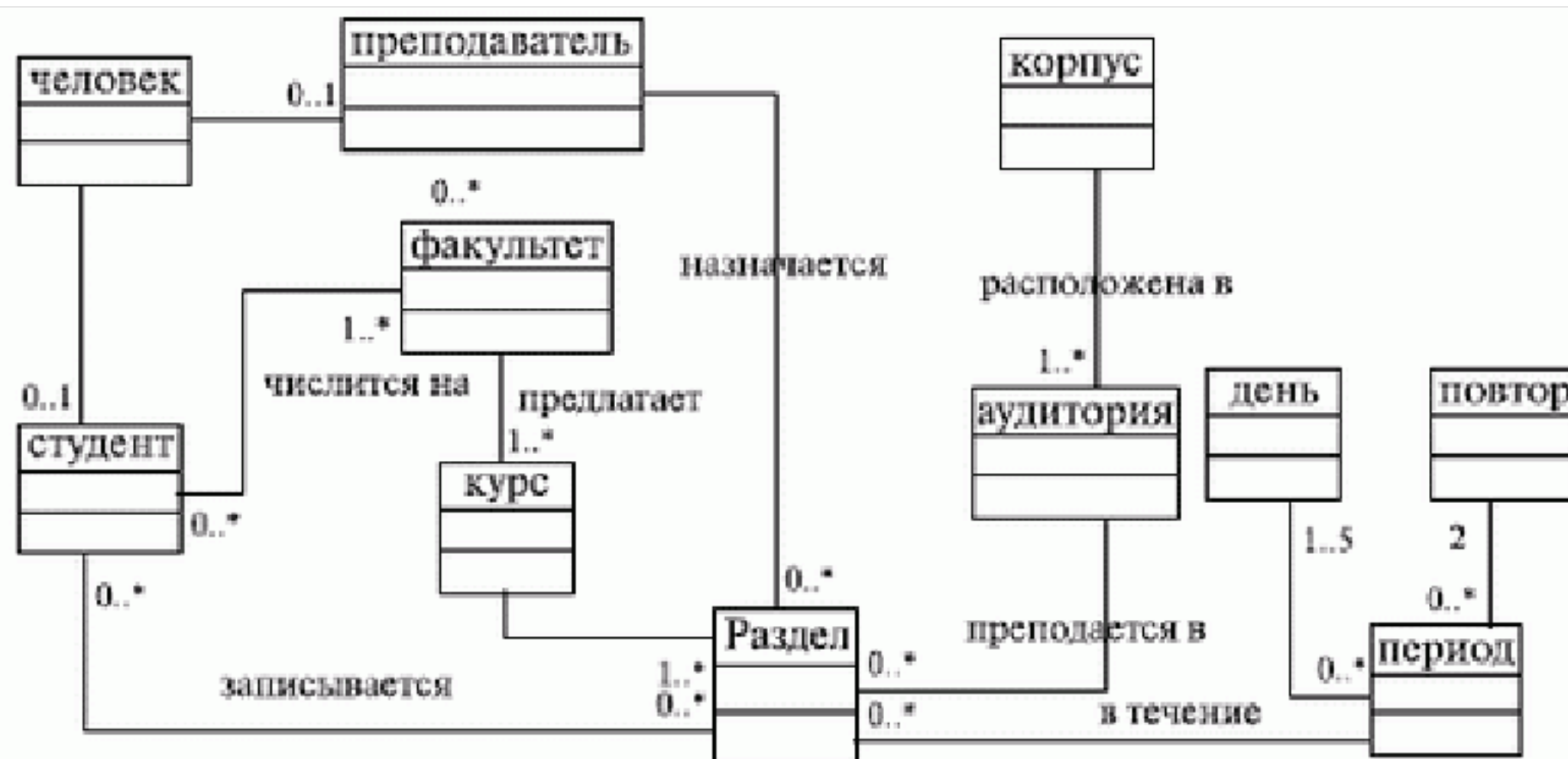


Рис. 2.2.

И опять-таки смысл этой диаграммы ясен без особых пояснений. Даже бегло рассмотрев ее, можно легко догадаться, что она описывает предметную область задачи об автоматизации работы некоего вуза или учебного центра. Обратите внимание на обозначения кратности на концах связей. А теперь немного усложним задачу (рис. 2.7):



Рис. 2.3.

Как видим, здесь уже все более серьезно - кроме кратности обозначены свойства (и их типы) и операции, и вообще, эта диаграмма производит впечатление набора классов для реализации, а не просто описания предметной области, как предыдущие. Но, тем не менее, все равно все просто и понятно.

Отметим, что более детально о диаграмме классов мы поговорим в следующей лекции. Там мы подробно разберем нотацию этого вида диаграмм и познакомимся с улучшениями, внесенными текущей версией UML.

Диаграммы прецедентов составляют модель прецедентов (вариантов использования, use-cases). Прецедент - это функциональность системы, позволяющая пользователю получить некий значимый для него, ощутимый и измеримый результат. Каждый прецедент соответствует отдельному сервису, предоставляемому моделируемой системой в ответ на запрос пользователя, т. е. определяет способ использования этой

системы. Именно *по* этой причине use cases, или прецеденты, часто в русской терминологии фигурируют как *варианты использования*. Варианты использования чаще всего применяются для спецификации внешних требований к проектируемой системе или для спецификации функционального поведения уже существующей системы. Кроме этого, варианты использования неявно описывают типичные способы взаимодействия пользователя с системой, позволяющие корректно работать с предоставляемыми системой сервисами.

Нефункциональные требования - это описание таких свойств системы, как особенности среды и реализации, *производительность, расширяемость, надежность* и т. д. Часто нефункциональные требования не привязаны к конкретному варианту использования и потому выносятся в отдельный *список* дополнительных требований к системе (рис. 6.2).

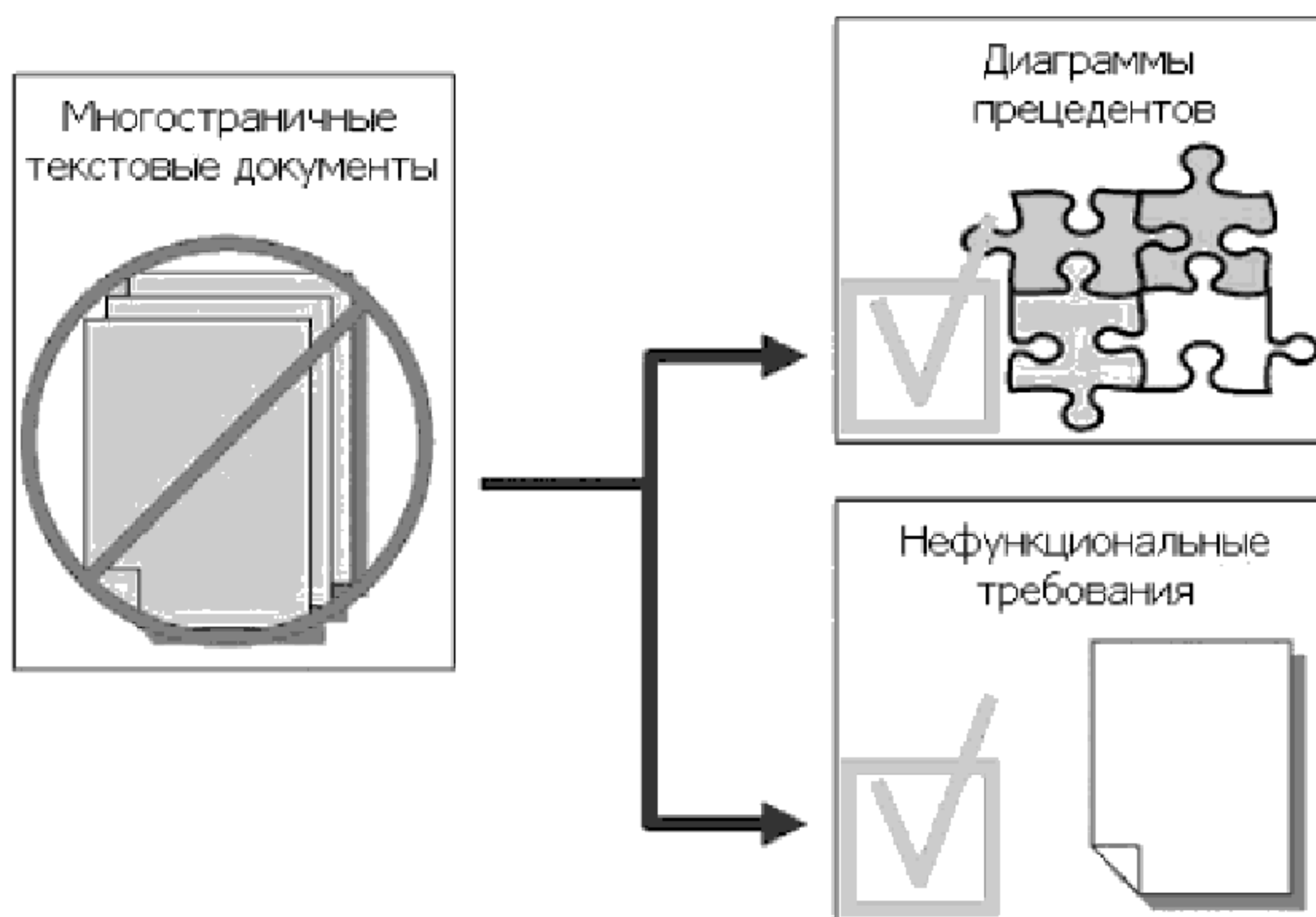


Рис. 2.4.

Но вернемся же к прецедентам (вариантам использования). Идентифицировать прецеденты и действующие лица - обязанность системного аналитика. И делает он это для того, чтобы:

- четко разграничить систему и ее окружение;
- определить, какие действующие лица и как именно взаимодействуют с системой, какой функционал (варианты использования) ожидается от системы;
- определить и описать в словаре предметной области (гlossарии) общие понятия, которые необходимы для детального описания функционала системы (прецедентов).

Подобный вид деятельности обычно выполняется в такой последовательности:

1. Определение действующих лиц.
2. Определение прецедентов.
3. Составление описания каждого прецедента.

4. Описание модели прецедентов в целом (этот этап включает в себя создание словаря предметной области).

Вначале требования оформляются в виде обычного текстового документа, который создается или самим пользователем, или пользователем и разработчиком вместе. Далее

требования оформляют в виде таблицы. В левую колонку помещают прецеденты, а в правую - действующих лиц, участвующих в прецеденте.

Рассмотрим пример. Секретарь размещает на сервере *меню* обеденных блюд на неделю. Сотрудники должны иметь возможность ознакомиться с *меню* и сделать заказ, выбрав блюда на каждый день следующей недели. *Офис-менеджер* должен иметь возможность сформировать счет и оплатить его. Система должна быть написана на *ASP.NET*. Такое вот нехитрое интернет-*приложение* для автоматизации заказов обедов в *офис*.

Думаем, здесь все понятно. *Таблица* с описанием требований может быть, например, такой:

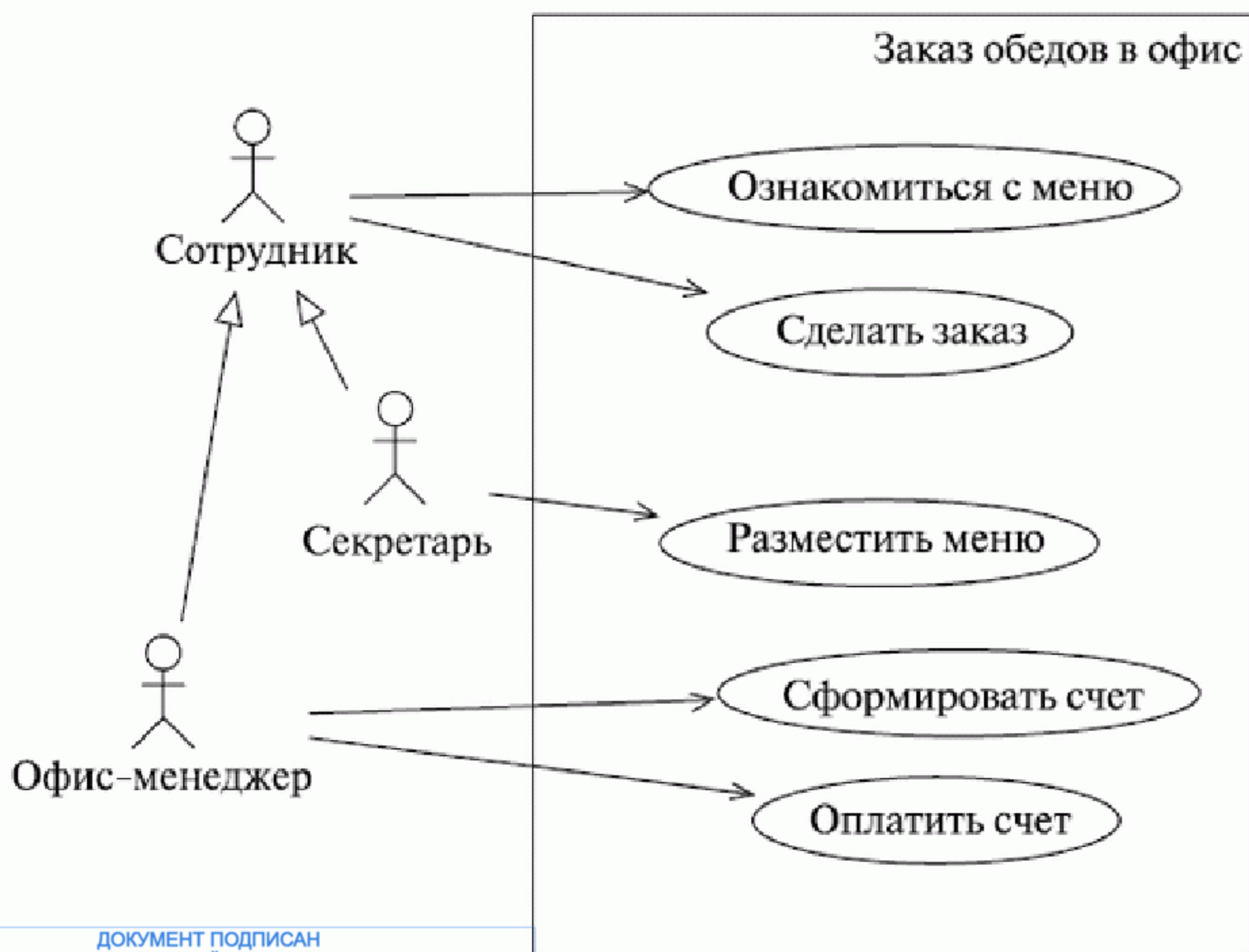
Прецедент

разместить меню
ознакомиться с меню
сделать заказ
сформировать счет
оплатить счет

Действующее лицо

секретарь
сотрудник, секретарь, офис-менеджер
сотрудник, секретарь, офис-менеджер
офис-менеджер
офис-менеджер

Здесь нигде не сказано о том, что система должна быть написана на *ASP.NET*. Почему - понятно: это ведь нефункциональное требование! И еще, очевидно, что секретарь и *офис-менеджер* тоже являются сотрудниками. Читатель, внимательно прочитавший предыдущие лекции, заподозрит, что в данном случае, создавая модель прецедентов, говоря о действующих лицах, можно бы применить генерализацию. Действительно, *диаграмма прецедентов*, построенная на основе этой таблицы, может быть, например, такой (рис. 2.5):



ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Первое, что бросается в глаза, - большой *прямоугольник*, внутри которого размещаются эллипсы, обозначающие, как мы уже поняли, прецеденты. В верхней части

прямоугольника указано название моделируемой системы, а называют его *рамками системы* (*system boundary, subject boundary*), *контекстом* или просто *системой*. Этот элемент диаграммы показывает границу между тем, что вы как *аналитик* показали в виде прецедентов (внутри этих рамок), и тем, что вы изобразили как действующие лица (вне их). Чаще всего таким прямоугольником показывают *границы самой моделируемой системы*. То есть внутри границы находятся прецеденты - тот функционал, который реализует система (и в этом смысле прецеденты могут рассматриваться как представления подсистем и классов модели), а снаружи - *действующие лица*: пользователи и другие *внешние сущности*, взаимодействующие с моделируемой системой.

Следует сказать, что рамки системы на диаграммах прецедентов изображают довольно редко, т. к. они неявно подразумеваются самой диаграммой. По сути, этот элемент не привносит в диаграмму какой-либо дополнительной значимой информации, так что его использование - дело вкуса аналитика. Появление рамок системы на диаграмме прецедентов чаще всего диктуется особенностями персонального стиля проектирования.

Кроме рамок системы или ее контекста на диаграмме мы видим еще два вида связанных с ней сущностей - это *действующие лица* (экторы, actors) и *прецеденты*. Начнем с экторов. Довольно часто в русскоязычной литературе *по UML* для обозначения действующих лиц можно встретить термин "*актер*". В принципе, смысл его более-менее понятен и оригинальному английскому термину он созвучен. Более того, есть еще одна причина такого перевода. Какое слово первым приходит к вам в голову, когда вы слышите слово "*актер*"? Да, конечно же - слово "*роль*"! Именно о ролях мы вскоре и поговорим, когда будем пытаться разобраться, что скрывается за понятием "*действующее лицо*". А пока, да простит нас читатель, далее мы все же будем пользоваться словом "*эктор*" - транскрипцией оригинального термина. Помнится, мы уже как-то писали о нашем отношении к буквальному переводу терминологии...

Итак, какой же смысл вкладывают в понятие эктора? *Эктор* - это набор ролей, которые исполняет *пользователь* в ходе взаимодействия с некоторой сущностью (системой, подсистемой, классом). Эктор может быть человеком, другой системой, подсистемой или классом, которые представляют нечто за пределами рассматриваемой сущности. Экторы "*общаются*" с системой путем обмена сообщениями. Четко выделив экторов, вы тем самым ясно определяете границу между тем, что внутри системы, и тем, что снаружи, - рамки системы.

Возможно, слова "*роли, исполняемые пользователем*" в определении эктора звучат не очень понятно. Очень забавно это понятие объясняется в Zicom Mentor:

роль - это не конкретный пользователь, а подобие шляпы, которую человек надевает, когда взаимодействует с сущностью.

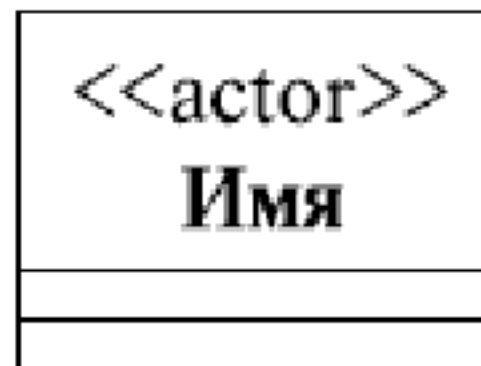
Действительно, наденьте шляпу пирата - и вы капитан Джек Воробей, а наденьте цилиндр и вы - Джек-потрошитель! Шутка... "*Физический*" *пользователь* может играть роль одного или даже нескольких экторов, выполняя их функции в ходе взаимодействия с системой. И наоборот, роль одного и того же эктора может выполняться несколькими пользователями.

На диаграммах *UML* экторы изображаются в виде стилизованных человечков, ведь, как вы, конечно, помните, идея была в создании нотации, любой символ которой легко может быть изображен от руки (рис. 2.6):



Рис. 2.6.

Несмотря на "человеческий" вид этого обозначения, не следует забывать, что экторы - это не обязательно люди. Эктором, как мы уже говорили ранее, может быть внешняя система, подсистема, *класс* и т. д. Кстати, человечек ("*stick-person*") - это не единственное обозначение эктора, используемое в *UML*. На диаграммах прецедентов обычно применяется именно "человекоподобная" форма эктора, но на других диаграммах, и особенно в случаях, когда эктор имеет атрибуты, которые важно показать, используется изображение эктора как класса со стереотипом `<<actor>>` :

**Рис. 2.7.**

С системой экторы, как мы уже сказали, общаются через сообщения, но если говорить на более высоком уровне абстракции, в терминах модели прецедентов, то взаимодействуют они с системой через прецеденты. Один и тот же эктор может быть связан с несколькими прецедентами, и наоборот, один *прецедент* может быть связан с несколькими разными экторами. Ассоциации между эктором и прецедентом всегда бинарные - т. е. представляют отношения типа "один к одному", использование кратности недопустимо. Это не противоречит сказанному выше: действительно, один эктор может быть связан с несколькими прецедентами, но только с помощью отдельных ассоциаций - по одной на каждый *прецедент*. Мы видели это в нашем примере. Кстати, там мы видели ассоциации, изображенные не просто в виде линий, а стрелками. Думаем, смысл этого обозначения вполне понятен: это *направленная ассоциация* и стрелка (как и на других диаграммах) всегда направлена в сторону той сущности, от которой что-то требуют, чьим сервисом пользуются и т. д.

И еще - экторы не могут быть связаны друг с другом. Единственное допустимое *отношение* между экторами - *генерализация* (*наследование*). Опять-таки, в нашем примере с заказом обедов в *офис*, вы могли увидеть именно такой вид отношений между экторами. Это не значит, что в реальной жизни *офис*-менеджер и секретарь (да и вообще любые два сотрудника) не могут общаться: просто при создании модели прецедентов такое общение не попадает в область наших интересов, считается несущественным.

Еще один тип элементов, встречающийся на диаграммах прецедентов, более того, давший им название, - это собственно *прецеденты*, или варианты использования. *Прецедент* - это описание набора последовательных событий (включая возможные варианты), выполняемых системой, которые приводят к наблюдаемому эктором результату. Прецеденты описывают сервисы, предоставляемые системой экторам, с которыми она взаимодействует. Причем *прецедент* никогда не объясняет, "как" работает сервис, а только описывает, "что" делается.

Изображаются прецеденты в виде эллипса, внутрь контура которого помещается имя (описание) прецедента. Имя прецедента обычно намного длиннее имен других элементов модели. Почему это так, в принципе, понятно: имя прецедента описывает взаимодействие эктора с системой, говорит о том, какими сообщениями они обмениваются между собой. В нашем примере с заказом обедов мы видели несколько прецедентов, и наверняка читатель заметил, что имя прецедента - это, скорее, название сценария, воспроизводимого в ходе взаимодействия эктора с системой. Причем это всегда описание с точки зрения эктора, описание услуг, предоставляемых системой

Сценарии также иногда можно увидеть на диаграмме прецедентов. Иногда их изображают в виде "листа бумаги", на котором написано *имя файла*, - прямоугольника с загнутым нижним левым углом. В этом случае указанный *файл* содержит в себе описание данного сценария. А иногда *сценарий* записывается в комментарий. Как вы, наверное, помните, комментарии (ноутсы, notes) изображаются прямоугольниками с загнутым верхним правым углом и соединяются с элементом, который они поясняют, пунктирной линией.



Рис. 2.9.

Как мы уже упоминали, сценарии могут быть записаны в различных формах. Это может быть структурированный, но неформализованный текст, формализованный структурированный текст, *псевдокод*, *таблица*, *диаграмма активностей*, наконец! Каждый *сценарий* описывает в повествовательной форме завершённое, конкретное взаимодействие, имеющее с точки зрения пользователя определённую цель. Если рассматривать табличную форму представления сценария, то линия, разделяющая левый и правый столбцы таблицы, символизируют собой границу, отделяющую действия пользователя от ответных действий системы. Табличная форма особо подчеркивает участие пользователя, что является очень важным аспектом при разработке пользовательского интерфейса.

Вот пример простого (неформализованного) текстового описания сценария.

Пользователь вводит логин, пароль, адрес электронной почты и код подтверждения и нажимает кнопку "Далее". Система запрашивает ввод проверочного кода. Пользователь вводит код и нажимает кнопку "Далее". Система проверяет соответствие кода изображенному на картинке.

Не правда ли, знакомая процедура? Да, это описание регистрации пользователя на некотором сайте. Правда, не совсем полное: не рассмотрены случаи, когда выбранный пользователем логин уже занят, *адрес* электронной почты введен неправильно, *пароль* не удовлетворяет требованиям или код не соответствует изображенному на картинке. О таких случаях - *альтернативных сценариях* - мы поговорим чуть позже.

