

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ**
Кафедра автоматизированных систем управления (АСУ)

М.В. Григорьева

УЧЕБНАЯ ПРАКТИКА

«Практика по получению первичных профессиональных
умений и навыков, в том числе первичных умений и
навыков научно-исследовательской деятельности»

**Учебное пособие для студентов направления
09.03.03 «Прикладная информатика» заочной формы
обучения**

Григорьева М.В.

Учебная практика «Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности»: учебное пособие для студентов направления 09.03.03 «Прикладная информатика» заочной формы обучения. — Томск: Кафедра автоматизированных систем управления, ТУСУР, 2017. — 93 с.

© Григорьева М.В., 2017

© Кафедра автоматизированных систем
управления, ТУСУР, 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ.....	5
1.1 Цели и задачи практики	5
1.2 Оформление отчета	6
1.3 Содержание основной части отчета по практике	9
3 МЕТОДИКИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ	20
3.1 Составные части программных средств	20
3.2 Моделирование предметной области ПС	23
3.3 Визуальное моделирование	27
3.4 Структурные методы анализа и проектирования ПО	33
3.4.1. Метод функционального моделирования SADT (IDEFO)	36
3.4.2 Моделирование потоков данных	45
3.4.3 Сравнительный анализ SADT-моделей и диаграмм потоков данных.....	54
3.4.4 Функциональные модели, используемые на стадии проектирования.....	56
3.5. Моделирование данных.....	58
4 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ.....	60
5 ТЕМАТИКА ПРАКТИКИ	64
6 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО СТРУКТУРЕ И ОФОРМЛЕНИЮ ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ	67
6.1 Структура отчета	67
6.2 Общие положения по оформлению отчета.....	69
6.3 Деление текста отчета	70
6.4 Таблицы, иллюстрации, формулы.....	71
СПИСОК ЛИТЕРАТУРЫ.....	74
ПРИЛОЖЕНИЕ А. ПРИМЕР ОФОРМЛЕНИЯ ТИТУЛЬНОГО ЛИСТА ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ.....	76
ПРИЛОЖЕНИЕ Б. ПРИМЕР ФОРМЫ ЗАДАНИЯ	77
ПРИЛОЖЕНИЕ В. ПРИМЕР ВВЕДЕНИЯ ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ.....	78
ПРИЛОЖЕНИЕ Г. ПРИМЕР ОФОРМЛЕНИЯ ФРАГМЕНТА РАБОТЫ.....	80
ПРИЛОЖЕНИЕ Д. ПРИМЕР ЗАКЛЮЧЕНИЯ ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ	81
ПРИЛОЖЕНИЕ Е. ПРИМЕР ОПИСАНИЯ БИЗНЕС-ПРОЦЕССА ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗАЦИИ И ФОРМАЛИЗОВАННОЙ ПОСТАНОВКИ ЗАДАЧИ	82
ПРИЛОЖЕНИЕ Ж. ПРИМЕРЫ БИБЛИОГРАФИЧЕСКОГО ОПИСАНИЯ КНИГ И ДРУГИХ ИЗДАНИЙ.....	88
ПРИЛОЖЕНИЕ И ПРИМЕРЫ ОФОРМЛЕНИЯ ТАБЛИЦ И РИСУНКОВ.....	92

ВВЕДЕНИЕ

Настоящее учебное пособие предназначено для студентов направления 09.03.03 «Прикладная информатика», обучающихся по заочной форме, в том числе с применением дистанционных образовательных технологий.

Учебная практика «Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности» (далее учебная практика) является важным этапом обучения студентов в вузе и имеет своей основной **целью** получение представления студентами о требованиях к профессиональной компетенции специалистов со стороны предприятий, организаций и учреждений.

Задачей учебной практики является изучение современных средств разработки информационных систем (ИС), их сравнительный анализ, а также приобретение практических навыков разработки ИС путем создания прототипа (макета, черновой, пробной версии) информационной системы.

В учебном пособии предложена ориентировочная структура отчета по учебной практике, которая снабжена расшифровкой отдельных пунктов работы. Включены главы по проектированию и моделированию программной части информационных систем. Сформулированы основные требования, предъявляемые к оформлению отчета по практике. В приложениях даны образцы оформления титульного листа, задания на практику, введения, заключения, списка используемых источников, заголовков разделов, рисунков, таблиц. Приведен пример формализованной постановки задачи на создание информационной системы из работы студента.