А вот тот же *сценарий* в табличном представлении:

Действия пользователя	Реакция системы
Ввод логина, пароля, адреса электронной почты и нажатие кнопки "Далее"	Запрос ввода проверочного кода
Ввод проверочного кода и нажатие кнопки "Далее"	Проверка кода на соответствие изображенному на картинке

Вы, конечно, заметили, что этот *сценарий* можно детализировать - например, прежде чем попросить ввести проверочный код, система отображает картинку, на которой этот самый код изображен. Т. е. *запрос* на ввод кода *включает в себя вывод* картинки с упомянутым кодом. Об этом мы тоже еще поговорим.

А пока попробуем ответить на второй вопрос, а именно: *как связаны понятия сценария и прецедента*. Прецеденты, как мы уже говорили, рождаются из требований к

системе. Но говорят они о том, что делает система. Как система это делает, говорят сценарии. Таким образом, *прецедент* можно специфицировать путем описания потока действий или событий в текстовой форме - в виде, понятном для "постороннего" (не занятого в непосредственной разработке системы) читателя. А ведь такое описание - это и есть *сценарий*! Таким образом, *сценарии специфицируют прецеденты*. И еще. Поскольку сценарии - это, по сути, рассказы, они являются весьма эффективным средством извлечения информации из бесед с заказчиком и предоставляют превосходное, понятное непрофессионалу описание создаваемого приложения. Сценарии, да и вообще *диаграммы прецедентов* (дополненные сценариями) являются отличным *средством общения между разработчиками и заказчиком*, причем, в силу простоты нотации, - средством, понятным обеим сторонам. В конечном итоге, взаимосвязь между требованиями, прецедентами и сценариями можно изобразить такой "псевдодиаграммой".

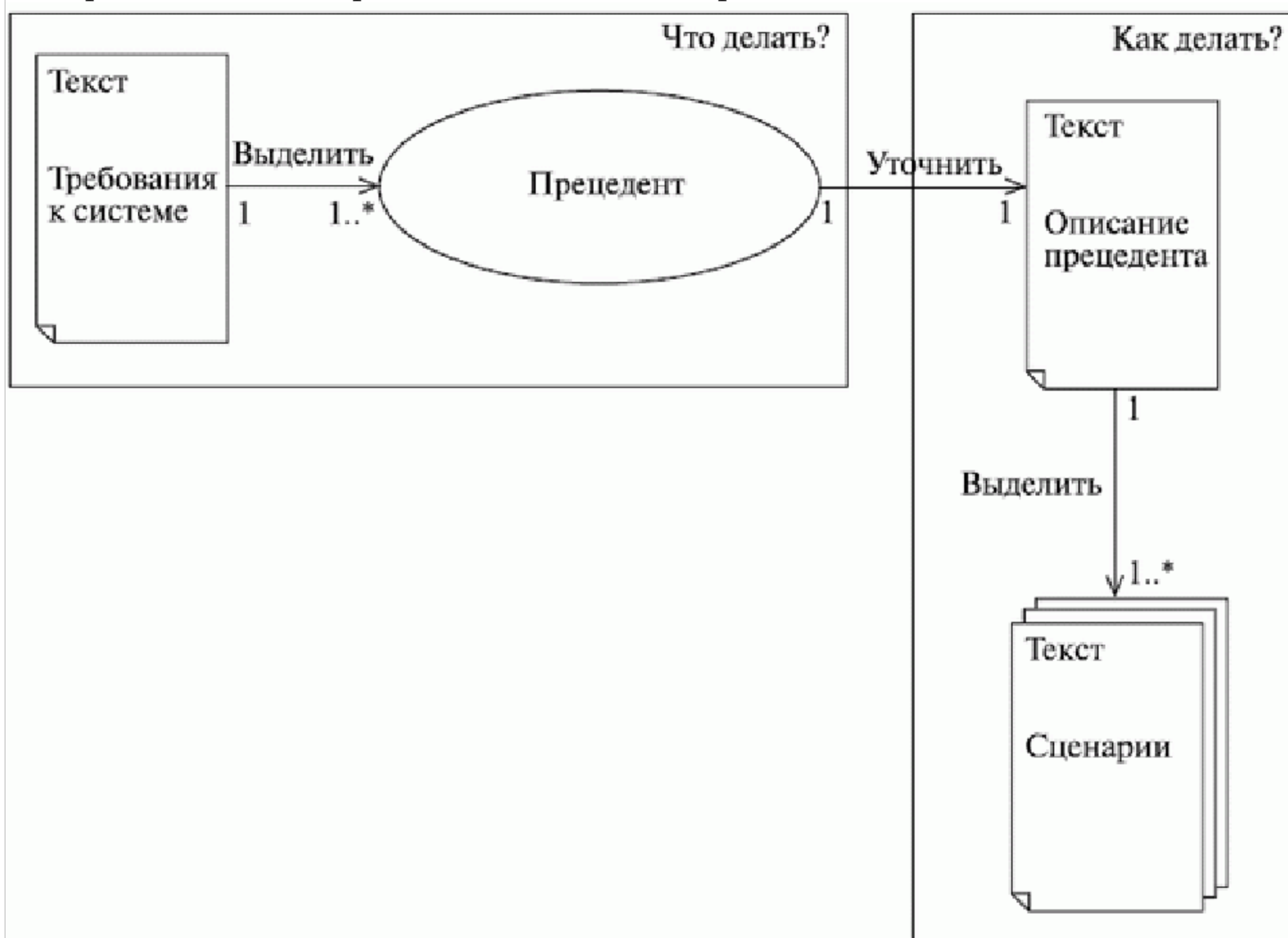


Рис. 2.10.

Как видите, для каждой ассоциации на диаграмме проставлена *кратность* и ее смысл вполне понятен, но все же о кратности следует поговорить отдельно. Один *прецедент* определяет несколько сценариев, каждый из которых представляет один из возможных вариантов определяемого прецедентом потока событий. Сценарии так же соотносятся с прецедентами, как экземпляры класса, т.е. *сценарий - это экземпляр прецедента*, как *объект - экземпляр класса*. Система может содержать, например, несколько десятков прецедентов, каждый из которых, в свою очередь, может разворачиваться в десятки сценариев. Как правило, *прецедент* описывает не одну последовательность действий, а множество, и выразить все детали рассматриваемого прецедента с помощью одной последовательности действий обычно не получается. Практически для любого прецедента можно выделить основной *сценарий*, описывающий "нормальную" последовательность действия, и вспомогательные,

описывающие *альтернативные* последовательности, которые инициируются в случае возникновения определенных условий.

Другой вопрос: требуется ли такое уточнение модели прецедентов, оправдано ли оно для данного уровня приближения, или "подразумевающиеся" *альтернативные* сценарии можно опустить? Например, в предыдущем примере с покупкой билета в сервисной кассе мы не изобразили сценарии (и, соответственно, прецеденты), соответствующие вариантам, когда билетов на выбранный пассажиром рейс уже не осталось, пассажир изменил свое решение и хочет взять билет на другой рейс, когда *оплата* идет наличными или *по* кредитной карте и т. д.

"Хватит ходить вокруг да около!" - воскликнет нетерпеливый читатель. Уже заканчиваем. Мы просто хотели мягко подвести читателя к вопросу об отношениях между прецедентами. А отношения эти весьма многообразны. Начнем со старого знакомого - отношения обобщения (наследования, генерализации). О генерализации мы уже говорили не раз, когда рассматривали *диаграммы классов*. Но все же напомним суть этого понятия. Как говорят классики, *обобщение* - это *отношение* специализации (обобщения), в котором объекты специализированного элемента (потомка) могут быть подставлены вместо объектов обобщенного элемента (родителя, или предка).

Точно так же, как мы обычно поступаем с классами, после того как мы выделили и описали каждый *прецедент*, мы должны просмотреть их все на предмет наличия одинаковых действий - поискать, а не выполняются ли (используются) некоторые действия совместно несколькими вариантами использования. Этот совместно используемый фрагмент лучше описать в отдельном прецеденте. Таким образом мы уменьшим *избыточность* модели за счет применения обобщения прецедентов (иногда, правда, говорят не об обобщении, а об *использовании* прецедентов; почему - сейчас поймете). Как это и "положено" при наследовании, экземпляры обобщенных прецедентов (потомков) сохраняют поведение, присущее обобщающему прецеденту (предку). Другими словами, наличие (использование) в варианте использования *X* обобщенного варианта использования *Y* говорит нам о том, что экземпляр прецедента *X* *включает в себя* поведение прецедента *Y*. Обобщения применяются, чтобы упростить понимание модели вариантов использования за счет многократного задействования "заготовок" прецедентов для создания прецедентов, необходимых заказчику (помните, как мы рассматривали вопрос о том, всегда ли необходимо создавать новый *класс*, или лучше воспользоваться готовым решением, чувствуете аналогию?). Такие "полные" прецеденты называются *конкретными прецедентами*. "Заготовки" прецедентов, созданные лишь для многократного использования в других прецедентах, называют абстрактными прецедентами. Абстрактный *прецедент* (как и *абстрактный класс*) не существует сам *по себе*, но экземпляр конкретного прецедента демонстрирует поведение, описываемое абстрактными прецедентами, которые он (повторно) использует. *Прецедент*, который экторы наблюдают при взаимодействии с системой ("полный" *прецедент*, как мы называли его ранее), часто называют еще "*реальным*" прецедентом.

Как мы уже говорили выше, *обобщение (наследование)* чаще всего используют между классами и интерфейсами. Однако другие элементы модели также могут находиться между собой в отношении наследования - например, пакеты (о которых мы тут не говорим), экторы, прецеденты...

Изображается *обобщение*, как, конечно, помнит внимательный читатель, линией с "незакрашенной" треугольной стрелкой на конце. *Обобщение* - это *отношение* между предком и потомком, и стрелка всегда указывает на предка. Если вспомнить, что потомки наследуют (используют) свойства предка, то вполне логично вспоминается наше утверждение о том, что стрелки в *UML* всегда направлены в сторону того, от кого что-то требует, чьи сервисами пользуются.

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

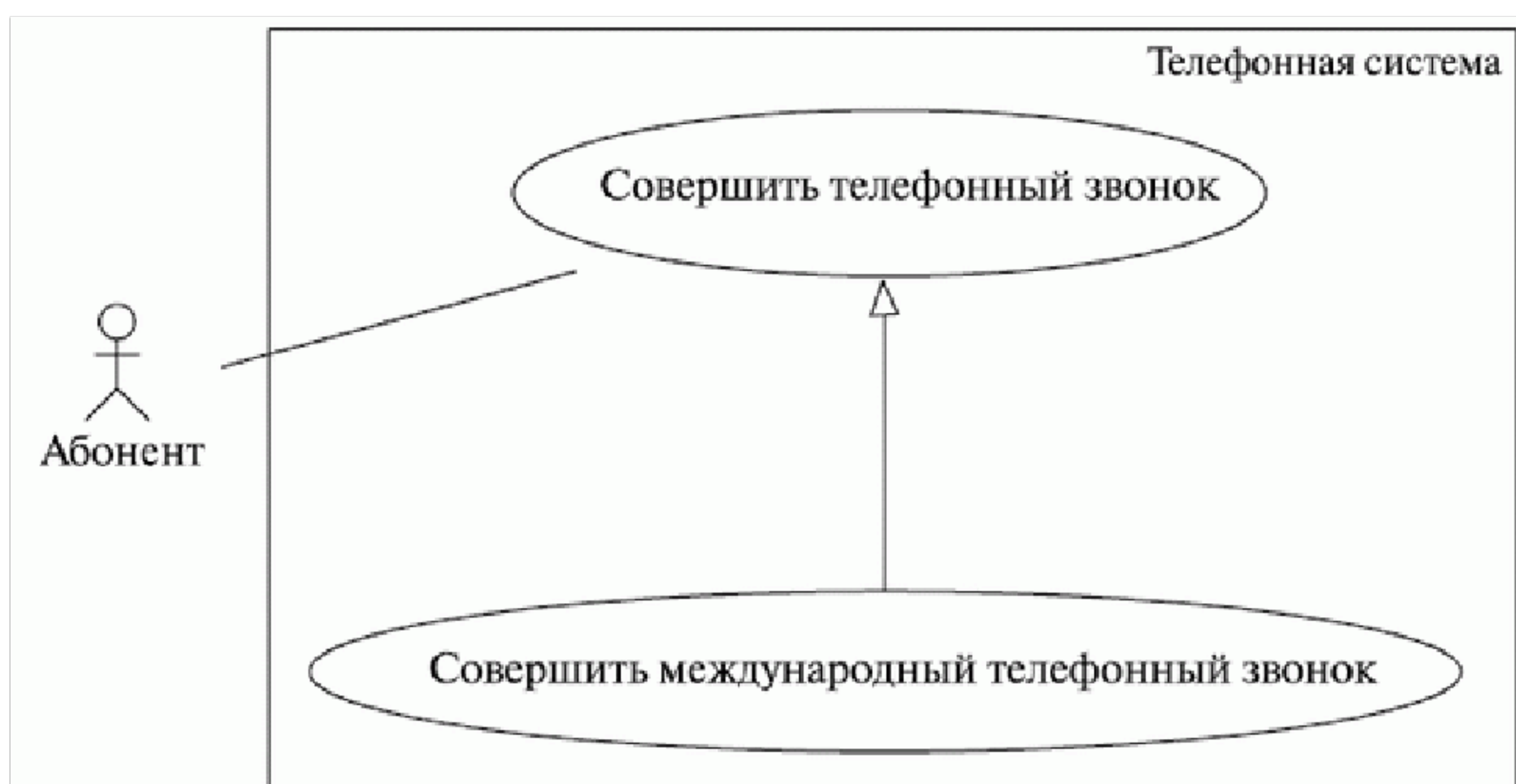


Рис. 2.10.

Как мы уже говорили ранее и видели в нашем первом примере *диаграммы прецедентов*, обобщение может использоваться для создания различных разновидностей экторов. Экторы-потомки наследуют от предка базовые характеристики и дополняют их своей спецификой. Точно так же *прецедент-потомок* наследует поведение и семантику прецедента-родителя и дополняет его поведение.

Следующий вид отношений между прецедентами - включение. *Отношение* включения означает, что *в некоторой точке базового прецедента содержится поведение другого прецедента*. Включаемый прецедент не существует сам по себе, а является всего лишь частью объемлющего прецедента. Таким образом, базовый прецедент как бы заимствует поведение включаемых, раскладываясь на более простые прецеденты. Например, когда мы покупаем в магазине некоторую вещь, в момент считывания кассиром штрих-кода обновляется состояние *базы данных* товаров, имеющих в наличии, - количество наличных единиц купленного товара уменьшается. То же самое действие выполняется и в том случае, если купленный *товар* оказался бракованным, непригодным к использованию или попросту нам не понравился: состояние упомянутой *базы данных* вновь обновляется - но теперь уже в сторону увеличения количества наличных единиц определенного товара. Т. е. оба этих действия - и покупка, и возврат - содержат (включают в себя) такое действие, как обновление содержимого *БД*.

А как же изображается включение? Да очень просто - как зависимость (пунктирная линия со стрелкой, помните?) со стереотипом `<<include>>`. При этом стрелка направлена, естественно, в сторону включаемого прецедента. Этот факт легко объяснить, если вспомнить утверждение, которое мы уже несколько раз использовали в этом курсе: стрелка всегда направлена в сторону того элемента, от которого что-то требуется, чьиими сервисами пользуются. А если считать, что объемлющий прецедент включает в себя, заимствует (использует) поведение включаемых прецедентов, становится ясно, что стрелка может быть направлена только таким образом. А вот и *диаграмма*, иллюстрирующая вышесказанное, которую мы позаимствовали из Zicom Mentor:

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

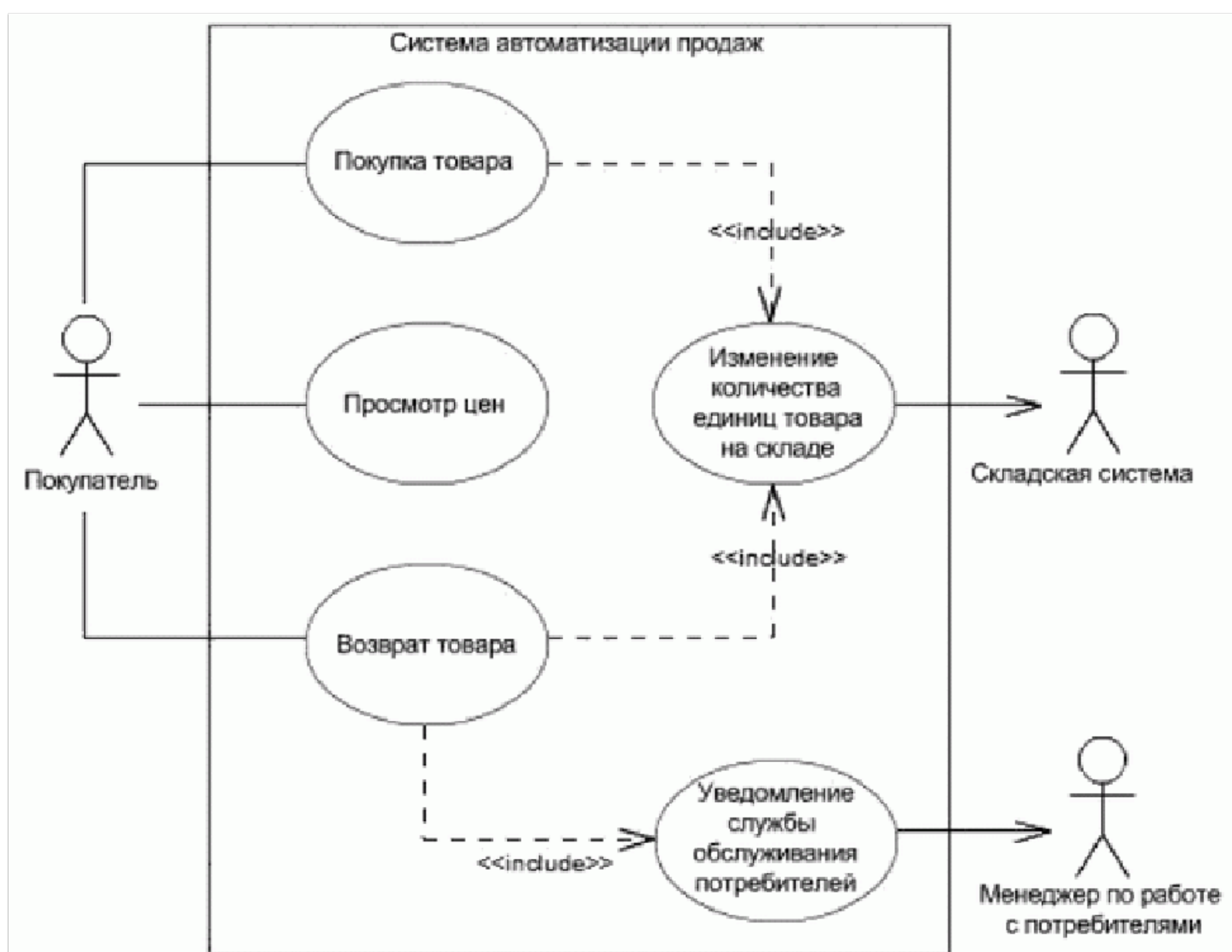


Рис. 2.11.

Как хорошо видно из этого примера, использование включения позволяет избежать многократного описания одного и того же набора действий - общее поведение можно просто описать в виде прецедента, включаемого в базовые.

Ход лабораторной работы:

Разработать диаграммы классов и диаграммы прецедентов для предметной области:

1. Разработка программы учета ресурсов для проекта информационной системы класса MRP.
2. Разработка программы обработки данных по производственной практике.
3. Разработка программы учета данных для проекта информационной системы «Турагентство».
4. Разработка проекта информационной системы «Санаторно-курортная деятельность».
5. Разработка программы вывода афиши для проекта информационной системы «Кинотеатр».
6. Разработка проекта информационной системы «Салон красоты».
7. Разработка программы ввода-вывода данных для проекта информационной системы «Аптека».
8. Разработка проекта информационной системы «Школа».
9. Разработка проекта информационной системы «Регистратура поликлиники».
10. Разработка проекта информационной системы «Гостиница».
11. Разработка проекта информационной системы «Авиабилеты».
12. Разработка проекта информационной системы «Кинотеатр».
13. Разработка проекта информационной системы «Книжный магазин».

Сертификат:
Владелец:

2C0000043E9AB8B952205E7BA500060000043E
Шаблон для электронной подписи

Действителен с 19.08.2022 по 19.08.2023

14. Разработка проекта информационной системы «Компьютерный магазин».

15. Разработка проекта информационной системы «Учет курсовых проектов».

Вопросы для обсуждения:

Диаграммы RationalRose назначения и краткие их характеристики. Диаграммы бизнес-процессов.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: отчет к лабораторной работе.

Тема 3. Предпроектное обследование предметной области. Согласование технического задания на разработку информационной системы

Лабораторная работа №3. Разработка технического задания на проект информационной системы.

Цель и содержание работы: Освоение принципов вербального описания предметной области. Приобретение практических навыков в создании вербальной модели предметной области.

Организационная форма занятий: решение задач предпроектного проектирования.

Вопросы для обсуждения на лабораторном занятии: Порядок вербального описания предметной области

Теоретическое обоснование

В иерархии формальных моделей вербальные модели занимают почетное место в основании этой «пирамиды». Такое положение действительно почетно, поскольку вербальные модели — это «альфа» и «омега» многоэтапного процесса моделирования — с этапа синтеза вербальной модели начинается процесс поэтапной формализации и вербальная же (в большинстве случаев) модель формируется на заключительном этапе функционирования модели. Это вызвано вполне понятными причинами — формализм вербальной модели легко воспринимается широким классом потребителей, а синтез вербальной модели (по крайней мере, в первом приближении) может быть осуществлен и специалистом, не обладающим специальными навыками в области построения формальных моделей. Благодаря тому, что языки естественного общения не ограничены рамками узкой предметной области, вербальные модели обладают наивысшей выразительной способностью и часто используются как инструмент интеграции формальных моделей и результатов их применения.

По существу, первичная вербальная модель представляет собой словесный портрет системы и проблемной ситуации, то есть представляет собой документ, аналогичный проекту технического (информационно-поискового и т. п.) задания, разрабатываемого некой организацией-заказчиком. Заметим: процесс синтеза первичной вербальной модели может осуществляться и при участии сторонних (приглашенных) специалистов. К этому шагу приходится прибегать в тех случаях, когда организация не располагает информацией, достаточной для принятия решения или выявления сущности противоречия. Заказчик не всегда в состоянии осознать суть проблем (например, проблем в области управления), с которыми он сталкивается. Находясь внутри системы, заказчик часто пребывает в состоянии информационной изоляции, лишен возможности наблюдать изменения, происшедшие в среде.

Часто на этапе синтеза вербальной модели применяются методы активизации интеллектуальной деятельности специалистов, методы извлечения экспертных знаний, призванные выявить неосознанные алгоритмические схемы функционирования отдельных сотрудников и организации в целом. Здесь могут проводиться в том числе и деловые игры, в ходе которых сторонний специалист пытается выявить алгоритмы функционирования системы, составить схему информационных процессов, информационных контуров управления.

Чтобы понять сущность вербального моделирования, разберемся, для чего создаются вербальные модели. Итак, вербальные модели создаются для:

- 1) получения на материальном носителе вербального описания:
 - структуры системы;
 - отношений между элементами;
 - функций системы и ее компонентов;
 - динамических параметров системы;
 - проблемной ситуации;
 - совокупности целей и задач деятельности;
 - разнообразных ограничений (в том числе — по ресурсам);
 - характеристик среды функционирования и возмущающих воздействий;
- 2) формирования массива исходных данных, используемых на этапе структурирования и формализации знаний о системе;
- 3) выявления специфики тезауруса, применяемого в данной предметной области (для внешнего эксперта), и упорядочения системы понятий, подлежащих выражению формальными средствами;
- 4) выявления неполноты системы знаний и организации процесса их пополнения как за счет внутренних ресурсов системы, так и с привлечением внешних информационных ресурсов;
- 5) установления характера неопределенностей, с которыми придется столкнуться на этапе синтеза формальной модели;
- 6) поиска базовых закономерностей и аналогий в смежных отраслях, которые могут быть использованы в дальнейшем.

Таким образом, вербальная модель создается для сокращения неопределенности, компенсации неполноты знаний и формирования гипотезы или набора гипотез. Но первая и главная задача вербального моделирования — **создание вербального описания на материальном носителе.**

Вербальная модель — это не обязательно исключительно текстовый документ — она может содержать в том числе и количественные характеристики, элементы структуризации (например, таблицы, схемы и графики).

В ходе дальнейшей формализации вербальная модель подвергается процедуре структурирования. На этом этапе **устанавливаются группы взаимосвязанных элементов системы** и с необходимой степенью детализации (для решения поставленной задачи) **описываются отношения между ними**, осуществляется **атрибуция элементов системы и данных о них** (устанавливается структура описаний, формулируются требования к точности и т. п.), а также производится группирование данных.

Важным этапом вербального моделирования является этап приведения (стандартизации) терминологии и сокращения избыточности описаний. Результатом выполнения этой процедуры является вербальная модель, построенная в едином стандартизованном тезаурусе, дальнейшее использование которой упрощает решение задач автоматизации процессов анализа и перевода модели на следующий уровень формального представления.

При решении задачи синтеза баз данных и систем информационного обеспечения деловых процессов, данных, полученных на этом этапе, зачастую оказывается достаточно для синтеза макета информационной системы.

По завершении этапа вербального моделирования системы/процесса, при условии, что логическая компонента модели была успешно выделена (не изъята, а именно выделена, маркирована или акцентирована), становится возможен переход на следующий уровень — **уровень логико-лингвистического моделирования**.

При выполнении лабораторной работы необходимо дать вербальное описание предметной области по заданию преподавателя с указанием:

- структуры системы;
- отношений между элементами;
- функций системы и ее компонентов;
- динамических параметров системы;
- проблемной ситуации;
- совокупности целей и задач деятельности;
- разнообразных ограничений (в том числе — по ресурсам);
- характеристик среды функционирования и возмущающих воздействий

Ход лабораторной работы:

Произвести предпроектное обследование заданной предметной области и разработать техническое задание на разработку информационной системы:

1. Разработка проекта информационной системы «Служба занятости».
2. Разработка проекта информационной системы «Фондовый рынок».
3. Разработка проекта информационной системы «Кредитный эксперт».
4. Разработка программы учета клиентов для проекта информационной системы класса CRM.
5. Разработка программы оперативного учета поступления товара на склад.
6. Разработка проекта информационной системы «Недвижимость».
7. Разработка проекта информационной системы «Каталог товаров».
8. Разработка проекта информационной системы «Зарплата».
9. Разработка проекта информационной системы «Поставщики предприятия».
10. Разработка проекта информационной системы инвентаризации аппаратных средств.
11. Разработка проекта информационной системы учета программных средств предприятия.
12. Разработка проекта информационной системы тестирования знаний.
13. Разработка проекта информационной системы «Абитуриент».
14. Разработка проекта информационной системы статистической обработки данных.
15. Разработка проекта информационной системы «Библиотека».
16. Разработка программы учета данных для проекта информационной системы «Реклама».
17. Разработка проекта информационной системы учета успеваемости студентов.
18. Разработка проекта информационной системы «Выпускник вуза».
19. Разработка проекта информационной системы «Отдел кадров».
20. Разработка программы оперативного учета продаж торгового предприятия.

Вопросы для обсуждения:

1. Техничко-экономический анализ предметной области.
2. Структура технического задания на разработку информационной системы.

Работа с литературой:

Рекомендуемые источники информации				
(№ источника)				
ДОКУМЕНТ ПОДПИСАН				
ЭЛЕКТРОННОЙ ПОДПИСЬЮ				
Сертификат:	Основная	Дополнительная	Методическая	Интернет-ресурсы
Владелец:	Шебзухова Татьяна Александровна			
	1-3	1-3	1-3	1-2

Оценочные средства: отчет к лабораторной работе.

Тема 4. Разработка прототипа информационной системы. Тестирование взаимодействия программного интерфейса с базой данных

Лабораторная работа № 4. Разработка прототипа информационной системы.

Цель и содержание работы: Приобретение практических навыков анализа предметной области, информационных задач и построения концептуальной модели базы данных..

Организационная форма занятий: решение задач разработки прототипа информационной системы.

Вопросы для обсуждения на лабораторном занятии: Цель процесса проектирования, техника проектирования.

Теоретическое обоснование

Проектирование базы данных (БД) – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). В результате её решения должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Основная цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

- корректность схемы БД, т.е. база должна быть гомоморфным образом моделируемой предметной области (ПО), где каждому объекту предметной области соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных;
- обеспечение ограничений ;
- эффективность функционирования ;
- защита данных (от аппаратных и программных сбоев и несанкционированного доступа);
- простота и удобство эксплуатации;
- гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

Техника проектирования баз данных может измениться в целом и в деталях в зависимости от назначения системы. Например, следует различать проектирование систем складирования данных и проектирование так называемых OLTP-систем, ориентированных на оперативную обработку транзакций. В данном учебном курсе рассматривается проектирование баз данных в основном для OLTP-систем. Именно на таких системах исторически сложилась техника проектирования баз данных.

Процесс проектирования включает в себя следующие этапы:

Концептуальное проектирование – это процедура конструирования информационной модели, не зависящей от каких-либо физических условий реализации.

Логическое проектирование – это процесс конструирования информационной модели на основе существующих моделей данных, не зависимо от используемой СУБД и других условий физической реализации.

Физическое проектирование – это процедура создания описания конкретной реализации БД с описанием структуры хранения данных, методов доступа к данным.

Концептуальное проектирование

Основными задачами концептуального проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО.

Концептуальная модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не отдельных объектов ПО и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое.

Рассмотрим основные подходы к созданию концептуальной модели предметной области.

Функциональный подход к проектированию БД

Этот метод реализует принцип "от задач" и применяется тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

Предметный подход к проектированию БД

Предметный подход к проектированию БД применяется в тех случаях, когда у разработчиков есть чёткое представление о самой ПО и о том, какую именно информацию они хотели бы хранить в БД, а структура запросов не определена или определена не полностью. Тогда основное внимание уделяется исследованию ПО и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

Проектирование с использованием метода "сущность-связь"

Метод "сущность-связь" (entity-relation, ER-method) является комбинацией двух предыдущих и обладает достоинствами обоих. Этап инфологического проектирования начинается с моделирования ПО. Проектировщик разбивает её на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое локальное представление моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов ПО. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Типы сущностей можно классифицировать как сильные и слабые. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки – сильная сущность, а абонемент этого читателя – слабая, которая зависит от наличия соответствующего читателя. Слабые сущности называют подчинёнными (дочерними), а сильные – базовыми (основными, родительскими).

Для каждой сущности выбираются свойства (атрибуты). Различают:

- Идентифицирующие и описательные атрибуты. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). В качестве ПК обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. Кроме того, ПК должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и заключают в себе интересующие свойства сущности.

- Составные и простые атрибуты. Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией

нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.

- Однозначные и многозначные атрибуты (могут иметь соответственно одно или много значений для каждого экземпляра сущности).
- Основные и производные атрибуты. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).

Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности – множества значений (или домена), которые может принимать данный атрибут.

Далее осуществляется спецификация связей внутри локального представления. Связи могут иметь различный содержательный смысл (семантику). Различают связи типа "сущность-сущность", "сущность-атрибут" и "атрибут-атрибут" для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и ту же связь типа "сущность-сущность".

Каждая связь характеризуется именем, обязательностью, типом и степенью. Различают факультативные и обязательные связи. Если вновь порождённый объект одного типа оказывается по необходимости связанным с объектом другого типа, то между этими типами объектов существует обязательная связь (обозначается двойной линией). Иначе связь является факультативной.

По типу различают множественные связи "один к одному" (1:1), "один ко многим" (1:N) и "многие ко многим" (M:N).

Степень связи определяется количеством сущностей, которые охвачены данной связью. Пример бинарной связи – связь между отделом и сотрудниками, которые в нём работают. Примером тернарной связи является связь типа экзамен между сущностями ДИСЦИПЛИНА, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ. Из последнего примера видно, что связь также может иметь атрибуты (в данном случае это Дата проведения и Оценка). Пример ER-диаграммы с указанием сущностей, их атрибутов и связей приведен на рис. 1.

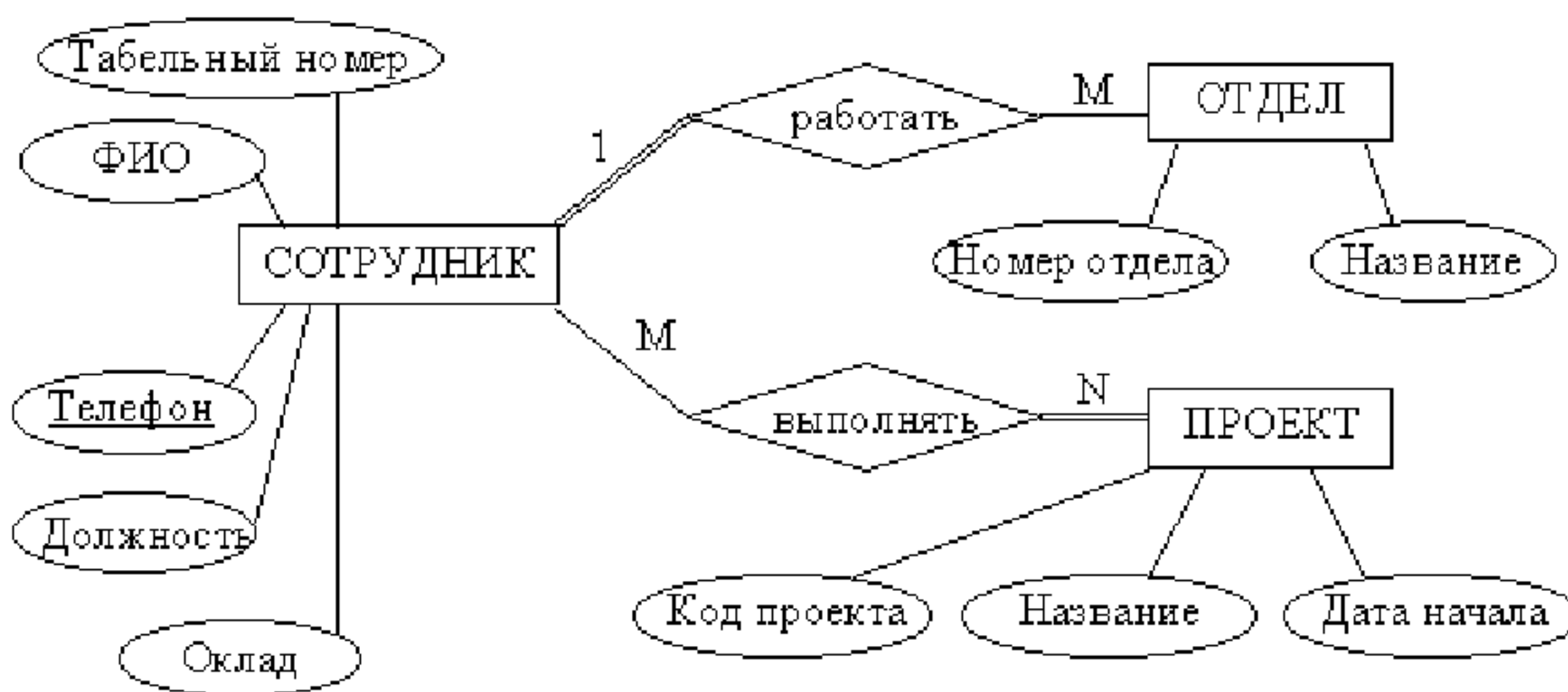


Рисунок 4.1 - Пример ER-диаграммы с однозначными и многозначными атрибутами

В качестве примера возьмем базу данных «Учебная» Т.И.Трофимовой.

В качестве примера возьмем базу данных компании, которая занимается издательской деятельностью.

База данных создаётся для информационного обслуживания редакторов, менеджеров и других сотрудников компании. БД должна содержать данные о сотрудниках компании.

книгах, авторах, финансовом состоянии компании и предоставлять возможность получать разнообразные отчёты.

В соответствии с предметной областью система строится с учётом следующих особенностей:

- каждая книга издаётся в рамках контракта;
- книга может быть написана несколькими авторами;
- контракт подписывается одним менеджером и всеми авторами книги;
- каждый автор может написать несколько книг (по разным контрактам);
- порядок, в котором авторы указаны на обложке, влияет на размер гонорара;
- если сотрудник является редактором, то он может работать одновременно над несколькими книгами;
- у каждой книги может быть несколько редакторов, один из них – ответственный редактор;
- каждый заказ оформляется на одного заказчика;
- в заказе на покупку может быть перечислено несколько книг.

Выделим базовые сущности этой предметной области:

1. Сотрудники компании. Атрибуты сотрудников – ФИО, табельный номер, пол, дата рождения, паспортные данные, ИНН, должность, оклад, домашний адрес и телефоны. Для редакторов необходимо хранить сведения о редактируемых книгах; для менеджеров – сведения о подписанных контрактах.
1. Авторы. Атрибуты авторов – ФИО, ИНН (индивидуальный номер налогоплательщика), паспортные данные, домашний адрес, телефоны. Для авторов необходимо хранить сведения о написанных книгах.
2. Книги. Атрибуты книги – авторы, название, тираж, дата выхода, цена одного экземпляра, общие затраты на издание, авторский гонорар.
3. Контракты будем рассматривать как связь между авторами, книгами и менеджерами. Атрибуты контракта – номер, дата подписания и участники.
4. Для отражения финансового положения компании в системе нужно учитывать заказы на книги. Для заказа необходимо хранить номер заказа, заказчика, адрес заказчика, дату поступления заказа, дату его выполнения, список заказанных книг с указанием количества экземпляров.

ER–диаграмма издательской компании приведена на рис. 2 (базовые сущности на рисунках выделены полужирным шрифтом).

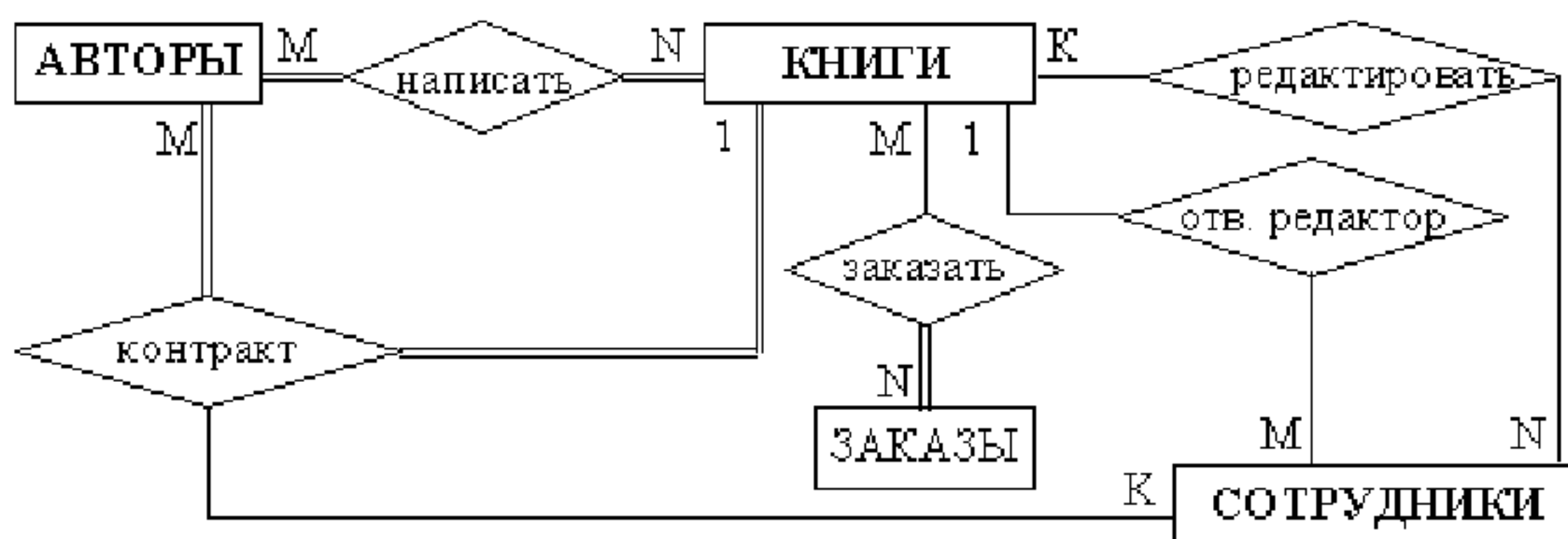


Рисунок 4.2 - ER–диаграмма издательской компании

Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

- администрация (дирекция);
- менеджеры;
- редакторы;
- сотрудники компании, обслуживающие заказы.

Сертификат:
Владелец:

2C0000043E9AB8B952205E7BA500060000043E
Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Определим границы информационной поддержки пользователей:

1) Функциональные возможности:

- ведение БД (запись, чтение, модификация, удаление в архив);
- обеспечение логической непротиворечивости БД;
- обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
- реализация наиболее часто встречающихся запросов в готовом виде;
- предоставление возможности сформировать произвольный запрос на языке манипулирования данными.

2) Готовые запросы:

- получение списка всех текущих проектов (книг, находящихся в печати и в продаже);
- получение списка редакторов, работающих над книгами;
- получение полной информации о книге (проекте);
- получение сведений о конкретном авторе (с перечнем всех книг);
- получение информации о продажах (по одному или по всем проектам);
- определение общей прибыли от продаж по текущим проектам;
- определение размера гонорара автора по конкретному проекту

При выполнении лабораторной работы по заданному преподавателем описанию предметной области построить концептуальную модель базы данных:

- Выделите типы сущностей;
- Выделите типы связей и определите для них показатели кардинальности и степень участия сторон;
- Выделите атрибуты и свяжите их типами сущностей и связей;
- Определите потенциальные и первичные ключи сущностей;
- Нарисуйте ER-диаграмму.

и проанализируйте информационные задачи и группы пользователей.

Техническое задание – документ, утвержденный в соответствующем порядке, определяющий цели, требования и основные исходные данные, необходимые для разработки аналитической информационной системы и содержащий предварительную оценку эффективности системы.

Утвержденное техническое задание является документом, которым разработчики руководствуются на всех этапах создания системы. Изменения, вносимые в техническое задание, должны согласовываться с заказчиком и разработчиком вместе. При разработке технического задания следует:

- установить общую цель создания ЭИС;
- разработать и обосновать требования к информационным подсистемам;
- разработать и обосновать требования к информационной базе, программному обеспечению;
- определить этапы создания системы и сроки их выполнения
- провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности.

При утверждении технического задания разрабатывается график работы и производится полный расчет затрат на разработку системы.

ТЗ содержит следующие разделы, которые могут быть разделены на подразделы:

- общие сведения;
- назначение и цели создания (развития) системы;
- характеристика объектов;
- требования к системе;
- состав и содержание работ по созданию системы;

- порядок контроля и приемки системы;
- требования к составу и содержанию работ по подготовке объекта разработки к вводу системы в действие;
- требования к документированию;
- источники разработки. В ТЗ могут включаться приложения.

В зависимости от вида, назначения, специфических особенностей проекта и условий функционирования системы допускается оформлять разделы ТЗ в виде приложений, вводить дополнительные, исключать или объединять подразделы ТЗ.

В ТЗ на части системы не включают разделы, дублирующие содержание разделов ТЗ в целом.

В разделе “Общие сведения” указывают:

- полное наименование системы и ее условное обозначение;
- шифр темы или шифр (номер) договора;
- наименование компаний разработчика и заказчика (пользователя) системы и их реквизиты;
- перечень документов, на основании которых создается система, кем и когда утверждены эти документы;
- плановые сроки начала и окончания работы по созданию системы;
- сведения об источниках и порядке финансирования работ;
- порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы.

Раздел “Назначение и цели создания (развития) системы” состоит из подразделов:

- назначение системы;
- цели создания системы.

В подразделе “Назначение системы” указывают вид деятельности системы (управление, проектирование и т. п.) и перечень объектов информатизации (объектов), на которых предполагается ее использовать.

В подразделе “Цели создания системы” приводят наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта информатизации, которые должны быть достигнуты в результате создания ИС, и указывают критерии оценки достижения целей создания системы.

В разделе “Характеристики объекта информатизации” приводят:

- краткие сведения об объекте информатизации или ссылки на документы, содержащие такую информацию;
- сведения об условиях эксплуатации объекта автоматизации.

Раздел “Требования к системе” состоит из следующих подразделов:

- требования к системе в целом;
- требования к функциям (задачам), выполняемым системой;
- требования к видам обеспечения.

Состав требований к системе, включаемых в данный раздел ТЗ на ИС, устанавливают в зависимости от вида, назначения, специфических особенностей и условий функционирования конкретной системы.

В подразделе “Требования к системе в целом” указывают:

- требования к структуре и функционированию системы;
- требования к численности и квалификации персонала системы и режиму его работы;
- показатели назначения;
- требования к надежности;
- требования безопасности;
- требования к эргономике и технической эстетике;

- требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
- требования к защите информации от несанкционированного доступа;
- требования по сохранности информации при авариях;
- требования к защите от влияния внешних воздействий;
- требования к патентной чистоте;
- требования по стандартизации и унификации;
- дополнительные требования.

В требованиях к структуре и функционированию системы приводят:

- перечень подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы;
- требования к способам и средствам связи для информационного обмена между компонентами системы;
- требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости, в том числе указания о способах обмена информацией (автоматически, пересылкой документов, по телефону и т. п.);
- требования к режимам функционирования системы;
- требования по диагностированию системы;
- перспективы развития, модернизации системы.

В требованиях к численности и квалификации персонала на ИС приводят:

- требования к численности персонала (пользователей) ИС;
- требования к квалификации персонала, порядку его подготовки и контроля знаний и навыков;
- требуемый режим работы персонала ИС.

В требованиях к показателям назначения ИС приводят значения параметров, характеризующие степень соответствия системы ее назначению.

В требованиях к надежности включают:

- состав и количественные значения показателей надежности для системы в целом или ее подсистем;
- перечень аварийных ситуаций, по которым должны быть регламентированы требования к надежности, и значения соответствующих показателей;
- требования к надежности технических средств и программного обеспечения;
- требования к методам оценки и контроля показателей надежности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами.

В требованиях по безопасности включают требования по обеспечению безопасности при поставке, наладке, эксплуатации и обслуживании системы.

В требованиях по эргономике и технической эстетике включают показатели ИС, задающие необходимое качество взаимодействия человека с машиной и комфортность условий работы персонала.

В требованиях к защите информации от несанкционированного доступа включают требования, установленные действующей в отрасли и информационной среде заказчика.

В требованиях по сохранности информации приводят перечень событий: аварий, отказов технических средств (в том числе - потеря питания) и т. п., при которых должна быть обеспечена сохранность информации в системе.

В требованиях по патентной чистоте указывают перечень стран, в отношении которых должна быть обеспечена патентная чистота системы и ее частей.

В дополнительных требованиях включают специальные требования по усмотрению разработчика или заказчика системы.

В подразделе "Требования к функциям (задачам)", выполняемым системой, приводят:

- по каждой подсистеме перечень функций, задач или их комплексов (в том числе обеспечивающих взаимодействие частей системы), подлежащих автоматизации;
- при создании системы в две или более очереди - перечень функциональных подсистем, отдельных функций или задач, вводимых в действие в 1-й и последующих очередях;
- временной регламент реализации каждой функции, задачи (или комплекса задач);
- требования к качеству реализации каждой функции (задачи или комплекса задач), к форме представления выходной информации, характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций, достоверности выдачи результатов;
- перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

В подразделе “Требования к видам обеспечения” в зависимости от вида системы приводят требования к математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другие видам обеспечения системы.

Для информационного обеспечения системы приводят требования:

- к составу, структуре и способам организации данных в системе;
- к информационному обмену между компонентами системы;
- к информационной совместимости со смежными системами;
- по применению систем управления базами данных;
- к структуре процесса сбора, обработки, передачи данных в системе и представлению данных;
- к защите данных;
- к контролю, хранению, обновлению и восстановлению данных;

Для лингвистического обеспечения системы приводят требования к применению в системе языков программирования высокого уровня, языков взаимодействия пользователей и технических средств системы, а также требования к кодированию и декодированию данных, к языкам ввода-вывода данных, языкам манипулирования данными, средствам описания предметной области, к способам организации диалога.

Для программного обеспечения системы приводят перечень покупных программных средств, а также требования:

- к зависимости программных средств от операционной среды;
- к качеству программных средств, а также к способам его обеспечения и контроля;

Для технического обеспечения системы приводят требования:

- к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию в системе;
- к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения системы.