1 СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ

1.1 Цели и задачи практики

Учебная практика «Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности» выполняется студентами заочной формы обучения по материалам, собранным студентом в течение предыдущих семестров во время выполнения учебной исследовательской и научной работ студента.

Цель учебной практики:

– формирование профессиональной компетенции студентов через применение полученных теоретических знаний в решении конкретных информационных и научно-исследовательских задач;

– ознакомление и изучение опыта создания и применения информационных технологий и систем информационного обеспечения для решения задач организационной, управленческой или научной деятельности для целей конкретных производств, организаций или фирм;

– изучение современных средств разработки информационных систем (ИС), их сравнительный анализ, а также приобретение практических навыков разработки ИС путем создания прототипа информационной системы.

Задачи учебной практики:

1) ознакомиться с литературными источниками (10—15 наименований включить в отчет), в которой освещается отечественный и зарубежный опыт проектирования и разработки информационных систем в конкретной области;

2) выполнить формализованную постановку задачи на автоматизацию бизнес-процесса предприятия, определить входную, выходную информацию, автоматизируемые функции;

3) провести обзор современных программных средств реализации информационных систем и обоснование выбранной программной среды

(например, СУБД Oracle, SQL Server, MS Access; Visual C++, VBA, Borland Delphi, Borland C++, 1С Предприятие и т. д.);

4) построить функциональные диаграммы IDF0 «КАК-ЕСТЬ» и/или «КАК-БУДЕТ»;

5) создать прототип информационной системы и его тестирование на контрольном примере.

На заключительном этапе учебной практики студент должен обобщить материал, собранный в период прохождения практики, и оформить **отчет по практике** (25—30 страниц).

1.2 Содержание отчета

Отчет по практике составляется индивидуально каждым студентом и должен отражать его деятельность в период практики.

Отчет состоит из следующих обязательных разделов и частей:

- титульный лист (приложение А);
- задание (приложение Б);
- оглавление;
- введение (приложение В);
- основная часть (приложение Г);
- заключение (приложение Д);
- список используемых источников (приложение Ж).

К необязательным разделам отчета по учебной практике относятся:

- список сокращений, обозначений и специальных терминов, который может располагаться со следующей страницы после заключения;
- приложения.

Примеры оформления титульного листа и задания приведены в приложениях А и В соответственно. Тема работы на титульном листе и содержание задания должно совпадать с содержанием раздела 3 дневника по учебной практике.

Оглавление – перечень основных частей работы с указанием листов (страниц), на которых их помещают. Оглавление должно отражать все материалы, представляемые в отчете. Слово «Оглавление» записывают в виде заголовка, симметрично тексту, с прописной буквы, без номера раздела. **Примечание** – Вместо слова «Оглавление» допускается использовать наименование «Содержание». В оглавлении перечисляют заголовки разделов, подразделов (глав, параграфов), список использованных источников, каждое приложение работы и указывают номера листов (страниц), на которых они начинаются.

Раздел «Задание» в оглавлении не указываются. Заголовки в оглавлении должны точно повторять заголовки в тексте работы. Последнее слово заголовка соединяют отточием с соответствующим ему номером листа (страницы) в правом столбце оглавления. При наличии проектных документов, помещаемых в работе, их перечисляют в оглавлении после остальных приложений с указанием обозначений документов (если они присвоены) и их наименований. Пример оформления оглавления приведен в приложении Д.

Введение должно обобщить собранные материалы и раскрыть основные вопросы и направления, которыми занимался студент во время прохождения практики. Введение должно содержать *цель и задачи практики, объект и предмет исследования, актуальность работы, используемые методы*. Введение должно составлять не менее одной полной страницы и не более 3-х страниц текста.

Заголовок «Введение» записывают симметрично тексту с прописной буквы. В приложении В приведен пример введения.

Основная часть включает в себя аналитическую записку по разделам примерного тематического плана учебной практики. Включаются в отчет и элементы научных исследований. Тематика этих исследований определяется заранее, согласовывается с Руководителем и увязывается с общим направлением работ. Это может быть, например, обзор современных средств

проектирования и разработки информационных систем, российский и международный опыт в области информатизации в соответствующих областях.

Основная часть отчета по практике может быть разбита на несколько разделов и подразделов. В приложении С приведен пример оформления заголовков в этом случае.

В **заключении** приводятся общие выводы, результаты проделанной работы, даются практические рекомендации и обозначаются основные проблемы.