В требованиях к метрологическому обеспечению приводят:

- предварительный перечень измерительных каналов;
- требования к точности измерений параметров и (или) к метрологическим характеристикам измерительных каналов;
- требования к метрологической совместимости технических средств системы;

перечень управляющих и вычислительных каналов системы, для которых необходимо оценивать точностные характеристики;

- требования к метрологическому обеспечению технических и программных средств, входящих в состав измерительных каналов системы, средств, встроенного

контроля, метрологической пригодности измерительных каналов и средств измерений, используемых при наладке и испытаниях системы;

- вид метрологической аттестации (государственная или ведомственная) с указанием порядка ее выполнения и организаций, проводящих аттестацию.

Для организационного обеспечения приводят требования:

- к структуре и функциям подразделений, участвующих в функционировании системы или обеспечивающих эксплуатацию;
- к организации функционирования системы и порядку взаимодействия персонала ИС и персонала объекта информатизации;
- к защите от ошибочных действий персонала системы.

Раздел “Состав и содержание работ по созданию (развитию) системы” должен содержать перечень стадий и этапов работ по созданию системы, сроки их выполнения, перечень организаций - исполнителей работ, ссылки на документы, подтверждающие согласие этих организаций на участие в создании системы, или запись, определяющую ответственного (заказчик или разработчик) за проведение этих работ.

В данном разделе также приводят:

- перечень документов предъявляемых по окончании соответствующих стадий и этапов работ;
- вид и порядок проведения экспертизы технической документации (стадия, этап, объем проверяемой документации, организация-эксперт);
- программу работ, направленных на обеспечение требуемого уровня надежности разрабатываемой системы (при необходимости);
- перечень работ по метрологическому обеспечению на всех стадиях создания системы с указанием их сроков выполнения и организаций-исполнителей (при необходимости).

В разделе “Порядок контроля и приемки системы” указывают:

- виды, состав, объем и методы испытаний системы и ее составных частей;
- общие требования к приемке работ по стадиям, порядок согласования и утверждения приемочной документации;

В разделе “Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие” необходимо привести перечень основных мероприятий и их исполнителей, которые следует выполнить при подготовке проекта к вводу ИС в действие.

В перечень основных мероприятий включают:

- приведение поступающей в систему информации (в соответствии с требованиями к информационному и лингвистическому обеспечению);
- создание условий функционирования проекта, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ;
- создание необходимых для функционирования системы подразделений и служб;
- сроки и порядок комплектования штатов и обучения персонала.

В разделе “Требования к документированию” приводят:

- согласованный разработчиком и Заказчиком системы перечень подлежащих разработке комплектов и видов документов; перечень документов, выпускаемых на машинных носителях;
- при отсутствии государственных стандартов, определяющих требования к документированию элементов системы, дополнительно включают требования к составу и содержанию таких документов.

В разделе “Источники разработки” должны быть перечислены документы и информационные материалы, на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании системы.

Ход лабораторной работы:

Разработать прототип информационной системы для заданной предметной области.
Провести тестирование ПО.

1. Разработка программы учета ресурсов для проекта информационной системы класса MRP.
2. Разработка программы обработки данных по производственной практике.
3. Разработка программы учета данных для проекта информационной системы «Турагентство».
4. Разработка проекта информационной системы «Санаторно-курортная деятельность».
5. Разработка программы вывода афиши для проекта информационной системы «Кинотеатр».
6. Разработка проекта информационной системы «Салон красоты».
7. Разработка программы ввода-вывода данных для проекта информационной системы «Аптека».
8. Разработка проекта информационной системы «Школа».
9. Разработка проекта информационной системы «Регистратура поликлиники».
10. Разработка проекта информационной системы «Гостиница».
11. Разработка проекта информационной системы «Авиабилеты».
12. Разработка проекта информационной системы «Кинотеатр».
13. Разработка проекта информационной системы «Книжный магазин».
14. Разработка проекта информационной системы «Компьютерный магазин».
15. Разработка проекта информационной системы «Учет курсовых проектов».

Вопросы для обсуждения:

1. Разработка структуры базы данных.
2. Разработка программного интерфейса информационной системы.
3. Тестирование функциональности прототипа информационной системы.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: отчет к лабораторной работе.

Тема 5. Спецификация функциональных и обеспечивающих подсистем информационной системы

Лабораторная работа № 5. Выбор архитектуры информационной системы.

Цель и содержание работы: Приобрести практические навыки в разработке спецификации функциональных и обеспечивающих подсистем информационной системы.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Организация процесса разработки спецификации.

Теоретическое обоснование

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен с 19.09.2022 по 19.09.2023

Реализация информационных технологий основана на рационально организованном технологическом процессе обработки экономической информации.

Технологический процесс автоматизированной обработки экономической информации представляет собой совокупность операций, выполняемых в строго регламентированной последовательности, начиная от сбора (регистрации) исходной информации до передачи результатной информации заинтересованным пользователям для выполнения функций управления.

Под **операцией** в данном случае понимается комплекс действий, выполняемых над информацией на одном рабочем месте.

Разработка технологического процесса должна обеспечить максимальную автоматизацию процессов обработки информации при использовании различных технологических средств и высокую достоверность получения результатной информации при минимальных трудовых и стоимостных затратах.

Состав операций и последовательность их выполнения зависят от характера решаемых экономических задач и имеющегося комплекса технических средств.

Характер задач в основном определяется объемами обрабатываемой информации, периодичностью решения, а также сложностью алгоритмов ее преобразования. Решение задачи на каждом уровне обработки может выполняться разными способами с применением различных вычислительных средств и методов контроля.

При определении состава операций технологического процесса создается возможность выбора наиболее рационального способа обработки информации.

В технологическом процессе автоматизированной обработки экономической информации при решении различных задач прослеживаются **три основных этапа**: первичный, основной и заключительный.

Операции **первичного этапа** связаны со сбором и подготовкой первичных документов к машинной обработке. От качества выполнения этих операций во многом зависит эффективность и достоверность непосредственной автоматизированной обработки.

Операции **основного этапа** обеспечивают ввод и обработку первичных документов, а также вывод результатных документов в соответствии с заданным алгоритмом решения экономических задач.

На **заключительном этапе** выполняются операции контроля и выпуска результатных документов с целью проверки качества их обработки, а также рассылки различным потребителям.

Так как при выполнении отдельных операций технологического процесса обработки информации могут быть использованы различные средства вычислительной техники и в различных сочетаниях, на практике существует множество вариантов технологических процессов. Эффективность проектирования автоматизированной обработки экономической информации требует их стандартизации.

Под **стандартизацией технологии автоматизированной обработки информации** понимается разработка комплекса детализированных и максимально унифицированных схем технологических процессов, в которых строго установлены состав и последовательность выполнения операций.

Ориентация на стандартные технологические процессы приводит к улучшению качества проектирования, облегчению и ускорению внедрения проектов автоматизированной обработки данных. Поэтому подробнее рассмотрим типовую схему технологии обработки информации при решении экономической задачи, представленную на рис. 5.1.

Первичный этап начинается со сбора первичных документов от различных источников информации (экономических служб предприятий или фирм), кодирования необходимых реквизитов, а также контроля полноты и качества их заполнения экономистами (опер. 1). Если обнаруживаются ошибки, то экономисты делают запросы к источникам информации (опер. 3).

В противном случае по ним производится подсчет и регистрация контрольных сумм (опер. 4). Контрольные суммы необходимы в дальнейшем для проверки правильности ввода данных в ЭВМ и в зависимости от структуры первичного документа они могут подсчитываться по каждой строке, графе или по отчету в целом.

Первичные данные и промежуточные результаты по задачам могут быть представлены в электронном виде, а при вводе машиночитаемых документов используются сканирующие устройства.

Основной этап начинается со ввода первичных данных с клавиатуры, их контроля (арифметического, логического и синтаксического) и, независимо от результатов контроля, записи информации в файл или информационную базу (опер. 5).

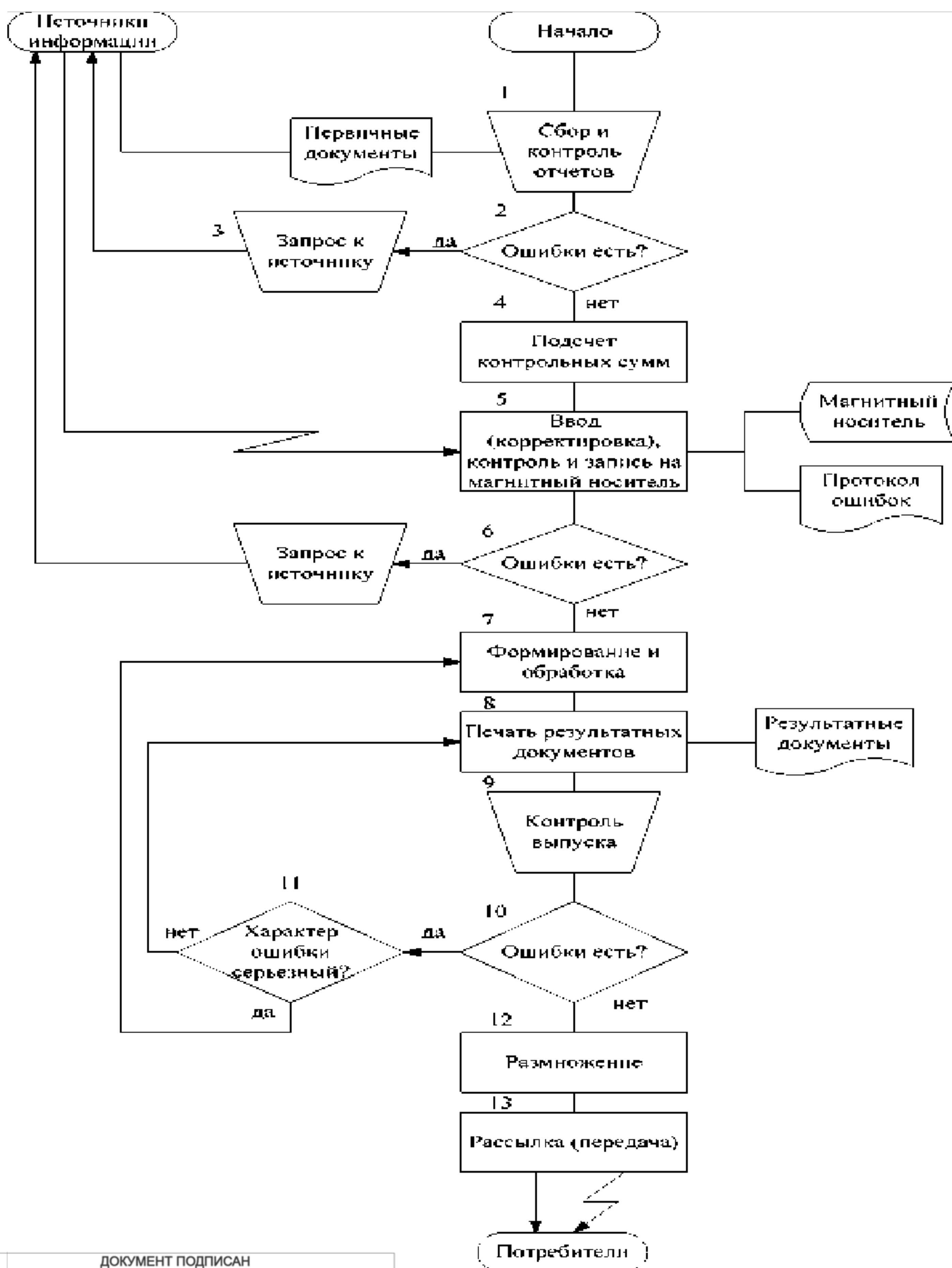
Арифметический контроль заключается в проверке баланса данных в строках и графах первичного документа с применением заранее подсчитанных контрольных сумм. При логическом контроле проверяются тип данных, допустимые отклонения значений от наиболее вероятных, предельные значения данных, а также различные смысловые проверки с использованием специальных алгоритмов. С помощью синтаксического контроля проверяется структура заполнения документа различными данными.

При этом осуществляется проверка полноты ввода и своевременная корректировка первичных документов на магнитных носителях, если ошибка связана с набором информации.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023



ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AD8B952205E7BA30000000043E
Владелец: Татьяна Александровна

Действителен до 12.01.2017

Рисунок 5.1. Типовая схема технологического процесса обработки экономической информации

Если при контроле обнаружилось нарушение, связанное с содержательной стороной представленного первичного документа, то производится печать протокола

ошибок, в котором указывается характер нарушений, конкретная ошибка и ее местонахождение в файле или информационной базе.

На основании этого протокола делается запрос к источникам информации (опер. 3), где выясняются ошибки и исправленные данные поступают по каналам связи (факс или электронная почта) для последующей корректировки информационной базы (опер. 5). Только после внесения всех изменений и проверки контроля полноты записи первичных документов в информационной базе приступают к формированию и обработке результатных данных (опер. 7) согласно заданному алгоритму решения задачи, а затем осуществляется печать результатных документов (опер. 8).

На заключительном этапе технологического процесса обработки информации экономистами осуществляется контроль выпуска результатных документов (опер. 9), состоящий в проверке на полноту вхождения исходных данных в заданный разрез обработки и качества их печати. Учитывая характер ошибки (опер. 11), формирование и обработка повторяются или результатные данные печатаются заново.

Затем осуществляется размножение результатных данных с помощью различных копировально-множительных средств (опер. 12) и рассылка их в виде документов и магнитных носителей или передача их по каналам связи различным заинтересованным потребителям (опер. 13).

Рассмотренная типовая схема предусматривает, что первичная информация от источника может сразу поступать на ввод и контроль (опер. 5) в зависимости от того, каким образом она будет передаваться (по факсу или по электронной почте).

Основными документами, регламентирующими технологический процесс обработки экономической информации, являются технологические и инструкционные карты.

Технологическая карта представляет собой описание последовательно выполняемых операций технологического процесса по каждой экономической задаче.

На каждую операцию технологического процесса разрабатывается **инструкционная карта**, в которой указываются сведения об исходной информации, о конечных результатах и о порядке выполнения конкретной работы.

При выполнении лабораторной работы по заданному преподавателем описанию предметной области разработать схему автоматизированного процесса обработки информации прикладной информационной технологии.

Ход лабораторной работы:

Разработать спецификацию информационной системы для заданной предметной области.

1. Разработка программы учета ресурсов для проекта информационной системы класса MRP.
2. Разработка проекта информационной системы «Служба занятости».
3. Разработка проекта информационной системы «Фондовый рынок».
4. Разработка проекта информационной системы «Кредитный эксперт».
5. Разработка программы учета клиентов для проекта информационной системы класса CRM.
6. Разработка программы оперативного учета поступления товара на склад.
7. Разработка проекта информационной системы «Недвижимость».
8. Разработка проекта информационной системы «Каталог товаров».
9. Разработка проекта информационной системы «Зарплата».
10. Разработка проекта информационной системы «Поставщики предприятия».
11. Разработка проекта информационной системы инвентаризации аппаратных средств предприятия.
12. Разработка проекта информационной системы учета программных средств предприятия.

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
 Владелец: Шебзукова Татьяна Александровна
 Действителен: с 19.08.2022 по 19.08.2023

13. Разработка проекта информационной системы тестирования знаний.
14. Разработка проекта информационной системы «Абитуриент».
15. Разработка проекта информационной системы статистической обработки данных.
16. Разработка проекта информационной системы «Библиотека».
17. Разработка программы учета данных для проекта информационной системы «Реклама».
18. Разработка проекта информационной системы учета успеваемости студентов.
19. Разработка проекта информационной системы «Выпускник вуза».
20. Разработка проекта информационной системы «Отдел кадров».
21. Разработка программы оперативного учета продаж торгового предприятия.

Вопросы для обсуждения:

1. Архитектура информационной системы.
2. Спецификация базы данных и программных модулей информационной системы.
Спецификация аппаратно-программно-аппаратной инфраструктуры информационной системы.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: отчет к лабораторной работе.

Тема 6. Контроль качества и надежности информационной системы

Лабораторная работа № 6. Оценка качества и надежности информационной системы

Цель и содержание работы: изучить основные характеристики качества информационных систем.

Организационная форма занятий: разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Совокупность свойств системы, обуславливающих возможность ее использования для удовлетворения определенных в соответствии с ее назначением потребностей.

Теоретическое обоснование

Качество информационной системы — это совокупность свойств системы, обуславливающих возможность ее использования для удовлетворения определенных в соответствии с ее назначением потребностей. Количественные характеристики этих свойств определяются показателями.

Основными показателями качества информационных систем являются надежность, достоверность, безопасность.

Надежность — свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения.

Надежность информационных систем не самоцель, а средство обеспечения своевременной и достоверной информации на ее выходе. Поэтому показатель достоверности функционирования имеет для информационных систем главенствующее значение, тем более что показатель своевременности информации в общем случае охватывается показателем достоверности.

Достоверность функционирования — свойство системы, обуславливающее безошибочность производимых ею преобразований информации. Достоверность функционирования информационной системы полностью определяется и измеряется достоверностью ее выходной информации.

Безопасность информационной системы — свойство, заключающееся в способности системы обеспечить конфиденциальность и целостность информации, то есть защиту информации от несанкционированного доступа с целью ее раскрытия, изменения или разрушения.

Эффективность — это свойство системы выполнять поставленную цель в заданных условиях использования и с определенным качеством. Показатели эффективности характеризуют степень приспособленности системы к выполнению поставленных перед ней задач и являются обобщающими показателями оптимальности функционирования ИС, зависящими от локальных показателей, каковыми являются надежность, достоверность, безопасность.

Кардинальным обобщающим показателем является *экономическая эффективность системы*, характеризующая целесообразность произведенных на создание и функционирование системы затрат.

Надежность информационных систем

Надежность — важнейшая характеристика качества любой системы, поэтому разработана специальная теория — теория надежности.

Теория надежности может быть определена как научная дисциплина, изучающая закономерности, которых следует придерживаться при разработке и эксплуатации систем для обеспечения оптимального уровня их надежности с минимальными затратами ресурсов.

Надежность — характеристика временная, она может быть ориентирована либо в прошлое, либо в будущее время и не допускает «точечных» во времени оценок. Иными словами, надежность — это свойство системы «штатно» функционировать во времени.

Надежность — комплексное свойство системы; оно включает в себя более простые свойства, такие как безотказность, ремонтпригодность, долговечность и т. д.

Безотказность — свойство системы сохранять работоспособное состояние в течение некоторого времени или наработки (наработка — продолжительность или объем работы системы).

Ремонтпригодность — свойство системы, заключающееся в приспособленности к предупреждению и обнаружению причин возникновения отказов, повреждений и поддержанию и восстановлению работоспособного состояния путем проведения технического обслуживания и ремонтов.

Долговечность — свойство системы сохранять при установленной системе технического обслуживания и ремонта работоспособное состояние до наступления предельного состояния, то есть такого момента, когда дальнейшее использование системы по назначению недопустимо или нецелесообразно.

Одним из основных понятий теории надежности является отказ. *Отказом* называют полную или частичную потерю работоспособности системы или ее элемента. Отказы бывают: внезапные и постепенные, зависимые и независимые, полные и частичные, устойчивые и самоустраняющиеся, аппаратные и программные и т. п.

Устойчивый отказ обуславливает длительную неработоспособность системы и устраняется только в результате ее технического обслуживания, то есть выполнения специальных мер, принятых для восстановления работоспособности системы.

Самоустраняющийся отказ (обычно его называют *сбоем*) — отказ, имеющий кратковременный характер и самоустраняющийся произвольно, без принятия специальных мер для его устранения. Ряд сбоев одного и того же характера, следующих друг за другом, называют перемежающимся отказом.

Аппаратный отказ обусловлен нарушением работоспособности технического элемента системы, соответственно эргатический – эргатического элемента, программный — программного элементов системы. В соответствии с приведенной классификацией отказов можно рассматривать и надежность:

- аппаратную;
- эргатическую;
- программную.

Вмногофункциональных системах часто вводят понятие *функциональной надежности* выполнения локальной функции системы. Это понятие важно тогда, когда разные функции системы различны по значимости, обеспечиваются различными подсистемами и дифференцируются по предъявляемым к ним требованиям.

Все системы в теории надежности классифицируются по ряду признаков. Важными классификационными группами являются:

- восстанавливаемые;
- невосстанавливаемые;
- обслуживаемые;
- необслуживаемые системы.

Восстанавливаемой называется такая системы, работоспособность которой в случае возникновения отказа подлежит восстановлению. *Невосстанавливаемая* система — такая система, работоспособность которой в случае отказа восстановлению не подлежит.

Обслуживаемая система — система, для которой предусматривается проведение регулярного технического обслуживания. *Необслуживаемая* система — система, для которой не предусматривается проведение регулярного технического обслуживания.

Информационные и вычислительные системы первых поколений, за редким исключением, относятся к восстанавливаемым обслуживаемым системам. Многие современные вычислительные системы относятся к необслуживаемым восстанавливаемым системам (например персональные компьютеры) и даже к необслуживаемым и невосстанавливаемым системам (отдельные узлы вычислительных систем, например микропроцессор).

Основные показатели надежности

Показатель надежности — это количественная характеристика одного или нескольких свойств, определяющих надежность системы. В основе большинства показателей надежности лежат оценки наработки системы, то есть продолжительности или объема работы, выполненной системой. Показатель надежности, относящийся к одному из свойств надежности, называется единичным. Комплексный показатель надежности характеризует несколько свойств, определяющих надежность системы.

Ниже приводятся наименования основных показателей надежности систем и их определения в соответствии с ГОСТ 27.002-80 «Надежность в технике. Термины и определения».

Единичные показатели надежности

К единичным показателям надежности в соответствии с ГОСТ 27.002-80 относятся показатели безотказности, показатели ремонтпригодности и показатели долговечности.

Показатели безотказности

1. Вероятность безотказной работы — вероятность того, что в пределах заданной наработки отказ системы не возникнет.

2. Вероятность отказа — обратная величина, вероятность того, что в пределах заданной наработки отказ системы возникнет.

3. Средняя наработка до отказа — математическое ожидание наработки системы до первого отказа (существенно для невосстанавливаемых систем).

4. Средняя наработка на отказ (T_0 , MTBF — Main Time Between Failures) — отношение наработки восстанавливаемой системы к математическому ожиданию числа ее отказов в пределах этой наработки (имеет смысл только для восстанавливаемых систем).

$$K_r = \frac{T_o}{T_o + T_{\text{в}}}$$

$$K_{\text{ин}} = \frac{T_o}{T_o + T_{\text{в}} + T_{\text{п}}}$$

5. Интенсивность отказов — условная плотность вероятности возникновения отказа невосстанавливаемой системы, определяемая для рассматриваемого момента времени при условии, что до этого момента отказ не возник.

6. Параметр потока отказов ($X(t)$) — отношение среднего числа отказов для восстанавливаемой системы за произвольно малую ее наработку к значению этой наработки.

Показатели ремонтпригодности

1. Вероятность восстановления работоспособного состояния — вероятность того, что время восстановления работоспособного состояния не превысит заданного.

2. Среднее время восстановления работоспособного состояния ($T_{\text{в}}$) — математическое ожидание времени восстановления работоспособного состояния системы.

Показатели долговечности

1. Средний ресурс — математическое ожидание наработки системы от начала ее эксплуатации или ее возобновления после ремонта до перехода в предельное состояние.

2. Срок службы ($T_{\text{ср}}$) — календарная продолжительность от начала эксплуатации системы или ее возобновления после ремонта до перехода в предельное состояние.

Комплексные показатели надежности

1. Коэффициент готовности ($K_{\text{г}}$) — вероятность того, что система окажется в работоспособном состоянии в произвольный момент времени, кроме планируемых периодов, в течение которых применение системы по назначению не предусматривается.

2. Коэффициент оперативной готовности — вероятность того, что система окажется в работоспособном состоянии в произвольный момент времени, кроме планируемых периодов, в течение которых применение системы по назначению не предусматривается, и начиная с этого момента будет работать безотказно в течение заданного времени.

3. Коэффициент технического использования — отношение математического ожидания интервалов времени пребывания системы в работоспособном состоянии за некоторый период эксплуатации к сумме математических ожиданий интервалов времени пребывания системы в работоспособном состоянии, простоев, обусловленных техническим обслуживанием, и ремонтов за тот же период эксплуатации.

где $T_{\text{п}}$ — время простоя системы, обусловленное выполнением планового технического обслуживания и ремонта (время профилактики), пересчитанное на один отказ.

4. Коэффициент сохранения эффективности — отношение значения показателя эффективности за определенную продолжительность эксплуатации к номинальному значению этого показателя, вычисленному при условии, что отказы в системе в течение того же периода эксплуатации не возникают.

Коэффициент сохранения эффективности характеризует степень влияния отказов в системе на эффективность ее применения по назначению. Из ранее приведенного определения теории надежности следует, что коэффициент сохранения эффективности может служить интегральным критерием оптимизации надежности системы. Действительно, критерий оптимизации — это показатель, для которого указана желаемая его величина или желаемое направление его изменения. Направление изменения коэффициента сохранения правильно выбранного показателя эффективности определяет основные ориентиры в поиске свойств системы, которые обеспечивают ее оптимальную надежность.

Для пользователей сложных информационных систем понятие их надежности ощущается в наибольшей степени по коэффициенту готовности системы, то есть по отношению времени работоспособного состояния системы к времени ее незапланированного простоя. Для типичного современного сервера $K_g = 0,99$, что означает примерно 3,5 суток простоя в год. Часто используется следующая классификация систем по уровню надежности:

Коэффициент готовности	Максимальное время простоя в год	Тип системы
0,99	3,5 сут.	Обычная (Conventional)
0,999	8,5 ч	Высокой надежности (High availability)
0,9999	1 ч	Отказоустойчивая (Fault resilient)
0,99999	5 мин.	Безотказная (Fault tolerant)

Обеспечение надежности функционирования ИС

Информационная система — это сложная человеко-машинная система, включающая в свой состав эргатические звенья, технические средства и программное обеспечение. Все методы обеспечения надежности и достоверности ИС можно отнести к двум классам. Один включает в себя методы, обеспечивающие безошибочность (безотказность, бессбойность) функциональных технических, эргатических и программных звеньев ИС, то есть, в конечном счете, повышающие их надежность. Другой — методы, обеспечивающие обнаружение и исправление ошибок, возникающих в информации, то есть методы контроля достоверности информации и ее коррекции, косвенно также повышающие функциональную надежность системы.

Названные классы не исключают, а взаимно дополняют друг друга, поскольку в такой сложной системе, как информационная, обеспечить высокую надежность и достоверность функционирования можно, только сочетая методы обоих классов.

Виды обеспечения надежности

Для построения надежных информационных систем можно использовать различные виды обеспечения:

- экономическое;
временное;
организационное;
структурное;
технологическое;
эксплуатационное;
социальное;
эргатическое;
алгоритмическое;

синтаксическое,
семантическое

Обеспечение можно определить как совокупность факторов (элементов, методов, приемов, процедур, ресурсов и т. п.), способствующих достижению поставленной цели.

Экономическое и временное обеспечения, обусловливаемые необходимостью соответственно материальных и временных затрат, используются для реализации процедур обеспечения достоверности. Организационное, эксплуатационное, техническое, социальное и эргатическое обеспечения применяются преимущественно для повышения надежности систем, а структурное и алгоритмическое обеспечения — для обоих классов методов.

Организационное обеспечение включает в себя вопросы разработки:

- правовых и методических аспектов функционирования ИС;
- нормативов достоверности информации по функциональным подсистемам и этапам преобразования информации;
- методики выбора и обоснования оптимальных структур, процессов и процедур
- преобразования информации и т. д.

Назначением структурного обеспечения является повышение надежности функционирования технических комплексов и эргатических звеньев, а также ИС в целом. Должно быть обосновано рациональное построение структуры ИС, непосредственно зависящее от качества решения таких вопросов, как выбор структуры технологического процесса преобразования информации, обеспечение

обоснованных взаимосвязей между отдельными звеньями системы, резервирование элементов, узлов, устройств системы и использование специальных устройств, осуществляющих процедуры аппаратного контроля и т. д.

Технологическое и эксплуатационное обеспечения предназначены для повышения надежности работы технических средств и технологических комплексов. Технологическое обеспечение включает в себя выбор схемных и конструктивных решений применения отдельных технических устройств, технологий и протоколов реализации информационных процессов. Эксплуатационное обеспечение связано с выбором режимов работы устройств, технологий профилактического их обслуживания.

Социальное и эргатическое обеспечения имеют своим назначением повышение надежности работы эргатических структурных звеньев системы. Поскольку подавляющее большинство ошибок в информации возникает как раз из-за функциональной ненадежности именно этих звеньев (человеческого фактора), в литературе особенности их работы рассмотрены весьма обстоятельно. Различают пять видов причин ошибок, возникающих в эргатических звеньях:

- психологические — неадекватность восприятия информации, выработка и реализация неоптимальной стратегии;
- мотивационные — неправильная постановка задачи, несогласованность целей субъекта с целями управления;
- эмоциональные — неустойчивые изменения преобразующих свойств субъекта от внешних и внутренних причин;
- интуитивные — неформализованный в сознании субъекта опыт, отражающий реальную ситуацию нерелевантно;
- эволюционные — устойчивые изменения преобразующих свойств субъекта в результате
- обучения или забывания.
- Указанные причины могут привести к субъективным ошибкам трех типов:
- потере части полезной информации;
- внесению дополнительной (полезной или вредной) информации, не содержащейся в исходном сообщении;
- неадекватному преобразованию информации.

К социальному обеспечению относятся, например, такие факторы, как создание здоровой психологической обстановки в коллективе, повышение ответственности за выполненную работу, повышение квалификации специалистов, увеличение моральной и

материальной заинтересованности в правильности выполнения работы. Особенно важно обеспечить согласованность целей субъекта с целями управления: лишь тогда, когда работник заинтересован в получении объективных, достоверных данных, они могут быть получены.

Эргатическое обеспечение включает в себя комплекс факторов, связанных с рациональной организацией работы человека в системе. Это, в первую очередь, правильное распределение функций между людьми и техническими средствами, обоснованность норм и стандартов работы, оптимальность интенсивности и ритмичности, построение рабочих мест в соответствии с требованиями эргономики.

Алгоритмическое обеспечение широко применяется для повышения надежности системы (обеспечение высокого качества и безошибочности алгоритмов и программ преобразования информации) и для реализации контроля достоверности информации.

Информационное синтаксическое и семантическое обеспечения заключаются во введении в ИС специальной информационной избыточности, соответственно, избыточности данных и смысловой избыточности, обуславливающих возможность проведения контроля достоверности информации.

Практическая реализация надежных информационных систем

Обеспечение надежности технических компонентов информационных систем чаще всего реализуется аппаратным и программным способами.

В первом случае ИС использует аппаратную избыточность:

- все операции выполняются параллельно на одинаковых компонентах системы, а результаты их работы затем сравниваются, что позволяет выявить ошибки;
- в случае выхода из строя какого-либо компонента его резервные аналоги продолжают
- работу без остановки, а отказавший компонент заменяется на работоспособный.

Программный способ предусматривает:

- последовательное во времени выполнение одних и тех же информационных процессов и дублирование данных;
- автоматическое восстановление отказавших операционных систем, приложений и
- искаженных данных.

На сегодняшний день разработано много конкретных практических способов повышения надежности информационных систем.

Для обеспечения надежности технических средств чаще всего производится:

- резервирование (дублирование) технических средств (компьютеров и их компонентов, сегментов сетей и т. д.);
- использование стандартных протоколов работы устройств ИС;
- применение специализированных технических средств защиты информации.

Для обеспечения надежности функционирования программного комплекса И С требуется:

- тщательное тестирование программ, опытное исполнение программы с целью обнаружения в ней ошибок (обязательное условие эффективного тестирования — по крайней мере один раз выполнить все разветвления программы в каждом из возможных направлений);
- использование стандартных протоколов, интерфейсов, библиотек процедур, лицензионных программных продуктов;
- использование структурных методов для обеспечения надежной работы программных комплексов (иерархическое построение программ, разбиение программ на сравнительно независимые модули и т. д.);

- изоляция параллельно работающих процессов, в результате чего ошибки в работе одной программы не влияют на работу операционной системы и других программ.

Контрольные точки (точки рестарта, точки отката) — место повторного запуска программы при аварийном ее завершении. В контрольных точках обычно выполняются: внесение изменений в БД (в том числе всех изменений, ожидающих своей очереди — неоперативные файлы), разблокирование всех файлов, на обращение к которым был наложен запрет, запись информации о контрольной точке в системный журнал.

Использование массивов RAID (Redundant Array of Inexpensive Disks — избыточный массив недорогих дисков) существенно уменьшает риск простоя системы из-за отказов накопителей на магнитных дисках, которые являются одним из наименее надежных компонентов современных компьютеров.

В качестве наиболее эффективных мер комплексного обеспечения надежности ИС можно назвать кластеризацию компьютеров и использование отказоустойчивых компьютеров.

Кластер — это несколько компьютеров (узлов кластера), соединенных коммуникационными каналами и разделяющих общие ресурсы. Кластер имеет общую файловую систему и пользователем воспринимается как единый компонент. Надежность работы кластера обеспечивается программами, регулирующими скоординированное использование общекластерных ресурсов, обмен информацией между узлами кластера, и осуществляющими взаимный контроль работоспособности этих узлов. Отличительной особенностью кластера является то, что каждый его работающий компьютер может взять на себя дополнительную нагрузку отказавшего узла. Кластерные системы разрабатываются многими известными фирмами (IBM, Hewlett-Packard, DEC и т. д.). Все известные кластерные решения обеспечивают высокую готовность системы (коэффициент готовности до 0,999 — high availability), возможность наращивания производительности за счет установки нового оборудования или замены устаревшего. Кластерные системы используют специальные программы, осуществляющие оптимальное распределение ресурсов и удобное администрирование.

Например, компания IBM разработала кластерную систему для четырех или пяти компьютеров IBM RISC System/6000, реализующую следующие стратегии:

- один компьютер находится в постоянной готовности и подстраховывает три других работающих компьютера;
- работают четыре компьютера, а пятый, дополнительный компьютер подстраховывает остальные;
- работают четыре компьютера, и в случае отказа одного из них нагрузка пере распределяется между тремя работоспособными;
- четыре компьютера работают одновременно с одними и теми же общими данными (отказоустойчивый вариант кластера).

В отличие от кластерных отказоустойчивые системы упор делают на аппаратное обеспечение надежности и гарантируют не просто сокращение времени простоя (увеличение коэффициента готовности), а вообще предотвращение и исключение возможности появления таких простоев. В основу архитектуры отказоустойчивых систем заложено дублирование, в том числе и многократное, технических компонентов.

В отказоустойчивых компьютерах любая команда выполняется одновременно на всех дублированных компонентах, и результаты выполнения команд сравниваются. Окончательное решение принимается по принципу мажоритирования (по большинству одинаковых результатов). Каждый из продублированных компонентов продолжает работу и в случае отказа одного из его дублей таким образом, что система не замечает этого отказа и на ее функционировании это не отражается. Но отказавший компонент идентифицируется и замещается в режиме «горячей замены», то есть без отключения системы.

Достоверность информационных систем

В силу специфики информационных систем, которые априори предназначены для преобразования информации, важнейшим их свойством является достоверность функционирования.

Достоверность функционирования — это свойство системы, обуславливающее безошибочность производимых ею преобразований информации.

Достоверность функционирования ИС полностью определяется и измеряется достоверностью ее результирующей информации. Для ИС достоверность функционирования является не просто одним из свойств их надежности, но приобретает и самостоятельное значение, поскольку именно достоверность конечной информации обуславливает требования к надежности системы.

Как уже указывалось, надежность ИС — не самоцель, а лишь средство обеспечения оптимальной достоверности ее выходной информации, обуславливающей наивысшую эффективность функционирования системы.

Достоверность информации — это свойство информации отражать реально существующие объекты с необходимой точностью. Достоверность (D) информации измеряется доверительной вероятностью необходимой точности, то есть вероятностью того, что отражаемое информацией значение параметра отличается от истинного значения этого параметра в пределах необходимой точности:

$$D = P\{O[n]\}$$

где — реальная точность отображения параметра, $[\Delta n]$ — диапазон необходимой точности отображения параметра.

Для более полного понимания вышеприведенного определения следует пояснить некоторые присутствующие в нем понятия.

Истинная информация — информация, объективно, точно и правильно отражающая характеристики и признаки какого-либо объекта или явления (адекватная заданному параметру объекта).

Точность информации — это характеристика, показывающая степень близости отображаемого значения параметра и истинного его значения.

Необходимая точность определяется функциональным назначением информации и должна обеспечивать правильность принятия управленческих решений.

Таким образом, при оценке истинности информации существуют две основные вероятностные задачи:

- определение точности информации или расчет математического ожидания абсолютной величины отклонения значения показателя от объективно существующего истинного значения отображаемого им параметра;
- определение достоверности информации или вычисление вероятности того, что погрешность показателя не выйдет за пределы допустимых значений.

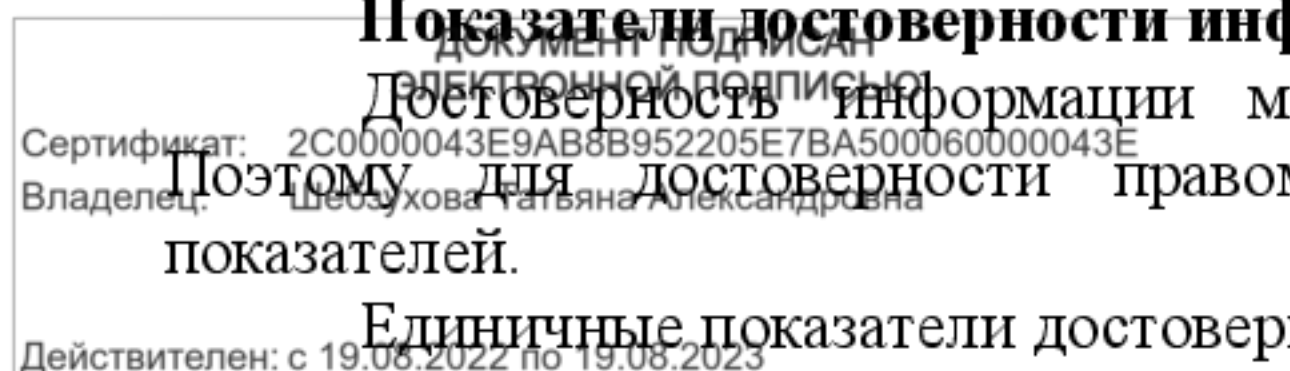
Адекватность отражения включает в себя понятия и точности, и достоверности, которые не должны смешиваться.

Из сказанного следует, что нарушение надежности ИС, приводящее к ухудшению точности результирующей информации в пределах необходимой точности, не снижает эффективности функционирования системы (коэффициента сохранения эффективности). И если отсутствие информации в положенное время (ее несвоевременность) трактовать в обобщенном виде как наличие недостоверной информации, то единственным показателем качества информации, зависящим от надежности ИС и влияющим на эффективность ее функционирования, является достоверность.

Показатели достоверности информации

Достоверность информации может рассматриваться с разных точек зрения. Поэтому для достоверности правомерно и целесообразно использовать систему показателей.

Единичные показатели достоверности информации



1. Доверительная вероятность необходимой точности (достоверность) — $D = 1 - P_{\text{ош}}$ Вероятность того, что в пределах заданной наработки (информационной совокупности — массива, показателя, реквизита, кодового слова, символа или иного информационного компонента) отсутствуют грубые погрешности, приводящие к нарушению необходимой точности.

2. Средняя наработка информации на ошибку — $Q = 1/P$. Отношение объема информации, преобразуемой в системе, к математическому ожиданию количества ошибок, возникающих в информации.

3. Вероятность ошибки (параметр потока ошибок) — $(P_{\text{ош}})$ Вероятность появления ошибки в очередной информационной совокупности.

Показатели корректируемости информационных систем

1. Вероятность коррекции в заданное время — $R_{\text{корр}}(\tau)$ — вероятность того, что время, затрачиваемое на идентификацию и исправление ошибки, не превысит заданного τ .

2. Среднее время коррекции информации — $T_{\text{и}}$ — математическое ожидание времени, затрачиваемого на идентификацию и исправление ошибки.

Комплексные показатели достоверности

1. Коэффициент информационной готовности — это вероятность того, что информационная система окажется способной к преобразованию информации в произвольный момент времени того периода ($T_{\text{раб}}$), который планировался для этого преобразования, то есть выполнения условия, что в данный момент времени система не будет находиться в состоянии внепланового обслуживания, вызванного устранением отказа или идентификацией и исправлением ошибки.

$$K_{\text{ИГ}} = \frac{T_{\text{раб}} - (T_{\text{В}} - T_{\text{И}})}{T_{\text{раб}}}$$

2. Коэффициент информационного технического использования — это отношение математического ожидания планируемого времени работы системы на преобразование информации, за вычетом времени восстановления, контроля, идентификации и исправления ошибок, к сумме планируемого времени работы системы и профилактического обслуживания.

$$K_{\text{ТИ}} = \frac{T_{\text{раб}} - (T_{\text{В}} + T_{\text{К}} + T_{\text{И}})}{T_{\text{раб}} + T_{\text{нф}}}$$

Наряду с понятием достоверности информации существует понятие достоверности данных, рассматриваемое в синтаксическом аспекте. Под достоверностью данных понимается их безошибочность. Она измеряется вероятностью отсутствия ошибок в данных (в отличие от достоверности информации, к снижению достоверности данных приводят любые погрешности, а не только грубые). Недостоверность данных может не повлиять на объем данных, но может и уменьшить и увеличить его, в отличие от недостоверности информации, всегда уменьшающей ее количество. Наконец, недостоверность данных может не нарушить достоверность информации (например, при наличии в последней необходимой избыточности).

Обеспечение достоверности информации

Одним из наиболее действенных средств обеспечения достоверности информации в ИС является ее контроль. Контроль — процесс получения и обработки информации с целью оценки соответствия фактического состояния объекта предъявляемым к нему требованиям и выработки соответствующего управляющего решения. Объектом контроля в нашем случае является достоверность информации, следовательно, при контроле должно быть выявлено соответствие фактической и необходимой точности представления информации или, с учетом рассмотренной ранее нормы этого соответствия, выявлено

наличие или отсутствие ошибок в контролируемой информации. При обнаружении ошибки должны быть приняты меры для ее устранения или, по крайней мере, выработаны соответствующие рекомендации по локализации и идентификации обнаруженной ошибки и уменьшению последствий ее влияния на функционирование ИС; исправление ошибок в последнем случае выполняется путем выполнения некоторых внешних относительно процедуры контроля операций.

Классификация методов контроля достоверности по назначению

По назначению следует различать профилактический, рабочий и генезисный контроль. Профилактический контроль и, например, одна из наиболее распространенных его форм —

тестовый контроль, предназначены для выявления состояния системы в целом и отдельных ее звеньев до включения системы в рабочий режим. Целью профилактического контроля, осуществляемого часто в утяжеленном режиме работы системы, является выявление и прогнозирование неисправностей в ее работе с последующим их устранением.

Рабочий контроль, или контроль в рабочем режиме, производится в процессе выполнения системой возложенных на нее функций. Он, в свою очередь, может быть разделен на функциональный контроль и контроль качества продукции. Функциональный контроль может преследовать цель либо только проверки работоспособности (отсутствия неисправностей) системы, либо, кроме того, установления места и причины неисправности (диагностический контроль). Контроль качества продукции в нашем случае как раз и является контролем достоверности информации как одним из важнейших показателей качества продукции выпускаемой ИС.

Генезисный контроль проводится для выяснения технического состояния системы в прошлые моменты времени с целью определения причин сбоев и отказов системы, имевших место ранее, сбора статистических данных об ошибках, их характере, величине и последствиях (экономических потерях) этих ошибок для ИС.

Классификация методов контроля достоверности по уровню исследования

По уровню исследования информации контроль может быть синтаксический, семантический и прагматический.

Синтаксический контроль — это, по существу, контроль достоверности данных, не затрагивающий содержательного, смыслового аспекта информации. Предметом синтаксического контроля являются отдельные символы, реквизиты, показатели: допустимость их наличия, допустимость их кодовой структуры, взаимных сочетаний и порядка следования.

Семантический контроль оценивает смысловое содержание информации, ее логичность, непротиворечивость, согласованность, диапазон возможных значений параметров, отражаемых информацией, динамику их изменения.

Прагматический контроль определяет потребительную стоимость (полезность, ценность) информации для управления, своевременность и актуальность информации, ее полноту и доступность.

Классификация методов контроля достоверности по способу реализации

По способу реализации контроль может быть организационным, программным, аппаратным и комбинированным.

Организационный контроль достоверности является одним из основных в ИС. Он представляет собой комплекс мероприятий, предназначенных для выявления ошибок на всех этапах участия эргатического звена в работе системы, причем обязательным элементом этих мероприятий является человек или коллектив людей.

Программный контроль основан на использовании специальных программ и логических методов проверки достоверности информации или правильности работы отдельных компонентов системы и всей системы в целом. Программный контроль, в свою очередь, подразделяется на программно-логический, алгоритмический и тестовый.

Программно-логический контроль базируется на использовании синтаксической или семантической избыточности; алгоритмический контроль использует как основу вспомогательный усеченный алгоритм преобразования информации, логически связанный с основным рабочим алгоритмом (тестовый контроль был рассмотрен чуть выше).

Аппаратный контроль реализуется посредством специально встроенных в систему дополнительных технических схем. Этот вид контроля также подразделяется на непрерывный и оперативный (аппаратно-логический) контроль достоверности, а также непрерывный контроль работоспособности.

Непрерывный контроль достоверности функционирует непрерывно в процессе работы системы параллельно с процедурами основного технологического процесса преобразования информации. Во время оперативного (аппаратно-логического) контроля достоверности выполнение основных технологических операций над информацией приостанавливается. Непрерывный контроль работоспособности — это уже не контроль достоверности информации, а контроль значений параметров компонентов системы с помощью встроенных в них датчиков.

Классификация методов контроля достоверности по степени выявления и коррекции ошибок

По степени выявления и коррекции ошибок контроль делится на:

- обнаруживающий, фиксирующий только сам факт наличия или отсутствия ошибки;
- локализирующий, позволяющий определить как факт наличия, так и место ошибки (например символ, реквизит и т. д.);
- исправляющий, выполняющий функции и обнаружения, и локализации, и исправления ошибки.

Наиболее эффективными и перспективными методами контроля достоверности информации являются методы, использующие корректирующие коды с обнаружением и исправлением ошибок. При относительно небольшой избыточности (по сравнению с методами верификации, повторного преобразования и т. п.) эти методы имеют высокую корректирующую способность.

В технических системах корректирующие коды получили чрезвычайно широкое применение: в настоящее время, вероятно, нет ни одной эффективно функционирующей сложной технической информационной системы, где бы многократно не использовались эти коды. В современных компьютерах, например, на основе кодов с обнаружением и кодов с автоматическим исправлением ошибок строится весьма разветвленный контроль достоверности многих блоков (ранее уже упоминалось об использовании в накопителях CD и DVD корректирующих кодов Рида-Соломона, в дисковых массивах RAID — циклических кодов с исправлением ошибок, в технологии SMART и в модулях оперативной памяти — кодов Хэмминга с исправлением ошибок и т. д.).

В эрготехнических компонентах ИС чаще используются обнаруживающие ошибки коды, а коды с автоматическим исправлением ошибок широкого использования пока не нашли, вероятно, в связи с большей сложностью недвоичного исполнения этих кодов и малого знакомства с ними разработчиков и пользователей информационных систем.

Безопасность информационных систем

Безопасность информационной системы — свойство, заключающееся в способности системы обеспечить конфиденциальность и целостность информации, то есть защиту информации от несанкционированного доступа, обращенного на ее раскрытие, изменение или разрушение.

Информационную безопасность часто указывают среди основных информационных проблем XXI века. Действительно, вопросы хищения информации, ее сознательного искажения и уничтожения часто приводят к трагическим для пострадавшей стороны последствиям, ведущим к разорению и банкротству фирм, к человеческим жертвам, наконец. Достаточно в качестве примера привести мировую трагедию,

приведшую к жертвам нескольких тысяч людей — атаку террористов на Всемирный торговый центр в Нью-Йорке и Министерство обороны США в Вашингтоне. Подобный террористический акт был бы невозможен, если бы террористы не вывели предварительно из строя компьютерную систему управления безопасностью страны, то есть не разрушили бы систему информационного обеспечения безопасности. А тысячи коммерческих компьютерных преступлений, приводящих к потерям сотен миллионов долларов, а моральные потери, связанные с хищением.

Ход лабораторной работы:

Произвести расчет надежности информационной системы для заданной предметной области.