Список использованных источников содержит библиографическое описание всех литературных источников, использованных в отчете по учебной практике. Сведения о каждом из источников располагают в порядке их упоминания в тексте или в алфавитном порядке. Образцы библиографического описания наиболее важных типов литературных источников (с учетом требований нормативных документов) приведены в приложении Ж. На каждый источник из списка в тексте отчета необходимо поставить ссылку, указывая после взятого из источника фрагмента номер источника из списка в квадратных скобках. Также, как это сделано в данном пособии ниже на этой странице.

В приложении могут быть примеры первичных экономических документов (входные и выходные формы и бланки или реальные заполненные документы); информация об объекте исследования (сложная структура предприятия, положения о видах деятельности, лицензии, фрагменты громоздких таблиц номенклатур продукции, различные справочники предприятий и т. д.), иллюстрации и таблицы, если они являются громоздкими и не вписываются в формат пояснительной записки.

Отчет необходимо оформить в соответствии с **образовательным стандартом вуза ОС ТУСУР 01-2013** «Работы студенческие по направлениям подготовки и специальностям технического профиля. Общие

требования и правила оформления», ознакомиться с которым можно на официальном сайте университета (<http://www.tusur.ru/>) в разделе *Студентам/ Правила оформления курсовых и выпускных работ*.

В разделе 3 данного пособия приведена содержательная структура отчета, в разделе 5 - некоторые требования по его оформлению.

В конце этого пособия приведен список литературы, которая призвана помочь студенту в выполнении задания [1-7] и оформлении отчета [8] по учебной практике.

1.3 Содержание основной части отчета по практике

Рассмотрим примерный порядок подготовки разделов отчета по учебной практике и их содержание. Основную часть отчета по практике можно разделить на четыре раздела.

Первый раздел содержит общую характеристику предметной области автоматизации, здесь необходимо дать краткий обзор деятельности предприятия и его организационной структуры управления, состояние и стратегия развития информационных технологий. Содержание этого раздела зависит от темы, выбранной студентом.

С одной стороны, нет необходимости приводить объемную информацию, не имеющую прямого отношения к решаемой задаче. В то же время, описание предметной области должно быть достаточным для понимания автоматизируемого бизнес-процесса и его места в деятельности предприятия.

Описание существующей организации бизнес- и информационных процессов может быть выполнено с использованием IDEF0, ARIS, DFD, UML методологий [1,2,6,7]. После этого необходимо провести анализ недостатков (проблем, узких мест) в организации бизнес- и информационных процессов и сформулировать предложение по автоматизации (информатизации) бизнес-процессов (решения задач, комплекса задач, подсистем) с учетом анализа

успешных ИТ-проектов в рассматриваемой области, рынка программного обеспечения и ИТ-технологий, выбора технологии проектирования ИС.

Далее приведен возможные состав информации этого раздела в зависимости от тематики учебной практики студента.

Общая характеристика предметной области:

– цели и задачи производства продукции, оказания услуг на предприятии;

– номенклатура выпускаемой продукции, особенности ее производства (используемые технологии, проектные решения);

– масштаб деятельности предприятия и его место на рынке;

– миссия и имидж предприятия в (регионе, стране, за рубежом);

– характер производственной деятельности, система снабжения и сбыта;

– географическое положение, климатические и природные условия местоположения предприятия;

– степень механизации и автоматизации производства и процессов управления, проблемы и задачи дальнейшей автоматизации;

– уровень специализации, кооперирования и концентрации производства; организационную структуру производства (организационно-правовые формы структурных подразделений и характер организационных отношений между ними);

– производственная структура предприятия;

– стратегия и тактика управления предприятием;

– уровень организационной культуры.

Организационная структура управления:

– организационная структура управления деятельностью предприятия с учетом его организационно-правовой формы;

- характер организационных отношений между структурными подразделениями;
- компоненты организационной структуры; линейные подразделения (управление основным производством), функциональные структурные подразделения (совещательные функции и функциональные полномочия), обеспечивающие структурные подразделения;
- методы, применяемые на предприятии для совершенствования и поощрения организационных структур управления на каждом уровне;
- структура и функции аппарата управления предприятия;
- регламентация деятельности структурных подразделений; их внутреннюю структуру, связи с другими структурными подразделениями;
- эффективность и экономичность структуры управления, механизмы ее совершенствования.

В этом разделе студент может привести схемы организационных структур управления предприятия, его структурных подразделений (отдела, цеха и т. д.), на базе которых выполняется учебная практика.

Состояние и стратегия развития информационных технологий:

- идентифицировать используемые на предприятии ИС и описать бизнес-процессы, которые они поддерживают;
- описать состав сетевой архитектуры, компьютерной техники и средств телекоммуникаций. Степень подробности зависит от сложности рассматриваемой сети, применения специализированного оборудования, средств маршрутизации, защиты сети и т. д. — обсуждается с консультантом по данному разделу диплома;
- указать используемое системное и прикладное программное обеспечение;
- описать работу ИТ-подразделений и служб;
- отметить имеющиеся решения по обеспечению информационной безопасности.

Необходимо кратко описать компьютерные средства, используемые в организации (подразделении), можно привести обобщенные параметры компьютеров. Примером такого описания может быть таблица 2.1.

Таблица 2.1 – Пример описания парка компьютерной техники

Компьютерная техника	Количество
Компьютеры всего	54
из них: компьютеры в ЛВС	42
серверы	7
несвязанных ЛВС	5
Принтеры	15

Программную архитектуру целесообразно формировать исходя из существующих программных систем (программных продуктов), которые функционируют в рамках или параллельно с прочими обеспечивающими системами. В качестве основы работы программных продуктов целесообразно опираться на операционную систему, в которой они функционируют.

Необходимо указать используемые в организации прикладные программные средства: офисные программы, бухгалтерские, складские и другие информационные системы. Необходимо показать, для решения каких задач они используются, например, в виде таблицы 2.2.

Таблица 2.2 – Программное обеспечение организации

Программа	Решаемая задача
1. Microsoft Office 2010	Ведение текущей документации, подготовка отчетов для вышестоящей организации
2. 1С: Предприятие 8.3	Введение бухгалтерского и налогового учета
...	...

В этом разделе следует отметить имеющиеся решения по политике безопасности в компании, а также программные и аппаратные средства информационной безопасности и защиты информации, если эти методы и средства используются, то каким образом.

В заключение (или в каждом подразделе) надо провести анализ и сделать выводы о достаточности и эффективности использования имеющихся

программных средств и компьютерного оборудования, степени обеспечения информационной безопасности и защиты информации, а также необходимости соответствующих разработок.

Управление производством. Если тема практики связана с автоматизацией функции планирования, то данный раздел предполагает изучение:

- производственной функции планирования на предприятии;
- структуры организационной системы, включающей подсистемы обеспечения, планирования и контроля;
- оперативного планирования выпуска продукции, нормирования труда и управления запасами;
- календарного планирования и диспетчеризации производства;
- мотивации и стимулирования труда в целях активизации человеческого фактора;
- управления производством на уровне высшего руководства, руководителей среднего и низшего звена.

Для выполнения вышеперечисленных задач студенту необходимо:

- ознакомиться с организацией учета, анализа и планирования производства на предприятии, методикой разработки планов, прогнозов, текущих производственных программ в целом по предприятию и по базовому производственному подразделению, в котором решаются задачи автоматизации;
- изучить процесс планирования объемов выпуска продукции и разработки соответствующих программ маркетинга, обосновать необходимость их использования на данном предприятии;
- проанализировать методы и приемы, используемые при обосновании производственной программы, с учетом потребностей рынка, производственных мощностей, материальных, трудовых и финансовых ресурсов;

- изучить систему показателей, характеризующих производственную деятельность, ее эффективность, а также научиться использовать их в процессе принятия управленческих решений;
- проанализировать систему организации производственных связей между звеньями, подразделениями предприятия, ее эффективность;
- ознакомиться с системой товародвижения, сбыта и продвижения продукции на предприятии (планированием товародвижения, выбором и оценкой каналов сбыта, формами организации торговли);
- дать комплексную оценку организации управления производственно-сбытовой деятельностью предприятия;
- изучить систему управления маркетингом (на предприятиях с маркетинговыми службами) и дать ей оценку, если выпускная квалификационная работа касается задач по маркетингу.