1. Разработка программы учета ресурсов для проекта информационной системы класса MRP.
2. Разработка проекта информационной системы «Служба занятости».
3. Разработка проекта информационной системы «Фондовый рынок».
4. Разработка проекта информационной системы «Кредитный эксперт».
5. Разработка программы учета клиентов для проекта информационной системы класса CRM.
6. Разработка программы оперативного учета поступления товара на склад.
7. Разработка проекта информационной системы «Недвижимость».
8. Разработка проекта информационной системы «Каталог товаров».
9. Разработка проекта информационной системы «Зарплата».
10. Разработка проекта информационной системы «Поставщики предприятия».
11. Разработка проекта информационной системы инвентаризации аппаратных средств.
12. Разработка проекта информационной системы учета программных средств предприятия.
13. Разработка проекта информационной системы тестирования знаний.
14. Разработка проекта информационной системы «Абитуриент».
15. Разработка проекта информационной системы статистической обработки данных.
16. Разработка проекта информационной системы «Библиотека».
17. Разработка программы учета данных для проекта информационной системы «Реклама».
18. Разработка проекта информационной системы учета успеваемости студентов.
19. Разработка проекта информационной системы «Выпускник вуза».
20. Разработка проекта информационной системы «Отдел кадров».
21. Разработка программы оперативного учета продаж торгового предприятия.

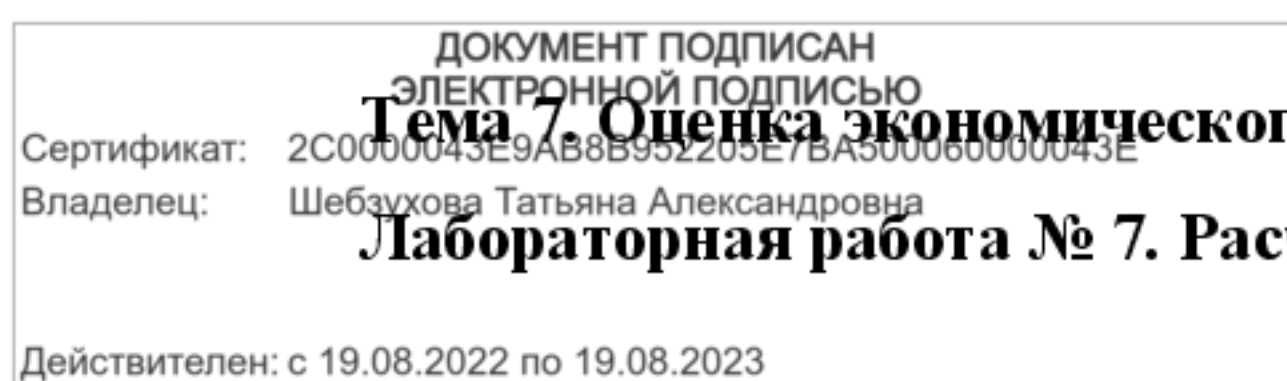
Вопросы для обсуждения:

Стандарты качества и надежности информационной системы.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: отчет к лабораторной работе.



Тема 7. Оценка экономического эффекта внедрения информационной системы

Лабораторная работа № 7. Расчеты экономических показателей

Цель и содержание работы: изучить возможности программы ArgoUML и ее установку.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Возможности ARGOUML

Теоретическое обоснование

Оценка эффективности информационных систем: понятие эффективности, современные методы оценки

В последнее время особую роль для организации играет эффективное использование существующих у нее информационных ресурсов. В этом случае ключевое значение получает информационная инфраструктура организации, в которой обычно выделяют техническое, программное и организационное обеспечение.

Техническое обеспечение включает в себя используемые в организации вычислительные машины, вычислительные сети и периферийное оборудование. Процесс выбора того или иного технического обеспечения в зависимости от потребности организации достаточно формализован и может быть решен силами самой организации при консультациях с поставщиками техники, а также через заказ у соответствующей проектной организации.

В то же время, вопросы выбора программного обеспечения до сих пор не имеют подобной проработки. В основном это связано с тем, что такие исследования проводились либо компаниями, выпускающими программные продукты, в том числе с целью рекламы своей продукции, либо сторонниками свободного ПО, которые также заинтересованы в продвижении конкретных продуктов. Соответственно, независимого всестороннего исследования проведено не было. Так как эффективное использование ПО является определяющим фактором использования ИТ-инфраструктуры, то задача выбора и оценки эффективности программного обеспечения в зависимости от особенностей конкретной организации является актуальной. Более того, без решения этой задачи невозможно создать эффективное организационное обеспечение ИТ-инфраструктуры, так как практически не существует специалистов, одновременно владеющих профессиональными навыками управления проприетарным и свободным ПО различных производителей. Даже если такой специалист есть, содержать его нерентабельно вследствие достаточно высокой заработной платы.

Одной из причин отсутствия универсальной методики выбора ПО является большое разнообразие в формах, структурах, задачах существующих организаций, каждая из них по-своему уникальна, что усложняет формализацию данной задачи.

Одной из проблем определения эффективности общесистемного и офисного программного обеспечения (ПО) является выбор методики оценки. В классической литературе, посвященной вопросу оценки эффективности, она рассчитывается по формуле:

$$\text{Эффективность} = \frac{\text{Эффект}}{\text{Затраты}}$$

Затраты - совокупные затраты на приобретение, установку и конфигурирование, сопровождение и поддержку, а также затраты связанные с простоем оборудования во время технического обслуживания или устранения неисправностей.

Эффект - эффект, достигаемый при внедрении ПО. Однако из-за специфики использования общесистемного и офисного ПО определить прямой эффект от их внедрения (во временных или финансовых показателях) затруднительно. Вследствие этого возникает задача выбора метода оценки, все множество которых можно разделить на:

1. *Затратные методы.* Оценка производится не на основе измерения конечного продукта или результата, а на основе затраченных ресурсов или сил.

2. *Методы оценки прямого результата.* Методика оценивает прямой измеримый результат, например, снижение стоимости владения, повышение функциональности системы, снижение трудозатрат или появление побочного продукта основного трудопроизводства.
3. *Методы, основанные на оценке идеальности процесса.* Такие методики базируются на статических или динамических сравнительных алгоритмах. Базовым показателем выбирается объект рассматриваемой системы, тогда идеальной считается информационная система с лучшими для отрасли показателями затрат на единицу выхода. Популярны также подходы на базе сравнения с альтернативным решением.
4. *Квалиметрические подходы.* Такие методики комплексно рассматривают информационную систему, организуют ее измерение и обрабатывают полученные результаты статистическими, социологическими и/или экспертными методами.

Затратные методы оценки

Котловый метод. Метод основан на определении соотношения объемов вложений в программное обеспечение, включая внедрение и сопровождение, с размерами предприятия и направлениями его бизнеса. Часто данное соотношение задается в виде максимально-допустимого объема вложений по отношению к годовому обороту компании, например не более 1% для небольших компаний и не более 3% для крупных.

Метод функциональной точки. Данный метод используется для приблизительной оценки стоимости создания и внедрения информационной системы (ИС) в зависимости от требований пользователя. Каждое такое требование оценивается как по шкале трудности (легкие, средние и трудные), так и по шкале важности для пользователя. Требования представляются в виде вектора (функциональной точки) в многомерном пространстве. Далее в соответствии с гипотезой «компактности» предполагается, что чем ближе функциональные точки проектов друг к другу в пространстве требований, тем их параметры, включая и эффективность, более схожи. Соответственно в базе ранее внедренных проектов находится такой, чья функциональная точка ближе всего находится к проектируемой ИС, и предполагается, что их эффективности максимально близки.

Total cost of ownership (ТСО - совокупная стоимость владения). Данный метод предполагает количественную оценку на внедрение и сопровождение программного обеспечения, рассчитываемую по формуле:

m

$$Z_{\text{ИТ}} = Z_{\text{Факт}} + E \cdot \int_0^T Z_{\text{ОЦ}} \cdot e^{-E \cdot t} dt$$

$t=T$

где $Z_{\text{ИТ}}$ - оценка интегрированных затрат по проекту в момент t ,
 E - норма дисконтирования, отражающая временной характер финансовых ресурсов, $Z_{\text{Факт}}$ - дисконтированная сумма фактически произведенных

интегральных затрат на момент t , T - период жизненного цикла системы, $Z_{\text{ОЦ}}$ - оценка интегральных затрат на проект в периоде t .

Модель ТСО позволяет разобраться в структуре расходов, связанных с ИС, и открывает широкие перспективы для их сокращения, также способствует выявлению текущих проблем, обеспечивает постоянную обратную связь в управлении затратами.

Методы оценки прямого результата

Потребительский индекс (Customer index). Этот метод предполагает оценку результатов внедрения ПО в виде совокупности индексов, отражающих положительные изменения в работе компании (увеличение доходов, снижение затрат, увеличение оборотов, увеличение клиентской базы и т.п.).

Applied information economics (AIE - прикладная информационная экономика) - методика аналогична потребителю индексу, но в отличие от нее также предполагает

оценку различных субъективных показателей, например, простота работы с системой, удовлетворенность клиентов и т.п.

Economic value sourced (EVS - источник экономической стоимости). Представляет собой оценку того, какую пользу ПО приносит компании при его использовании, оценивается по четырем показателям: увеличение доходов, повышение производительности труда, сокращение времени выпуска продуктов, снижение рисков.

Economic value added (EVA - экономическая добавленная стоимость). Данная методика предполагает определение эффекта как фактическую прибыль от использования ПО, которая равна чистой операционной прибыли за минусом стоимости капитала. Применительно к ИТ проектам EVA означает, что:

- > при использовании капитала в ИТ проектах, необходимо учитывать его стоимость, за него необходимо платить также, как и за труд работников;
- > предполагается, что ИТ-специалисты продают свои услуги другим подразделениям по рыночным расценкам.

Это позволяет рассматривать ИТ как центр прибыли, а не затрат, при этом четко отображая, как увеличиваются доходы.

Методики, основанные на идеальности процесса

Данные методы основаны на сравнении результатов внедрения ПО с уже существующими хорошими (идеальными) примерами. И предполагается, чем ближе мы приближаемся к этим примерам, тем выше эффективность внедряемого ПО. К таким методам относятся:

Среднеотраслевые результаты. В этом случае оценка результативности внедрения ПО проводится по сравнению со средними отраслевыми результатами. Эти результаты обычно приводятся в открытых публикациях и маркетинговых материалах.

Gartner Measurement (Гартнер-измерение). Согласно этому методу эффективность определяет, насколько данная информационная система соответствует нуждам пользователя. При этом ориентирование идет не только на внутренние возможности системы, но и на субъективное мнение клиентов и объективные данные различных вариантов внедрения. Для этого качественно оцениваются такие критерии как время, затраченное на настройку системы, реализованные функциональные возможности, среднее число пользователей на один сервер, среднее и пиковое число транзакций в единицу времени, стоимость одной транзакции, среднее и пиковое время отклика системы, используемые методы обучения, стоимость инфраструктуры информационной системы на одного пользователя. На основе такого исследования оценивается конкретный вариант внедрения, при этом он сравнивается с другими (ранее внедренными) и на основе анализа даются рекомендации по улучшению работы информационной системы, подбору оптимальной конфигурации ПО, по наиболее эффективным для данного клиента методикам обучения, по интеграции информационных систем с другими системами заказчика.

Return of investment (ROI - возвратность инвестиций). Суть методики заключается в выборе для компании типового проекта, оптимального по показателю сроков возврата инвестиций в ПО.

Квалиметрические методы

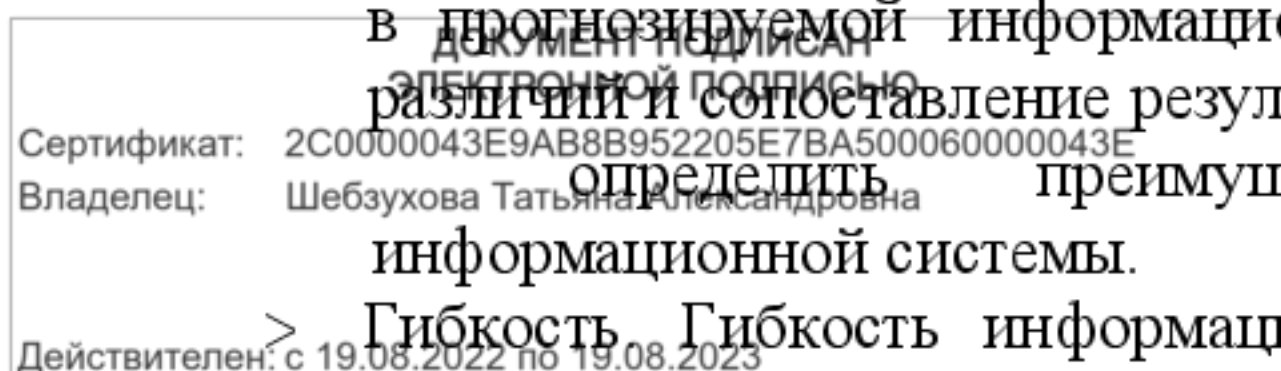
Total economic impact (TEI - модель совокупного экономического эффекта). В качестве затратной компоненты данного метода используется модель ТСО, а эффект рассчитывается на основе следующих факторов:

- > Преимущества. Сравнение вариантов организации труда существующей и

в прогнозируемой информационной системе (как было - как будет). Оценка различий и сопоставление результатов с целями проекта позволяет

определить преимущества или недостатки новой информационной системы.

- > Гибкость. Гибкость информационной системы оценивается с точки зрения ее



расширяемости, а также ее адаптируемости к новым условиям. Одним из гарантов гибкости является использование стандартизированных и унифицированных решений, а также продуманная архитектура информационной системы.

- > Риск. Подразумевается вероятность финансовых потерь при инвестировании в ИТ.

Balanced scorecard (BSC - сбалансированная система показателей). Это система стратегического управления организацией на основе измерения и оценки ее эффективности через использование комплексной функции, включающей набор показателей, учитывающих все аспекты деятельности компании (финансовые, маркетинговые и т.д.). К таким показателям обычно относят:

- > критические факторы успеха (Critical Factors of Success, CFS) - стратегические показатели: финансы, клиенты, внутренние бизнеспроцессы, обучение и рост;
- > ключевые показатели эффективности (Key performance indicators, KPI), включая достигнутые результаты деятельности компании.

Состав и количество сбалансированных показателей определяются исходя из специфики каждой компании.

Выбор метода оценки общесистемного и офисного ПО

При выборе метода оценки эффективности общесистемного и офисного ПО необходимо учитывать следующие факторы:

- > оценка как эффекта, так и затратной компоненты эффективности; возможность определения эффекта применительно к общесистемному и офисному ПО; (возможность оценки финансовых и временных показателей, таких как производительность труда, снижение себестоимости продукции и т.п.)
- > возможность определения показателей без проведения глубокого обследования бизнес-процессов организации (данное обследование является очень затратным и трудоемким. И как правило необходимо лишь для специализированного программного обеспечения);
- > универсальность методики - определяется универсальностью параметров и силе их влияния (при изменении) на алгоритм расчета.

Результаты возможности применения различных методов приведены в таблице.

Сравнение	методов	оценки	эффективности	внедрения
общесистемного и офисного ПО				

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шибзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Таблица 1.

Метод	Оценка эффекта и затрат	Определение эффекта для общесистемного и офисного ПО	Необходимость глубокого обследования организации	Универсальность
<i>Котловой метод</i>	затраты	не считается	не требуется	универсален
<i>Метод функциональной точки</i>	эффект, затраты	применим	не требуется	не универсален
<i>ТСО</i>	затраты	не считается	не требуется	универсален
<i>Потребительский индекс</i>	эффект	не применим	требуется	не универсален
<i>AIE</i>	эффект	применим	не требуется	универсален
<i>EVS</i>	эффект	не применим	требуется	не универсален
<i>EVA</i>	эффект, затраты	применим	требуется	универсален
<i>Среднеотраслевые результаты</i>	эффект	не применим	не требуется	универсален
<i>Gartner Measurement</i>	эффект, затраты	применим	не требуется	универсален
<i>Return investment</i>	эффект, затраты	не применим	не требуется	универсален
<i>TEI</i>	эффект, затраты (ТСО)	применим	не требуется	универсален
<i>BSC</i>	эффект, затраты	применим	требуется	универсален

Выводы

Только два метода оценки соответствуют всем перечисленным факторам: это Gartner Measurement и TEI.

Особенность использования метода Gartner Measurement для оценки эффективности, является необходимость в большой и подробной базе данных (БД) ранее проведенных инсталляций. Что делает его использование весьма затруднительным.

Что касается метода TEI, то его использование для оценки эффективности не нуждается в наличии большой БД, а использование модели ТСО в качестве затратной компоненты данного метода позволяет полностью разобраться в структуре расходов связанных с ПО. Эффект рассчитывается на основе дополнительных факторов, которые

позволяют определить преимущества или недостатки новой информационной системы, оценить гибкость новой ИС, а так же учесть возможные финансовые риски при внедрении новой ИС. Все это позволяет получить достаточно точную оценку и делает метод ТЕІ оптимальным для расчета эффективности общесистемного и офисного ПО.

Ход лабораторной работы:

Произвести расчеты эффективности внедрения информационной системы для заданной предметной области.

1. Разработка программы учета ресурсов для проекта информационной системы класса MRP.
2. Разработка проекта информационной системы «Служба занятости».
3. Разработка проекта информационной системы «Фондовый рынок».
4. Разработка проекта информационной системы «Кредитный эксперт».
5. Разработка программы учета клиентов для проекта информационной системы класса CRM.
6. Разработка программы оперативного учета поступления товара на склад.
7. Разработка проекта информационной системы «Недвижимость».
8. Разработка проекта информационной системы «Каталог товаров».
9. Разработка проекта информационной системы «Зарплата».
10. Разработка проекта информационной системы «Поставщики предприятия».
11. Разработка проекта информационной системы инвентаризации аппаратных средств.
12. Разработка проекта информационной системы учета программных средств предприятия.
13. Разработка проекта информационной системы тестирования знаний.
14. Разработка проекта информационной системы «Абитуриент».
15. Разработка проекта информационной системы статистической обработки данных.
16. Разработка проекта информационной системы «Библиотека».
17. Разработка программы учета данных для проекта информационной системы «Реклама».
18. Разработка проекта информационной системы учета успеваемости студентов.
19. Разработка проекта информационной системы «Выпускник вуза».
20. Разработка проекта информационной системы «Отдел кадров».
21. Разработка программы оперативного учета продаж торгового предприятия.

Вопросы для обсуждения:

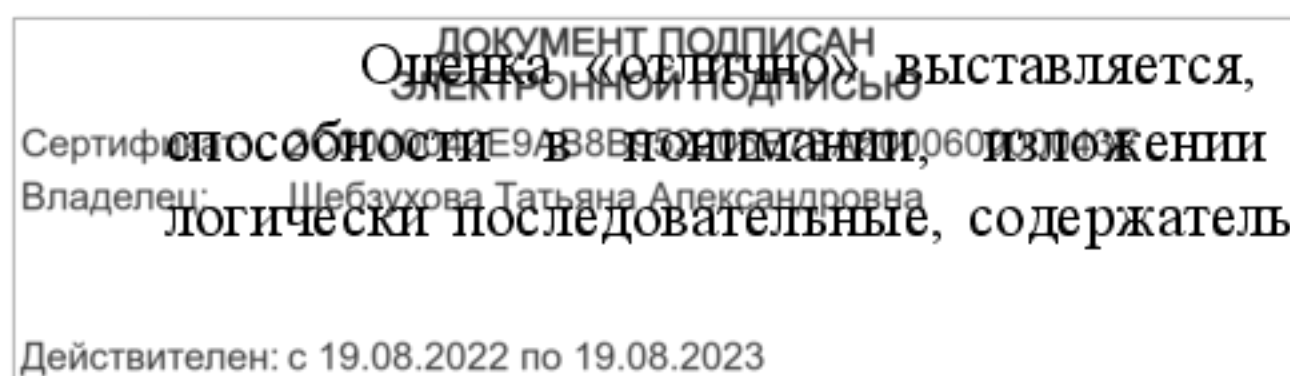
Методы оценки экономического эффекта внедрения информационной системы

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: отчет к лабораторной работе.

3. КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ



Оценка «отлично» выставляется, если глубокие, исчерпывающие знания и творческие способности в понимании, изложении и использовании учебно-программного материала; логически последовательные, содержательные, полные, правильные и конкретные ответы на все

поставленные вопросы и дополнительные вопросы преподавателя, свободное владение основной и дополнительной литературой, рекомендованной учебной программой.

Оценка «хорошо» выставляется, если твердые и достаточно полные знания всего программного материала, правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном устранении замечаний по отдельным вопросам; достаточное владение литературой, рекомендованной учебной программой.

Оценка «удовлетворительно» выставляется, если твердые знания и понимание основного программного материала; правильные, без грубых ошибок ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточное владение литературой, рекомендованной учебной программой.

Оценка «неудовлетворительно» выставляется студенту, если неправильные ответы на основные вопросы, допущены грубые ошибки в ответах, непонимание сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы.

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

Текущая аттестация студентов проводится преподавателями, ведущими лабораторные занятия по дисциплине, в следующих формах: отчет письменный по заданию преподавателя, собеседование, контрольная работа.

Допуск к лабораторным работам происходит при наличии у студентов печатного варианта отчета. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя.

Максимальное количество баллов студент получает, если оформление отчета соответствует установленным требованиям, а отчет полностью раскрывает суть работы. Основанием для снижения оценки являются:

- частично не соответствует установленным требованиям;
- в отчете не полностью раскрывает суть работы.

Отчет может быть отправлен на доработку в следующих случаях:

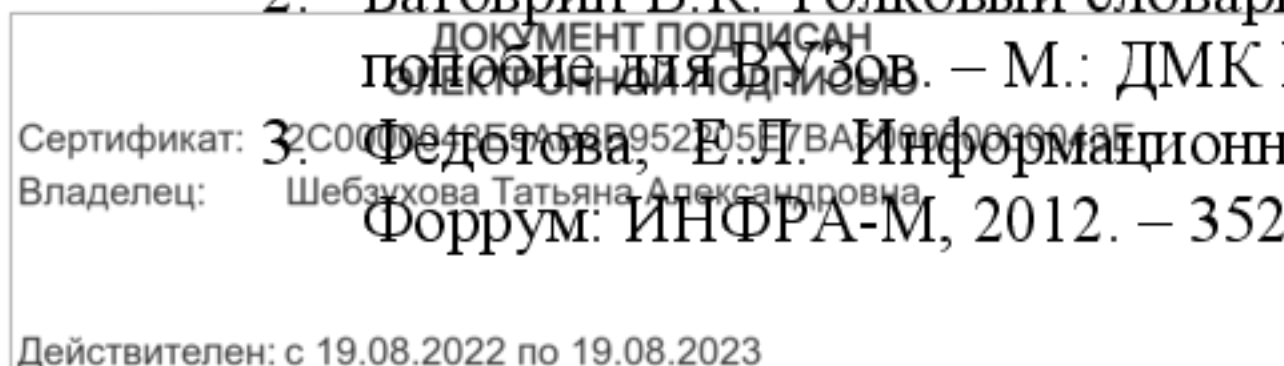
- полностью не соответствует установленным требованиям;
- не раскрыта суть работы.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Рекомендуемая литература

5.1.1. Основная литература

1. Информационные системы: учебник для вузов/ Ю.С. Избачков и др. - 3-е изд. - СПб.: Питер, 2013.- 258 с.
2. Батоврин В.К. Толковый словарь по системной и программной инженерии: учебное пособие для вузов. – М.: ДМК Пресс, 2012. – 280 с.
3. Федотова, Е.Н. Информационные технологии и системы: учеб. пособие. – М.: Форум: ИНФРА-М, 2012. – 352 с.



5.1.2. Дополнительная литература:

1. Корпоративные информационные системы управления: учебник/ Н.М. Адбикеев и др.; ред.: Н.М. Адбикеев, О.В. Китова- М.: ИНФРА-М, 2011.
2. Информационные системы в экономике: учеб. пособие/ ред. Д. В. Чистов. - М.: ИНФРА-М, 2011.
3. Исаев Г.Н. Информационные технологии: учеб. пособие/ Г. Н. Исаев. - М. Омега-Л, 2012.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОРГАНИЗАЦИИ
САМОСТОЯТЕЛЬНОЙ РАБОТЫ
ПО ДИСЦИПЛИНЕ
УЧЕБНЫЙ ПРОФЕССИОНАЛЬНЫЙ ПРОЕКТ**

Направление подготовки	09.04.02
Направленность (профиль)	Информационные системы и технологии «Технологии работы с данными и знаниями, анализ информации»
Квалификация выпускника	Магистр

Пятигорск, 2023

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

СОДЕРЖАНИЕ

Введение.....	63
1. Общая характеристика самостоятельной работы студента.....	63
2. Технологическая карта самостоятельной работы студента.....	64
3. Самостоятельное изучение тем лекций.....	64
4. Паспорт фонда оценочных средств для проверки самостоятельной работы.....	66
4.1 Содержание самостоятельной работы.....	66
4.2 Комплект вопросов для собеседования.....	176
4.3 Темы индивидуальных заданий для письменного отчета.....	177
4.4 Комплект заданий для контрольных работ.....	178
5. Учебно-методическое и информационное обеспечение дисциплины.....	180

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Введение

Целью освоения дисциплины «Учебный профессиональный проект» является формирование набора общекультурных компетенций будущего магистра по направлению подготовки 09.04.02 «Информационные системы и технологии», для решения прикладных задач в рамках направленности (профиля) «Технологии работы с данными и знаниями, анализ информации».