Управление персоналом. Если тема практик связана с автоматизацией работ структурных подразделений, отвечающих за работу с кадрами (например, отдел кадров), необходимо изучить кадровый потенциал предприятия и его формирование, в том числе:

- профессиональный состав, квалификационный и образовательный уровень кадров (по категориям персонала);
- планирование и прогнозирование потребности в рабочей силе (по категориям персонала);
- процесс подбора кадров руководителей и специалистов;
- систему материального и морального стимулирования;
- функции управленческого персонала;
- требования, предъявляемые к работникам аппарата управления, их соответствие квалификационным требованиям общей и специальной подготовки;
- систему работы с кадрами (ротация, повышение квалификации, подготовка и переподготовка кадров, работа с резервом на выдвижение и др.);

- методы повышения эффективности управленческого труда.

Информационная система управления. Если тема практик касается автоматизации коммуникационных процессов на предприятии, то по данному разделу студенты должны изучить:

- коммуникационные процессы, используемые на данном предприятии;
- особенности коммуникаций между уровнями управления и подразделениями;
- особенности коммуникаций между предприятием (организацией) и внешней средой (органы государственного управления и регулирования, потребители, поставщики, конкуренты и др.);
- характеристику общей схемы информационных потоков в организации;
- информационный процесс (передача, преобразование, хранение, оценка и использование информации);
- массивы информации, их средства передачи и алгоритмы преобразования;
- документооборот предприятия и его характеристику;
- организацию делопроизводства, рационализацию документооборота предприятия;
- пути устранения недостатков коммуникационного процесса;
- методы и пути совершенствования коммуникаций в организации.

Экономические службы и финансовый менеджмент. Если тема практик касается автоматизации задач финансовых структур на предприятии, то по этому разделу студенту необходимо:

- изучить структуру и содержание деятельности экономических служб;
- ознакомиться с системой цен на выпускаемую продукцию и методикой их формирования;

- приобрести навыки составления финансового плана, баланса доходов и расходов предприятия;
- на основе бухгалтерского баланса уметь рассчитывать и оценивать показатели финансового состояния предприятия;
- изучить механизм формирования, распределения и использования прибыли, а также резервы увеличения прибыли и рентабельности;
- осуществить комплексную оценку финансового положения предприятия, определить основные финансовые проблемы;
- внести предложения по выработке соответствующих управленческих решений в области ценообразования и финансов, по их реализации и контролю за выполнением принятых решений.

Управление маркетингом. Если тема практик касается автоматизации задач службы маркетинга, то студент должен отразить в отчете:

- задачи службы маркетинга;
- систему, организации службы маркетинга;
- планирование продукции и товарную политику;
- задачи и механизм действия подсистемы маркетинговых исследований рынка и возможностей фирмы;
- структуру и функции маркетинговой информационной системы;
- управление качеством товара;
- стадии жизненного цикла товара и функции маркетинговых служб;
- инновационную политику;
- ассортиментную политику предприятия и направления его ассортиментной стратегии;
- роль посредников в товародвижении;
- условия выбора и интенсивность каналов товародвижения и сбыта;
- требования к торговым посредникам;
- планирование продвижения товара на рынке;
- коммуникационную политику фирмы;

- основные методы реализации изделий (услуг);
- стратегию и тактику фирмы в области рекламы и ее эффективность;
- инструменты стимулирования сбыта;
- ценовую политику и конкуренцию;
- факторы, определяющие решения по ценам;
- ценовую рекламу;
- ценовые гарантии и особые соглашения;
- определение цен на новые изделия и их поведение на рынке;
- предложения по повышению эффективности маркетинговой деятельности предприятия.

Заканчивается первый раздел формализованной постановкой задачи автоматизации (информатизации) бизнес-процессов (решения задач, комплекса задач, подсистем).

В приложении Е приведен небольшой пример описания бизнес-процесса и постановки задачи учета заказов на ремонт бытовой и компьютерной техники и их выполнения.

Второй раздел отчета по учебной практике включает обзор современных программных средств реализации информационных систем и обоснование выбранной программной среды (например, СУБД Oracle, SQL Server, MS Access; Visual C++, VBA, Borland Delphi, Borland C++, 1С Предприятие и т. д.).

Необходимо провести обзор 2-3 средств реализации, их достоинств и недостатков, обосновать выбор средств, которые студент будет использовать для решения задач учебной практики.

В случае использования разных средств для реализации базы данных и интерфейса, необходимо проработать материал в двух направлениях.

Обзор средств реализации следует закончить выводом, обобщающей таблицей, которые объясняют выбор студентом конкретной среды реализации.

Часто предприятие, выступающее заказчиком на создание информационной системы, сами определяют среду разработки. В этом случае студенту необходимо подробно раскрыть возможности среды для целей решения поставленной задачи.

Третий раздел отчета описывает проектирование и разработку прототипа информационной системы. В рамках учебной практики достаточно описать построение функциональной диаграммы в нотации IDf0, проектирование базы (или структуры) данных, интерфейса в зависимости от тех средств, которыми студент будет реализовывать систему.

Создание прототипа информационной системы является предварительным этапом разработки информационной системы, процесс создания макета (черновой, пробной версии) программы, обычно с целью проверки пригодности предлагаемых для применения концепций, архитектурных и/или технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки.

Процесс создания прототипа обычно состоит из следующих шагов:

- 1) определение начальных требований;
- 2) разработка первого варианта прототипа, который содержит только пользовательский интерфейс системы, функции системы могут быть реализованы частично;
- 3) изучение прототипа заказчиком и конечными пользователями, получение обратной связи о необходимых изменениях и дополнениях;
- 4) переработка и улучшение прототипа: с учетом полученных замечаний и предложений изменяются как спецификации, так и прототип, после этого шаги 3 и 4 могут повторяться.

Более подробно с технологиями проектирования можно познакомиться в разделе 3 данного пособия

Четвертый раздел содержит пример тестирования созданного прототипа информационной системы на контрольном примере. Основные функции информационной системы представляются в отчете рисунками с копиями экранных форм, приводятся формируемые системой (возможно, пока не полностью) документы и отчеты.

Отдельно следует отметить, что студенту необходимо ознакомиться с литературными источниками (10—15 наименований включить в отчет), в которых освещается отечественный и зарубежный опыт проектирования и разработки информационных систем. На все источники должны быть оформлены ссылки по тексту отчета.

Используйте приведенную в конце пособия литературу для проектирования информационных систем, баз данных, создания программных систем путем объектно-ориентированного подхода, визуального программирования [1-7].

3 МЕТОДИКИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

3.1 Составные части программных средств

Предметная область программного обеспечения (ПО) определяет некоторую структуру данных, т.е. организацию входных, промежуточных и выходных данных. Эти структурированные данные называются информационной базой программного средства (ПС), соответствующей своей предметной области.

Для реализации выбранных пользователем конкретных действий ПС должно воспринимать от пользователя управляющую информацию. Эта управляющая информация представляется на формальном языке – входном языке пакета. Описание конкретного задания пользователя на входном языке называют *программой на входном языке (ПВЯ)*.

ПО состоит из нескольких программных единиц. Такие программные единицы обычно называют программными модулями. Решение каждой задачи сводится к выполнению соответствующего алгоритма. Программные модули ПС, реализующие алгоритмы решения задач, будем называть *обрабатывающими модулями*. Обработывающие модули выполняют преобразование данных, составляющих *информационную базу ПС*.

Для того чтобы преобразовать задание пользователя в последовательность вызовов обрабатывающих модулей, в ПС должны входить управляющие модули.

Чтобы обеспечить взаимодействие пакета с пользователем и управляющих модулей ПС с информационной базой и обрабатывающими модулями, в состав ПС включаются обслуживающие модули.

Таким образом, ПС можно рассматривать как объединение входного языка, информационной базы, управляющих, обслуживающих и обрабатывающих программных модулей.

Взаимодействие составных частей пакета схематически показано на рисунке 3.1. Средствами операционной системы запускается головной

управляющий модуль пакета (ведущий модуль). Затем организуются прием задания пользователя, представляемого в форме программ на входном языке (ПВЯ), и выполнение этого задания путем вызова в нужной последовательности обрабатывающих и обслуживающих модулей.