Задачи освоения дисциплины: изучение основных понятий системной инженерии, освоение методов и инструментов системной инженерии.

1. Общая характеристика самостоятельной работы студента

Основными видами учебной работы по достижению результат овосвоения дисциплины являются лекции, лабораторные работы, самостоятельная работа студентов (СРС).

На лекциях раскрываются основные положения и понятия курса, формируются знания в области управления IT-проектами. На лабораторных занятиях формируются умения и навыки, необходимые для решения задач профессиональной деятельности.

Приступая к изучению учебной дисциплины, необходимо ознакомиться с учебной программой, получить в библиотеке рекомендованные учебные пособия, а также получить у ведущего преподавателя в электронном виде конспекты лекций, методические рекомендации к лабораторным работам, завести отдельные тетради для конспектирования лекций, выполнения лабораторных работ.

Для изучения дисциплины предлагается список основной и дополнительной литературы. Основная литература предназначена для обязательного изучения, дополнительная – поможет более глубоко освоить отдельные вопросы, подготовить исследовательские задания и выполнить задания для самостоятельной работы.

В ходе лекционных занятий студент обязан осуществлять конспектирование учебного материала, особое внимание, обращая на категории, формулировки, раскрывающие содержание тех или иных понятий, процессов, эффектов. В рабочих конспектах желательно оставлять поля, на которых следует делать пометки из рекомендованной литературы, дополнять материал прослушанной лекции, формулировать научные выводы и практические рекомендации. Студент имеет право задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Самостоятельная работа студентов над материалом учебной дисциплины является неотъемлемой частью учебного процесса и должна предполагать углубление знания учебного материала, излагаемого на аудиторных занятиях, и приобретение дополнительных знаний по отдельным вопросам самостоятельно.

Основными видами самостоятельной работы студентов по учебной дисциплине являются:

- самостоятельное изучение учебного материала по заданным темам;
- подготовка и оформление отчета по выполненной лабораторной работе и подготовка к собеседованию по результатам работы.

Основными методами самостоятельной работы студентов являются:

1) для овладения знаниями:

- чтение текста (конспекта лекций, учебника, дополнительной литературы) и конспектирование текста;
- поиск информации в Интернет;

2) для закрепления и систематизации знаний:

- работа с конспектом лекции (обработка текста);
- повторная работа над учебным материалом (учебника, дополнительной литературы, слайдами);
- ответы на контрольные вопросы;

3) для формирования умений:

- подготовка к лабораторным работам;
- решение разноуровневых задач лабораторных работ.
- подготовка отчетов по лабораторным занятиям;

В случае пропуска учебного занятия студент может воспользоваться содержанием различных блоков учебно-методического комплекса (лекции, практические занятия, контрольные вопросы) для самоподготовки и освоения темы. Для самоконтроля необходимо использовать вопросы и задания, предлагаемые к лабораторным работам.

2. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций, индикатора (ов)	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе		
			СРС	Контактная работа с преподавателем	Всего
2 семестр					
ПК-1 (ИД 1 ПК-1, ИД 2 ПК-1, ИД 3 ПК-1) ПК-6 (ИД 1 ПК-6, ИД 2 ПК-6, ИД 3 ПК-6)	Подготовка к лекциям	Собеседование	0,27	0,03	0,3
ПК-10 (ИД 1 ПК-10, ИД 2 ПК-10, ИД 3 ПК-10) ПК-12 (ИД 1 ПК-12, ИД 2 ПК-12, ИД 3 ПК-12), ПК-13 (ИД 1 ПК-13, ИД 2 ПК-13, ИД 3 ПК-13)	Самостоятельное изучение литературы	Собеседование	74,61	8,29	82,9

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

ПК-7 (ИД 1 ПК-7, ИД 2 ПК-7, ИД 3 ПК-7), ПК-8 (ИД 1 ПК-8, ИД 2 ПК-8, ИД 3 ПК-8), ПК-9 (ИД 1 ПК-9, ИД 2 ПК-9, ИД 3 ПК-9),	Подготовка и выполнение лабораторных работ	Отчет письменный	2,52	0,28	2,8
ПК-7 (ИД 1 ПК-7, ИД 2 ПК-7, ИД 3 ПК-7), ПК-8 (ИД 1 ПК-8, ИД 2 ПК-8, ИД 3 ПК-8), ПК-9 (ИД 1 ПК-9, ИД 2 ПК-9, ИД 3 ПК-9),	Выполнение контрольной работы	Контрольная работа	9	1	10
Итого за 2 семестр			86,4	9,6	96

3. Самостоятельное изучение тем лекций

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации:

№ п/п	Темы для самостоятельного изучения	Рекомендуемые источники информации (№ источника)			
		Основная	Дополнительная	Методическая	Интернет-ресурсы
	Раздел 1. Методология и инструментарий проектирования информационных систем				
1	Тема 2. Методы объектно-ориентированного и функционального моделирования информационных систем	1-3	1-3	1-3	1-3
	Раздел 2. Реализация этапов жизненного цикла проекта информационной системы				
2	Тема 4. Разработка прототипа информационной системы. Тестирование взаимодействия	1-3	1-3	1-3	1-3

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
 Владелец: Шебухова Татьяна Александровна
 Действителен с 19.06.2024 по 19.06.2023

типовых решений. Типовое проектное содержание операций типового элемента

пакеты прикладных программ (ППП) как основа ТПР. Адаптация типовой ИС. Методы и средства прототипного проектирования ИС.

Каноническое проектирование ИС

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

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

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

Стадия 1. Формирование требований к ИС.

На начальной стадии проектирования выделяют следующие этапы работ:

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

Стадия 2. Разработка концепции ИС.

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

Стадия 3. Техническое задание.

- разработка и утверждение технического задания на создание ИС.

Стадия 4. Эскизный проект.

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

Стадия 5. Технический проект.

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

Стадия 6. Рабочая документация.

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

Стадия 7. Ввод в действие.

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

Стадия 8. Сопровождение ИС.

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

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

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

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

Основная задача первого этапа обследования - оценка реального объема проекта, его целей и задач на основе выявленных функций и информационных элементов автоматизируемого объекта высокого уровня [8]. Эти задачи могут быть реализованы или заказчиком ИС самостоятельно, или с привлечением консалтинговых организаций. Этап предполагает тесное взаимодействие с основными потенциальными пользователями системы и бизнес-экспертами. Основная задача взаимодействия - получить полное и однозначное понимание требований заказчика. Как правило, нужная информация может быть получена в результате интервью, бесед или семинаров с руководством, экспертами и пользователями.

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

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

Ориентировочное содержание этого документа:

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

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

- инструктивно-методические и директивные материалы, на основании которых определяются состав подсистем и перечень задач;
- возможности применения новых методов решения задач.

Аналитики собирают и фиксируют информацию в двух взаимосвязанных формах:

- функции - информация о событиях и процессах, которые происходят в бизнесе;

- сущности - информация о вещах, имеющих значение для организации и о которых что-то известно.

При изучении каждой функциональной задачи управления определяются:

- наименование задачи; сроки и периодичность ее решения;
- степень формализуемости задачи;
- источники информации, необходимые для решения задачи;
- показатели и их количественные характеристики;
- порядок корректировки информации;
- действующие алгоритмы расчета показателей и возможные методы контроля;
- действующие средства сбора, передачи и обработки информации;
- действующие средства связи;
- принятая точность решения задачи;
- трудоемкость решения задачи;
- действующие формы представления исходных данных и результатов их обработки в виде документов;
- потребители результатной информации по задаче.

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

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

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

На этапе обследования следует классифицировать планируемые функции системы по степени важности. Один из возможных форматов представления такой классификации - MuSCoW [9].

Эта аббревиатура расшифровывается так: Must have - необходимые функции; Should have - желательные функции; Could have - возможные функции; Won't have - отсутствующие функции.

Функции первой категории обеспечивают критичные для успешной работы системы возможности.

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

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

Модели деятельности организации создаются в двух видах:

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

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

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

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

Для автоматизации тестирования следует использовать системы отслеживания ошибок (bug tracking). Это позволяет иметь единое хранилище ошибок, отслеживать их повторное появление, контролировать скорость и эффективность исправления ошибок, видеть наиболее нестабильные компоненты системы, а также поддерживать связь между группой разработчиков и группой тестирования (уведомления об изменениях по e-mail и т.п.). Чем больше проект, тем сильнее потребность в bug tracking.

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

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

При разработке технического задания необходимо решить следующие задачи:

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

Типовые требования к составу и содержанию технического задания приведены в [таблице 3.1](#).

Таблица 3.1. Состав и содержание технического задания (ГОСТ 34.602- 89)

№ п/п	Раздел	Содержание
1	Общие сведения	• полное наименование системы и ее условное обозначение • шифр темы или шифр (номер) договора; • наименование предприятий разработчика и заказчика системы, их реквизиты • перечень документов, на основании которых создается ИС • плановые сроки начала и окончания работ • сведения об источниках и порядке финансирования работ • порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств
2	Назначение и цели создания (развития) системы	• вид автоматизируемой деятельности • перечень объектов, на которых предполагается использование системы

- наименования и требуемые значения технических, технологических, производственно-экономических и др. показателей объекта, которые должны быть достигнуты при внедрении ИС
- 3 Характеристика объектов автоматизации
- краткие сведения об объекте автоматизации
 - сведения об условиях эксплуатации и характеристиках окружающей среды
- 4 Требования к системе
- Требования к системе в целом:
- требования к структуре и функционированию системы (перечень подсистем, уровни иерархии, степень централизации, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы)
 - требования к персоналу (численность пользователей, квалификация, режим работы, порядок подготовки)
 - показатели назначения (степень приспособляемости системы к изменениям процессов управления и значений параметров)
 - требования к надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию и ремонту, защите и сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации
- Требования к функциям (по подсистемам) :
- перечень подлежащих автоматизации задач
 - временной регламент реализации каждой функции
 - требования к качеству реализации каждой функции, к форме представления выходной информации, характеристики точности, достоверности выдачи результатов
 - перечень и критерии отказов
- Требования к видам обеспечения:
- математическому (состав и область применения мат. моделей и методов, типовых и разрабатываемых алгоритмов)
 - информационному (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые классификаторы, СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным документам)
 - лингвистическому (языки программирования, языки взаимодействия пользователей с системой,

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

- системы кодирования, языки ввода- вывода)
- программному (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ)
 - техническому
 - метрологическому
 - организационному (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала)
 - методическому (состав нормативно-технической документации)
- 5 Состав и содержание работ по созданию системы
- перечень стадий и этапов работ
 - сроки исполнения
 - состав организаций — исполнителей работ
 - вид и порядок экспертизы технической документации
 - программа обеспечения надежности
 - программа метрологического обеспечения
- 6 Порядок контроля и приемки системы
- виды, состав, объем и методы испытаний системы
 - общие требования к приемке работ по стадиям
 - статус приемной комиссии
- 7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие
- преобразование входной информации к машиночитаемому виду
 - изменения в объекте автоматизации
 - сроки и порядок комплектования и обучения персонала
- 8 Требования к документированию
- перечень подлежащих разработке документов
 - перечень документов на машинных носителях
- 9 Источники разработки документы и информационные материалы, на основании которых разрабатывается ТЗ и система

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

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

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

- функции ИС;
- функции подсистем, их цели и ожидаемый эффект от внедрения;
- состав комплексов задач и отдельных задач;
- концепция информационной базы и ее укрупненная структура;
- функции системы управления базой данных;
- состав вычислительной системы и других технических средств;
- функции и параметры основных программных средств.

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

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

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

Состав и содержание технического проекта приведены в [таблице 3.2](#).

Таблица 3.2. Содержание технического проекта

№ п/п	Раздел	Содержание
1	Пояснительная записка	• основания для разработки системы • перечень организаций разработчиков • краткая характеристика объекта с указанием основных технико-экономических показателей его функционирования и связей с другими объектами • краткие сведения об основных проектных решениях по функциональной и обеспечивающим частям системы
2	Функциональная и организационная структура системы	• обоснование выделяемых подсистем, их перечень и назначение • перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания • схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы
3	Постановка задач и алгоритмы решения	• организационно-экономическая сущность задачи (наименование, цель решения, краткое содержание, метод, периодичность и время решения задачи, способы сбора и передачи данных, связь задачи с другими задачами, характер использования результатов решения, в которых они используются) • экономико-математическая модель задачи (структурная и развернутая форма представления) • входная оперативная информация (характеристика показателей, диапазон изменения, формы представления) • нормативно-справочная информация (НСИ) (содержание и формы представления) • информация, хранимая для связи с другими задачами • информация, накапливаемая для последующих решений данной задачи • информация по внесению изменений (система внесения изменений и перечень информации, подвергающейся изменениям) • алгоритм решения задачи (последовательность этапов расчета, схема, расчетные формулы) • контрольный пример (набор заполненных данными форм входных документов, условные документы с накапливаемой и хранимой информацией, формы выходных документов, заполненные по результатам решения экономико-технической задачи и в

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

- | | | |
|---|---|--|
| | | соответствии с разработанным алгоритмом расчета) |
| 4 | Организация информационной базы | <ul style="list-style-type: none"> • источники поступления информации и способы ее передачи • совокупность показателей, используемых в системе • состав документов, сроки и периодичность их поступления • основные проектные решения по организации фонда НСИ • состав НСИ, включая перечень реквизитов, их определение, диапазон изменения и перечень документов НСИ • перечень массивов НСИ, их объем, порядок и частота корректировки информации • структура фонда НСИ с описанием связи между его элементами; требования к технологии создания и ведения фонда • методы хранения, поиска, внесения изменений и контроля • определение объемов и потоков информации НСИ • контрольный пример по внесению изменений в НСИ • предложения по унификации документации |
| 5 | Альбом форм документов | |
| 6 | Система математического обеспечения | <ul style="list-style-type: none"> • обоснование структуры математического обеспечения • обоснование выбора системы программирования • перечень стандартных программ |
| 7 | Принцип построения комплекса технических средств | <ul style="list-style-type: none"> • описание и обоснование схемы технологического процесса обработки данных • обоснование и выбор структуры комплекса технических средств и его функциональных групп • обоснование требований к разработке нестандартного оборудования • комплекс мероприятий по обеспечению надежности функционирования технических средств |
| 8 | Расчет экономической эффективности системы | <ul style="list-style-type: none"> • сводная смета затрат, связанных с эксплуатацией систем • расчет годовой экономической эффективности, источниками которой являются оптимизация производственной структуры хозяйства (объединения), снижение себестоимости продукции за счет рационального использования производственных ресурсов и уменьшения потерь, улучшения принимаемых управленческих решений |
| 9 | Мероприятия по подготовке объекта к внедрению системы | <ul style="list-style-type: none"> • перечень организационных мероприятий по совершенствованию бизнес-процессов • перечень работ по внедрению системы, которые необходимо выполнить на стадии рабочего проектирования, с указанием сроков и ответственных лиц |

документов

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

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

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

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

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

Предварительные испытания проводят для определения работоспособности системы и решения вопроса о возможности ее приемки в опытную эксплуатацию. Предварительные испытания следует выполнять после проведения разработчиком отладки и тестирования поставляемых программных и технических средств системы и представления им соответствующих документов об их готовности к испытаниям, а также после ознакомления персонала ИС с эксплуатационной документацией.

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

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

Типовое проектирование ИС

Методы типового проектирования ИС достаточно подробно рассмотрены в литературе [10]. В данном курсе приведены основные определения и представлено задание для разработки проекта ИС методом типового проектирования (кейс "Проектирование ИС предприятия оптовой торговли лекарственными препаратами").

Типовое проектирование ИС предполагает создание системы из готовых типовых элементов. Основопологающим требованием для применения методов типового проектирования является возможность декомпозиции проектируемой ИС на множество составляющих компонентов (подсистем, комплексов задач, программных модулей и т.д.).

Для реализации выделенных компонентов выбираются имеющиеся на рынке типовые проектные решения, которые настраиваются на особенности конкретного предприятия.

Типовое проектное решение (ТПР) - это тиражируемое (пригодное к многократному использованию) проектное решение.

Принятая классификация ТПР основана на уровне декомпозиции системы. Выделяются следующие классы ТПР:

- элементные ТПР - типовые решения по задаче или по отдельному виду обеспечения задачи (информационному, программному, техническому, математическому, организационному);
- подсистемные ТПР - в качестве элементов типизации выступают отдельные подсистемы, разработанные с учетом функциональной полноты и минимизации внешних информационных связей;
- объектные ТПР - типовые отраслевые проекты, которые включают полный набор функциональных и обеспечивающих подсистем ИС.

Каждое типовое решение предполагает наличие, кроме собственно функциональных элементов (программных или аппаратных), документации с детальным описанием ТПР и процедур настройки в соответствии с требованиями разрабатываемой системы.

Основные особенности различных классов ТПР приведены в [таблице 3.3](#).

Таблица 3.3. Достоинства и недостатки ТПР

Класс ТПР	Достоинства	Недостатки
Реализация ТПР		
Элементные ТПР Библиотеки методо- ориентированных программ	• обеспечивается применение модульного подхода к проектированию и документированию ИС	• большие затраты времени на сопряжение разнородных элементов вследствие информационной, программной и технической несовместимости • большие затраты времени на доработку ТПР отдельных элементов
Подсистемные ТПР Пакеты прикладных программ	• достигается высокая степень интеграции элементов ИС • позволяют осуществлять: модульное проектирование; параметрическую настройку программных компонентов на различные объекты управления • обеспечивают: сокращение затрат на проектирование и программирование взаимосвязанных компонентов; хорошее документирование отображаемых процессов обработки информации	• адаптивность ТПР недостаточна с позиции непрерывного инжиниринга деловых процессов • возникают проблемы в комплексировании разных функциональных подсистем, особенно в случае использования решений нескольких производителей программного обеспечения
Объектные ТПР Отраслевые проекты ИС	• комплексирование всех компонентов ИС за счет методологического единства информационной, программной и технической	• проблемы привязки типового проекта к конкретному объекту управления, что вызывает в некоторых случаях даже необходимость изменения организационно-

Документ подписан
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

- | | |
|---|--|
| совместимости
• открытость архитектуры — позволяет устанавливать ТПР на разных программно-технических платформах
• масштабируемость — допускает конфигурацию ИС для переменного числа рабочих мест
• конфигурируемость — позволяет выбирать необходимое подмножество компонентов | экономической структуры
объекта автоматизации |
|---|--|

Для реализации типового проектирования используются два подхода: параметрически-ориентированное и модельно-ориентированное проектирование.

Параметрически-ориентированное проектирование включает следующие этапы: определение критериев оценки пригодности пакетов прикладных программ (ППП) для решения поставленных задач, анализ и оценка доступных ППП по сформулированным критериям, выбор и закупка наиболее подходящего пакета, настройка параметров (доработка) закупленного ППП.

Критерии оценки ППП делятся на следующие группы:

- назначение и возможности пакета;
- отличительные признаки и свойства пакета;
- требования к техническим и программным средствам;
- документация пакета;
- факторы финансового порядка;
- особенности установки пакета;
- особенности эксплуатации пакета;
- помощь поставщика по внедрению и поддержанию пакета;
- оценка качества пакета и опыт его использования;
- перспективы развития пакета.

Внутри каждой группы критериев выделяется некоторое подмножество частных показателей, детализирующих каждый из десяти выделенных аспектов анализа выбираемых ППП. Достаточно полный перечень показателей можно найти в литературе [10].

Числовые значения показателей для конкретных ППП устанавливаются экспертами по выбранной шкале оценок (например, 10-балльной). На их основе формируются групповые оценки и комплексная оценка пакета (путем вычисления средневзвешенных значений). Нормированные взвешивающие коэффициенты также получают экспертным путем.

Модельно-ориентированное проектирование заключается в адаптации состава и характеристик типовой ИС в соответствии с моделью объекта автоматизации.

Технология проектирования в этом случае должна обеспечивать единые средства для работы как с моделью типовой ИС, так и с моделью конкретного предприятия.

Типовая ИС в специальной базе метаданных - репозитории - содержит модель объекта автоматизации, на основе которой осуществляется конфигурирование

программного обеспечения. Таким образом, модельно-ориентированное проектирование ИС предполагает прежде всего, построение модели объекта автоматизации с использованием специального программного инструментария (например, SAP Business Engineering Workbench (BEW), BAAN Enterprise Modeler). Возможно также создание системы на базе типовой модели ИС из репозитория, который поставляется вместе с

программным продуктом и расширяется по мере накопления опыта проектирования информационных систем для различных отраслей и типов производства.

Репозиторий содержит базовую (ссылочную) модель ИС, типовые (референтные) модели определенных классов ИС, модели конкретных ИС предприятий.

Базовая модель ИС в репозитории содержит описание бизнес-функций, бизнес-процессов, бизнес-объектов, бизнес-правил, организационной структуры, которые поддерживаются программными модулями типовой ИС.

Типовые модели описывают конфигурации информационной системы для определенных отраслей или типов производства.

Модель конкретного предприятия строится либо путем выбора фрагментов основной или типовой модели в соответствии со специфическими особенностями предприятия (BAAN Enterprise Modeler), либо путем автоматизированной адаптации этих моделей в результате экспертного опроса (SAP Business Engineering Workbench).

Построенная модель предприятия в виде метаописания хранится в репозитории и при необходимости может быть откорректирована. На основе этой модели автоматически осуществляется конфигурирование и настройка информационной системы.

Бизнес-правила определяют условия корректности совместного применения различных компонентов ИС и используются для поддержания целостности создаваемой системы.

Модель бизнес-функций представляет собой иерархическую декомпозицию функциональной деятельности предприятия (подробное описание см. в разделе "Анализ и моделирование функциональной области внедрения ИС").

Модель бизнес-процессов отражает выполнение работ для функций самого нижнего уровня модели бизнес-функций (подробное описание см. в разделе "Спецификация функциональных требований к ИС"). Для отображения процессов используется модель управления событиями (EPC - Event-driven Process Chain). Именно модель бизнес-процессов позволяет выполнить настройку программных модулей - приложений информационной системы в соответствии с характерными особенностями конкретного предприятия.

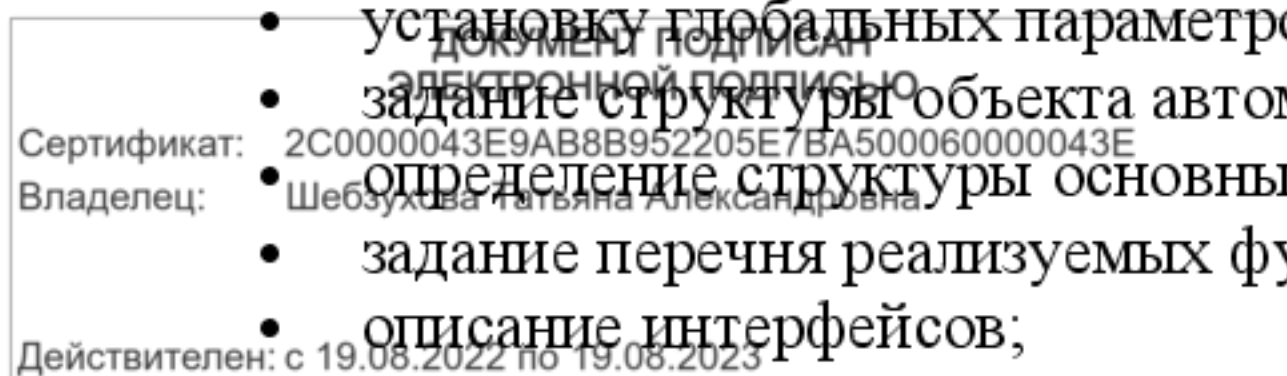
Модели бизнес-объектов используются для интеграции приложений, поддерживающих исполнение различных бизнес-процессов (подробное описание см. в разделе "Этапы проектирования ИС с применением UML").