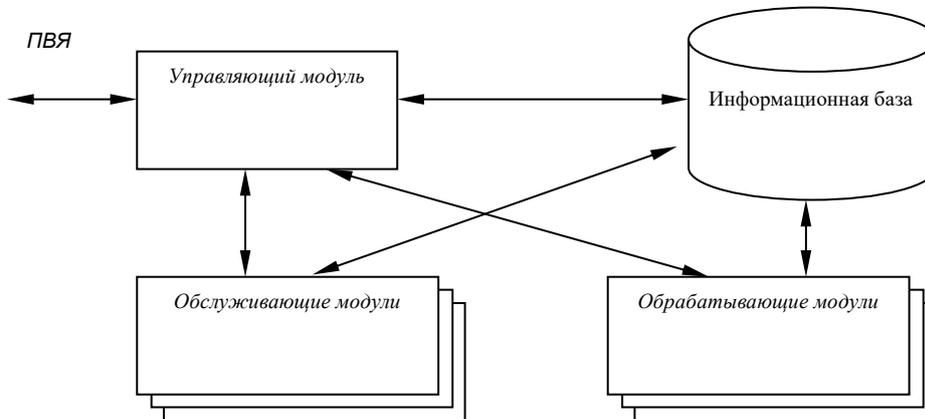


Рис. 3.1 – Составные части ПС

Анализируя структуры современных ПС, можно отметить, что они содержат не зависящие от содержания предметной области составные следующие части:

- управляющие программы – мониторы для вызова модулей и библиотечных подпрограмм;
- языковые процессоры для перевода формулировки прикладной задачи на язык программирования;
- архивные подсистемы;
- специализированные базы данных;
- средства диалогового взаимодействия с пользователем и т.д.

Отсюда следует возможность разработки комплексов базовых (типовых) программных средств, поддерживающих общую структуру пакета, его связь с системным ПО и пользователем, и настраивающихся на конкретные средства внешнего управления и конкретные модели предметных областей. Эти комплексы и получили название: **системное наполнение пакета**, или оболочка пакета. В них входят управляющие и обслуживающие модули. Тогда комплекс специальных программ, определяющих конкретную область

применения ПС можно назвать *функциональным наполнением пакета*. Этот комплекс включает в себя обрабатывающие модули.

Для настройки ПС на конкретную предметную область необходимо погрузить в оболочку пакета описание информационной базы пакета, описания функциональных связей и связей по определению, а также подключить обрабатывающие модули.

Таким образом, появляется возможность разработки программных средств генерации ПС для различных предметных областей, использующих одну и ту же оболочку.

Ярким примером такого подхода является платформа 1С v8.x - программный framework, имеющий встроенный язык 1С, сложные прикладные механизмы, объекты и инструменты для построения учетной системы, объединяющий средство разработки и исполняющую платформу прикладных решений.

Пакетный режим работы. Вся управляющая информация для конкретного выполнения пакета передается в виде законченной программы на входном языке при запуске ПС и дальнейшая работа проходит без участия пользователя.

Пакетный режим удобен, когда:

- а) требуется решать много однотипных задач с использованием одной и той же программы на входном языке;
- б) время, затрачиваемое на решение каждой задачи, достаточно велико;
- в) программа на входном языке сложна и имеет значительный объем.

Диалоговый режим работы. Большинство ПС, применяемых на персональных ЭВМ, ориентировано на диалоговое взаимодействие с пользователем в ходе решения задач.

Простейший диалоговый режим состоит в том, что пользователь инициирует выполнение пакета, вводит задание в форме программы на входном языке и на этом заканчивает управление выполнением пакета.

Практически этот режим отличается от пакетного только возможностью исправления ошибок в ПВЯ, повторного запуска пакета при неудачах.

Более сложный вариант диалогового режима, называемый также *режимом сопровождения*, предусматривает возможность динамического управления выполнением пакета. Управляющая информация вводится по частям и формируется пользователем в процессе работы с пакетом на основе анализа промежуточных результатов. Такая работа в большинстве случаев более естественна для пользователя, в частности, при использовании пакетов редактирования текстов, при работе с электронными таблицами, при решении многих расчетных задач.

Выбор того или иного способа применения ПС зависит от многих факторов, из которых наиболее существенными являются возможности операционной системы и выбранного языка программирования, объемы обрабатываемых данных, продолжительность решения задачи, частота использования ПС, особенности квалификации пользователей ПС и требования к оперативности решения задач (допустимому времени ожидания результатов расчетов).