Модель организационной структуры предприятия представляет собой традиционную иерархическую структуру подчинения подразделений и персонала (подробное описание см. в разделе "Анализ и моделирование функциональной области внедрения ИС").

Внедрение типовой информационной системы начинается с анализа требований к конкретной ИС, которые выявляются на основе результатов предпроектного обследования объекта автоматизации (см. раздел "Анализ и моделирование функциональной области внедрения ИС"). Для оценки соответствия этим требованиям программных продуктов может использоваться описанная выше методика оценки ППП. После выбора программного продукта на базе имеющихся в нем референтных моделей строится предварительная модель ИС, в которой отражаются все особенности реализации ИС для конкретного предприятия. Предварительная модель является основой для выбора типовой модели системы и определения перечня компонентов, которые будут реализованы с использованием других программных средств или потребуют разработки с помощью имеющихся в составе типовой ИС инструментальных средств (например, ABAP в SAP, Tools в BAAN).

Реализация типового проекта предусматривает выполнение следующих операций:

- установку глобальных параметров системы;
- задание структуры объекта автоматизации;
- определение структуры основных данных;
- задание перечня реализуемых функций и процессов;
- описание интерфейсов;



- описание отчетов;
- настройку авторизации доступа;
- настройку системы архивирования.

Каноническое проектирование ИС

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

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

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

Стадия 1. Формирование требований к ИС.

На начальной стадии проектирования выделяют следующие этапы работ:

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

Стадия 2. Разработка концепции ИС.

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

Стадия 3. Техническое задание.

- разработка и утверждение технического задания на создание ИС.

Стадия 4. Эскизный проект.

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

Стадия 5. Технический проект.

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

Стадия 6. Рабочая документация.

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

Стадия 7. Ввод в действие.

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

Стадия 8. Сопровождение ИС.

- выполнение работ в соответствии с гарантийными обязательствами;

- послегарантийное обслуживание.

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

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

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

Основная задача первого этапа обследования - оценка реального объема проекта, его целей и задач на основе выявленных функций и информационных элементов автоматизируемого объекта высокого уровня [8]. Эти задачи могут быть реализованы или заказчиком ИС самостоятельно, или с привлечением консалтинговых организаций. Этап предполагает тесное взаимодействие с основными потенциальными пользователями системы и бизнес-экспертами. Основная задача взаимодействия - получить полное и однозначное понимание требований заказчика. Как правило, нужная информация может быть получена в результате интервью, бесед или семинаров с руководством, экспертами и пользователями.

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

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

Ориентировочное содержание этого документа:

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

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

- инструктивно-методические и директивные материалы, на основании которых определяются состав подсистем и перечень задач;
- возможности применения новых методов решения задач.

Аналитики собирают и фиксируют информацию в двух взаимосвязанных формах:

- функции - информация о событиях и процессах, которые происходят в бизнесе;
- сущности - информация о вещах, имеющих значение для организации и о которых что-то известно.

При изучении каждой функциональной задачи управления определяются:

- наименование задачи; сроки и периодичность ее решения;
- степень формализуемости задачи;
- источники информации, необходимые для решения задачи;
- показатели и их количественные характеристики;
- порядок корректировки информации;
- действующие алгоритмы расчета показателей и возможные методы контроля;
- действующие средства сбора, передачи и обработки информации;
- действующие средства связи;
- принятая точность решения задачи;
- трудоемкость решения задачи;
- действующие формы представления исходных данных и результатов их обработки в виде документов;
- потребители результатной информации по задаче.

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

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

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

На этапе обследования следует классифицировать планируемые функции системы по степени важности. Один из возможных форматов представления такой классификации - MuSCoW [9].

Эта аббревиатура расшифровывается так: Must have - необходимые функции; Should have - желательные функции; Could have - возможные функции; Won't have - отсутствующие функции.

Функции первой категории обеспечивают критичные для успешной работы системы возможности.

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

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

Модели деятельности организации создаются в двух видах:

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

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

- получения сравнительных характеристик предполагаемых к использованию аппаратных платформ, операционных систем, СУБД, иного окружения;

- разработки плана работ по обеспечению надежности информационной системы и ее тестирования.

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

Для автоматизации тестирования следует использовать системы отслеживания ошибок (bug tracking). Это позволяет иметь единое хранилище ошибок, отслеживать их повторное появление, контролировать скорость и эффективность исправления ошибок, видеть наиболее нестабильные компоненты системы, а также поддерживать связь между группой разработчиков и группой тестирования (уведомления об изменениях по e-mail и т.п.). Чем больше проект, тем сильнее потребность в bug tracking.

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

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

При разработке технического задания необходимо решить следующие задачи:

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

Типовые требования к составу и содержанию технического задания приведены в [таблице 3.1](#).

Таблица 3.1. Состав и содержание технического задания (ГОСТ 34.602- 89)

№ п/п	Раздел	Содержание
1	Общие сведения	• полное наименование системы и ее условное обозначение • шифр темы или шифр (номер) договора; • наименование предприятий разработчика и заказчика системы, их реквизиты • перечень документов, на основании которых создается ИС • плановые сроки начала и окончания работ • сведения об источниках и порядке финансирования работ • порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств
2	Назначение и цели создания	• вид автоматизируемой деятельности

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
2
Действителен: с 19.08.2022 по 19.08.2023

(развития) системы	• перечень объектов, на которых предполагается использование системы • наименования и требуемые значения технических, технологических, производственно-экономических и др. показателей объекта, которые должны быть достигнуты при внедрении ИС
3 Характеристика объектов автоматизации	• краткие сведения об объекте автоматизации • сведения об условиях эксплуатации и характеристиках окружающей среды
4 Требования к системе	Требования к системе в целом: • требования к структуре и функционированию системы (перечень подсистем, уровни иерархии, степень централизации, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы) • требования к персоналу (численность пользователей, квалификация, режим работы, порядок подготовки) • показатели назначения (степень приспособляемости системы к изменениям процессов управления и значений параметров) • требования к надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию и ремонту, защите и сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации Требования к функциям (по подсистемам) : • перечень подлежащих автоматизации задач • временной регламент реализации каждой функции • требования к качеству реализации каждой функции, к форме представления выходной информации, характеристики точности, достоверности выдачи результатов • перечень и критерии отказов Требования к видам обеспечения: • математическому (состав и область применения мат. моделей и методов, типовых и разрабатываемых алгоритмов) • информационному (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые классификаторы, СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным документам)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

- лингвистическому (языки программирования, языки взаимодействия пользователей с системой, системы кодирования, языки ввода- вывода)
 - программному (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ)
 - техническому
 - метрологическому
 - организационному (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала)
 - методическому (состав нормативно-технической документации)
- 5 Состав и содержание работ по созданию системы
- перечень стадий и этапов работ
 - сроки исполнения
 - состав организаций — исполнителей работ
 - вид и порядок экспертизы технической документации
 - программа обеспечения надежности
 - программа метрологического обеспечения
- 6 Порядок контроля и приемки системы
- виды, состав, объем и методы испытаний системы
 - общие требования к приемке работ по стадиям
 - статус приемной комиссии
- 7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие
- преобразование входной информации к машиночитаемому виду
 - изменения в объекте автоматизации
 - сроки и порядок комплектования и обучения персонала
- 8 Требования к документированию
- перечень подлежащих разработке документов
 - перечень документов на машинных носителях
- 9 Источники разработки документы и информационные материалы, на основании которых разрабатывается ТЗ и система

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

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

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

- функции ИС;
- функции подсистем, их цели и ожидаемый эффект от внедрения;
- состав комплексов задач и отдельных задач;
- концепция информационной базы и ее укрупненная структура;
- функции системы управления базой данных;
- состав вычислительной системы и других технических средств;
- функции и параметры основных программных средств.

По результатам проделанной работы оформляется, согласовывается и утверждается документация в объеме, необходимом для описания полной совокупности принятых

проектных решений и достаточном для дальнейшего выполнения работ по созданию системы.

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

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

Состав и содержание технического проекта приведены в [таблице 3.2](#).

Таблица 3.2. Содержание технического проекта

№ п/п	Раздел	Содержание
1	Пояснительная записка	• основания для разработки системы • перечень организаций разработчиков • краткая характеристика объекта с указанием основных технико-экономических показателей его функционирования и связей с другими объектами • краткие сведения об основных проектных решениях по функциональной и обеспечивающим частям системы
2	Функциональная и организационная структура системы	• обоснование выделяемых подсистем, их перечень и назначение • перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания • схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы
3	Постановка задач и алгоритмы решения	• организационно-экономическая сущность задачи (наименование, цель решения, краткое содержание, метод, периодичность и время решения задачи, способы сбора и передачи данных, связь задачи с другими задачами, характер использования результатов решения, в которых они используются) • экономико-математическая модель задачи (структурная и развернутая форма представления) • входная оперативная информация (характеристика показателей, диапазон изменения, формы представления) • нормативно-справочная информация (НСИ) (содержание и формы представления) • информация, хранимая для связи с другими задачами • информация, накапливаемая для последующих решений данной задачи • информация по внесению изменений (система внесения изменений и перечень информации, подвергающейся изменениям) • алгоритм решения задачи (последовательность этапов расчета, схема, расчетные формулы) • контрольный пример (набор заполненных данными форм входных документов, условные документы с накапливаемой и хранимой информацией, формы

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

- ДОКУМЕНТ ПОДПИСАН**
- ЭЛЕКТРОННОЙ ПОДПИСЬЮ**
- подготовка объекта к внедрению системы**
- Сертификат: 2019-08-26 ШЕБЗУХОВА ТАТЬЯНА АЛЕКСАНДРОВНА 7B500060000043E
Владелец: Шебзухова Татьяна Александровна
- Действителен: с 19.08.2022 по 19.08.2023

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

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

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

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

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

Методы типового проектирования ИС достаточно подробно рассмотрены в литературе [10]. В данном курсе приведены основные определения и представлено задание для разработки проекта ИС методом типового проектирования (кейс "Проектирование ИС предприятия оптовой торговли лекарственными препаратами").

Типовое проектирование ИС предполагает создание системы из готовых типовых элементов. Основопологающим требованием для применения методов типового проектирования является возможность декомпозиции проектируемой ИС на множество составляющих компонентов (подсистем, комплексов задач, программных модулей и т.д.).

Для реализации выделенных компонентов выбираются имеющиеся на рынке типовые проектные решения, которые настраиваются на особенности конкретного предприятия.

Типовое проектное решение (ТПР)- это тиражируемое (пригодное к многократному использованию) проектное решение.

Принятая классификация ТПР основана на уровне декомпозиции системы. Выделяются следующие классы ТПР:

- элементные ТПР - типовые решения по задаче или по отдельному виду обеспечения задачи (информационному, программному, техническому, математическому, организационному);
- подсистемные ТПР - в качестве элементов типизации выступают отдельные подсистемы, разработанные с учетом функциональной полноты и минимизации внешних информационных связей;
- объектные ТПР - типовые отраслевые проекты, которые включают полный набор функциональных и обеспечивающих подсистем ИС.

Каждое типовое решение предполагает наличие, кроме собственно функциональных элементов (программных или аппаратных), документации с детальным описанием ТПР и процедур настройки в соответствии с требованиями разрабатываемой системы.

Основные особенности различных классов ТПР приведены в [таблице 3.3](#).

Таблица 3.3. Достоинства и недостатки ТПР

Класс ТПР	Достоинства	Недостатки
Реализация ТПР		
Элементные ТПР Библиотеки методо- ориентированных программ	• обеспечивается применение модульного подхода к проектированию и документированию ИС	• большие затраты времени на сопряжение разнородных элементов вследствие информационной, программной и технической несовместимости • большие затраты времени на доработку ТПР отдельных элементов
Подсистемные ТПР Пакеты прикладных программ	• достигается высокая степень интеграции элементов ИС • позволяют осуществлять: модульное проектирование; параметрическую настройку программных компонентов на различные объекты управления • обеспечивают: сокращение затрат на проектирование и программирование взаимосвязанных компонентов; хорошее документирование отображаемых процессов обработки информации	• адаптивность ТПР недостаточна с позиции непрерывного инжиниринга деловых процессов • возникают проблемы в комплексировании разных подсистем, особенно в случае использования решений нескольких производителей программного обеспечения
Объектные ТПР Отраслевые проекты ИС	• комплексирование всех компонентов ИС за счет методологического	• проблемы привязки типового проекта к конкретному объекту

- | | | |
|--------------------------|-------------|--------------------------|
| единства | и | управления, что вызывает |
| информационной, | | в некоторых случаях даже |
| программной | и | необходимость изменения |
| технической | | организационно- |
| совместимости | | экономической структуры |
| • открытость архитектуры | | объекта автоматизации |
| — | позволяет | |
| устанавливать ТПР | на | |
| разных | программно- | |
| технических платформах | | |
| • масштабируемость | — | |
| допускает конфигурацию | | |
| ИС для переменного числа | | |
| рабочих мест | | |
| • конфигурируемость | — | |
| позволяет | выбирать | |
| необходимое | | |
| подмножество | | |
| компонентов | | |

Для реализации типового проектирования используются два подхода: параметрически-ориентированное и модельно-ориентированное проектирование.

Параметрически-ориентированное проектирование включает следующие этапы: определение критериев оценки пригодности пакетов прикладных программ (ППП) для решения поставленных задач, анализ и оценка доступных ППП по сформулированным критериям, выбор и закупка наиболее подходящего пакета, настройка параметров (доработка) закупленного ППП.

Критерии оценки ППП делятся на следующие группы:

- назначение и возможности пакета;
- отличительные признаки и свойства пакета;
- требования к техническим и программным средствам;
- документация пакета;
- факторы финансового порядка;
- особенности установки пакета;
- особенности эксплуатации пакета;
- помощь поставщика по внедрению и поддержанию пакета;
- оценка качества пакета и опыт его использования;
- перспективы развития пакета.

Внутри каждой группы критериев выделяется некоторое подмножество частных показателей, детализирующих каждый из десяти выделенных аспектов анализа выбираемых ППП. Достаточно полный перечень показателей можно найти в литературе [10].

Числовые значения показателей для конкретных ППП устанавливаются экспертами по выбранной шкале оценок (например, 10-балльной). На их основе формируются групповые оценки и комплексная оценка пакета (путем вычисления средневзвешенных значений). Нормированные взвешивающие коэффициенты также получают экспертным путем.

Модельно-ориентированное проектирование заключается в адаптации состава и характеристик типовой ИС в соответствии с моделью объекта автоматизации.

Технология проектирования в этом случае должна обеспечивать единые средства для работы как с моделью типовой ИС, так и с моделью конкретного предприятия.

Типовая ИС в специальной базе метаданных - репозитории - содержит модель объекта автоматизации, на основе которой осуществляется конфигурирование программного обеспечения. Таким образом, модельно-ориентированное проектирование

ИС предполагает, прежде всего, построение модели объекта автоматизации с использованием специального программного инструментария (например, SAP Business Engineering Workbench (BEW), BAAN Enterprise Modeler). Возможно также создание системы на базе типовой модели ИС из репозитория, который поставляется вместе с программным продуктом и расширяется по мере накопления опыта проектирования информационных систем для различных отраслей и типов производства.

Репозиторий содержит базовую (ссылочную) модель ИС, типовые (референтные) модели определенных классов ИС, модели конкретных ИС предприятий.

Базовая модель ИС в репозитории содержит описание бизнес-функций, бизнес-процессов, бизнес-объектов, бизнес-правил, организационной структуры, которые поддерживаются программными модулями типовой ИС.

Типовые модели описывают конфигурации информационной системы для определенных отраслей или типов производства.

Модель конкретного предприятия строится либо путем выбора фрагментов основной или типовой модели в соответствии со специфическими особенностями предприятия (BAAN Enterprise Modeler), либо путем автоматизированной адаптации этих моделей в результате экспертного опроса (SAP Business Engineering Workbench).

Построенная модель предприятия в виде метаописания хранится в репозитории и при необходимости может быть откорректирована. На основе этой модели автоматически осуществляется конфигурирование и настройка информационной системы.

Бизнес-правила определяют условия корректности совместного применения различных компонентов ИС и используются для поддержания целостности создаваемой системы.

Модель бизнес-функций представляет собой иерархическую декомпозицию функциональной деятельности предприятия (подробное описание см. в разделе "Анализ и моделирование функциональной области внедрения ИС").

Модель бизнес-процессов отражает выполнение работ для функций самого нижнего уровня модели бизнес-функций (подробное описание см. в разделе "Спецификация функциональных требований к ИС"). Для отображения процессов используется модель управления событиями (EPC - Event-driven Process Chain). Именно модель бизнес-процессов позволяет выполнить настройку программных модулей - приложений информационной системы в соответствии с характерными особенностями конкретного предприятия.

Модели бизнес-объектов используются для интеграции приложений, поддерживающих исполнение различных бизнес-процессов (подробное описание см. в разделе "Этапы проектирования ИС с применением UML").

Модель организационной структуры предприятия представляет собой традиционную иерархическую структуру подчинения подразделений и персонала (подробное описание см. в разделе "Анализ и моделирование функциональной области внедрения ИС").

Внедрение типовой информационной системы начинается с анализа требований к конкретной ИС, которые выявляются на основе результатов предпроектного обследования объекта автоматизации (см. раздел "Анализ и моделирование функциональной области внедрения ИС"). Для оценки соответствия этим требованиям программных продуктов может использоваться описанная выше методика оценки ППП. После выбора программного продукта на базе имеющихся в нем референтных моделей строится предварительная модель ИС, в которой отражаются все особенности реализации ИС для конкретного предприятия. Предварительная модель является основой для выбора типовой модели системы и определения перечня компонентов, которые будут реализованы с использованием других программных средств или потребуют разработки с помощью имеющихся в составе типовой ИС инструментальных средств (например, ABAP в SAP, Tools в BAAN).

Реализация типового проекта предусматривает выполнение следующих операций:

- установку глобальных параметров системы;

- задание структуры объекта автоматизации;
 - определение структуры основных данных;
 - задание перечня реализуемых функций и процессов;
 - описание интерфейсов;
 - описание отчетов;
 - настройку авторизации доступа;
- настройку системы архивирования.

Тема 4. Разработка прототипа информационной системы. Тестирование взаимодействия программного интерфейса с базой данных

Тема самостоятельного изучения № 2. Разработка прототипа информационной системы. Тестирование взаимодействия программного интерфейса с базой данных

Вид деятельности студентов: самостоятельное изучение учебно-методической литературы

Итоговый продукт самостоятельной работы: конспект

Средства и технологии оценки: собеседование

План собеседования:

Понятие прототипа. Основные этапы разработки прототипа. Инструментарий для разработки прототипа. Разработка базы данных. Использование механизмов для доступа к базе данных. Основные этапы тестирования.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-2	1-2

Учебный проект: "Разработка ИС предприятия оптовой торговли лекарственными препаратами": версия для печати и PDA

Методика выполнения учебного задания основана на опыте ряда успешных проектов внедрения КИС Navision и Ахарта.

Практическое задание позволяет изучить состав, содержание и процедуры формирования основных документов, которые создаются в процессе типового проектирования ИС (см. раздел "Организация разработки ИС"), приобрести навыки разработки диаграмм бизнес-процессов на основе их вербального описания, которое получается в результате обследования деятельности предприятий.

Порядок выполнения практического задания

В процессе выполнения практического задания проводится анализ и оформление результатов обследования деятельности гипотетического предприятия "МЕД", и на его основе разрабатываются документы, необходимые для настройки типовой ИС. По итогам проведения обследования обычно формируются следующие документы:

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Щербук Татьяна Александровна
Действителен с 19.06.2022 по 19.06.2023

- Предварительная информация.
- Видение выполнения проекта и границы проекта.
- Отчет об обследовании.

Предварительная информация

Предполагается, что в начале обследования проведен предварительный сбор информации о компании, по итогам которого получены следующие данные:

- Краткая информация о компании (профиль клиента).
- Цели проекта.
- Подразделения и пользователи системы.

На основе предварительной информации сформировано и согласовано с заказчиком общее представление о проекте:

Видение выполнения проекта и границы проекта - документ, который кратко описывает, в каких подразделениях и в какой функциональности будет внедряться ИС. Затем выполняется детальное обследование предприятия, результаты которого оформляются в виде отдельного документа - отчета об обследовании.

Отчет об обследовании содержит следующие разделы:

- Анализ существующего уровня автоматизации.
- Составляется список программного обеспечения, используемого в компании, и приводятся данные об использовании этих пакетов в каждом из подразделений организации.
- Общие требования к ИС
- Формулируются общие требования к функциональности разрабатываемой системы.
- Формы документов
- Устанавливается перечень и структура документов, которые должны формироваться системой.
- Описание системы учета
- Описание системы учета включает в себя следующие документы:
- Учетная политика компании
- План счетов и используемых аналитик
- Список типовых хозяйственных операций и их отражение в проводках
- Описание справочников
- По каждому справочнику, проектируемому в системе, дается описание необходимой иерархической структуры.
- Организационная диаграмма
- Организационная диаграмма используется для отражения организационной структуры подразделений предприятия и их зон ответственности.
- Описание состава автоматизируемых бизнес-процессов
- Все бизнес-процессы компании должны быть перечислены в общем списке и каждый должен иметь свой уникальный номер.
- Диаграммы прецедентов
- Для выделения автоматизируемых бизнес-процессов и их основных исполнителей используются диаграммы прецедентов.
- Физическая диаграмма
- Физическая диаграмма служит для того, чтобы описать взаимодействие организации на верхнем уровне с внешними контрагентами.
- Описания бизнес-процессов (книга бизнес-процессов).

Далее в **отчете об обследовании** включается книга бизнес-процессов, содержащая подробное описание автоматизируемых бизнес-процессов. Модели бизнес-процессов позволяют выделить отдельные операции, выполнение которых должно поддерживаться разрабатываемой ИС.

На последнем этапе осуществляется отображение модели предметной области на функциональность типовой системы - выбираются модули системы для поддержки выделенных операций, определяются особенности их настройки, выявляется необходимость разработки дополнительных программных элементов.

Краткая информация о компании "МЕД"

Компания - дистрибьютор "МЕД" закупает медицинские препараты отечественных и зарубежных производителей и реализует их через собственную дистрибьюторскую сеть и сеть аптек. Компания осуществляет доставку товаров как собственным транспортом, так и с помощью услуг сторонних организаций.

Основные бизнес-процессы компании - закупки, складирование запасов, продажи, взаиморасчеты с поставщиками и клиентами.

Уровень конкуренции для компании в последнее время возрос, так как на рынок вышли два новых конкурента, к которым перешла часть клиентов и ряд наиболее квалифицированных сотрудников ЗАО "МЕД". ЗАО "МЕД" имеет два филиала - в Курске и Санкт-Петербурге. Каждый филиал функционирует как самостоятельное юридическое лицо, являясь полностью принадлежащей ЗАО "МЕД" дочерней компанией.

По предварительным планам, Компания намерена открыть также дочернее предприятие для организации производства в непосредственной близости к своим заказчикам.

Адреса и телефоны

Москва, К-123 Центральная улица, д. 20, стр. 7, офис 709

Телефон: (095) 345-6789, факс: (095) 345-9876

Контактные лица

Борис Нефедьев - Генеральный директор

Дмитрий Кононов - Исполнительный директор

Артур Иванченко - Директор по маркетингу

Сотрудники

На момент проведения Диагностики штат компании составляет 110 сотрудников.

Основными целями проекта автоматизации компании "МЕД" являются:

- Разработка и внедрение комплексной автоматизированной системы поддержки логистических процессов компании.
- Повышение эффективности работы всех подразделений компании и обеспечение ведения учета в единой информационной системе.

Видение выполнения проекта и границы проекта

В рамках проекта развертывание новой системы предполагается осуществить только в следующих подразделениях ЗАО "МЕД":

- Отдел закупок;
- Отдел приемки;
- Отдел продаж;
- Отдел маркетинга;
- Группа планирования и маркетинга;
- Группа логистики;
- Учетно-операционный отдел;
- Учетный отдел;
- Отдел сертификации (в части учета сертификатов на медикаменты);
- Бухгалтерия (только в части учета закупок, продаж, поступлений и платежей).

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владельцем документа является ЗАО "МЕД". Не рассматривается в границах проекта автоматизация учета основных средств, расчета и начисления заработной платы, управления кадрами. Выходит за рамки проекта

Действие документа ограничено автоматизацией процессов взаимоотношений с клиентами.