3.2 Моделирование предметной области ПС

Как было отмечено ранее, одной из основных проблем, которые приходится решать при создании больших и сложных систем любой природы, в том числе и ПО, является проблема сложности. Ни один разработчик не в состоянии выйти за пределы человеческих возможностей и понять всю систему в целом. Единственный эффективный подход к решению этой проблемы, который выработало человечество за всю свою историю, заключается в построении сложной системы из небольшого количества крупных частей, каждая из которых, в свою очередь, строится из частей меньшего размера, и т.д., до тех пор, пока самые небольшие части можно будет строить из имеющегося материала. Этот подход известен под самыми разными названиями, среди них такие, как *«разделяй и властвуй»* (divide et impera),

иерархическая декомпозиция и др. По отношению к проектированию сложной программной системы это означает, что ее необходимо разделить (декомпозировать) на небольшие подсистемы, каждую из которых можно разрабатывать независимо от других. Это позволяет при разработке подсистемы любого уровня иметь дело только с ней, а не со всеми остальными частями системы. Правильная декомпозиция является главным способом преодоления сложности разработки больших систем ПО. Понятие «правильная» по отношению к декомпозиции означает следующее:

- количество связей между отдельными подсистемами должно быть минимальным (принцип «слабой связанности» — Low Coupling);
- связность отдельных частей внутри каждой подсистемы должна быть максимальной (принцип «сильного сцепления» - High Cohesion).

Более подробно эти принципы будут рассмотрены в рамках объектно-ориентированного анализа.

Структура системы должна быть такой, чтобы все взаимодействия между ее подсистемами укладывались в ограниченные, стандартные рамки, т.е.:

- каждая подсистема должна **инкапсулировать** свое содержимое (скрывать его от других подсистем);
- каждая подсистема должна иметь четко определенный **интерфейс** с другими подсистемами.

Инкапсуляция (принцип «черного ящика») позволяет рассматривать структуру каждой подсистемы независимо от других подсистем.

Интерфейсы позволяют строить систему более высокого уровня, рассматривая каждую подсистему как единое целое и игнорируя ее внутреннее устройство.

Существуют два основных подхода к декомпозиции систем. Первый подход называют **функционально-модульным**, он является частью более общего **структурного подхода**. В его основу положен принцип функциональной **декомпозиции**, при которой структура системы описывается

в терминах *иерархии* ее функций и передачи информации между отдельными функциональными элементами. Второй, ***объектно-ориентированный подход***, использует объектную декомпозицию. При этом структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами.

В 1970-1980 годах при разработке ПО достаточно широко применялись структурные методы, предоставляющие в распоряжение разработчиков строгие формализованные методы описания проектных решений — спецификаций ПО (в настоящее время такое же распространение получают объектно-ориентированные методы). Эти методы основаны на использовании наглядных графических моделей: для описания архитектуры ПО с различных точек зрения (как статической структуры, так и динамики поведения системы) используются схемы и диаграммы. Наглядность и строгость средств структурного и объектно-ориентированного анализа позволяет разработчикам и будущим пользователям системы с самого начала неформально участвовать в ее создании, обсуждать и закреплять понимание основных технических решений. Однако широкое применение этих методов и следование их рекомендациям при разработке конкретных систем ПО сдерживалось отсутствием адекватных инструментальных средств, поскольку при неавтоматизированной (ручной) разработке все их преимущества практически сведены к нулю. Действительно, вручную очень трудно разработать и графически представить строгие формальные спецификации системы, проверить их на полноту и непротиворечивость и тем более изменить. Если все же удастся создать строгую систему проектных документов, то ее переработка при появлении серьезных изменений практически неосуществима. Ручная разработка обычно порождала следующие проблемы:

- неадекватная спецификация требований;
- неспособность обнаруживать ошибки в проектных решениях;
- низкое качество документации, снижающее эксплуатационные характеристики;

– затяжной цикл и неудовлетворительные результаты тестирования.

С другой стороны, разработчики ПО исторически всегда стояли последними в ряду тех, кто использовал компьютерные технологии для повышения качества, надежности и производительности в своей собственной работе (феномен «сапожника без сапог»).

Перечисленные факторы способствовали появлению программно-технологических средств специального класса – CASE-средств, реализующих CASE-технологии создания ПО. Понятие CASE (Computer Aided Software Engineering) используется в настоящее время в весьма широком смысле. Первоначальное значение этого понятия, ограниченное только задачами автоматизации разработки ПО, в настоящее время приобрело новый смысл, охватывающий большинство процессов жизненного цикла ПО.

Появлению CASE-технологии и CASE-средств предшествовали исследования в области программирования: разработка и внедрение языков высокого уровня, методов структурного и модульного программирования, языков проектирования и средств их поддержки, формальных и неформальных языков описаний системных требований и спецификаций и т.д. Кроме того, этому способствовали следующие факторы:

- подготовка аналитиков и программистов, восприимчивых к концепциям модульного и структурного программирования;
- широкое внедрение и постоянный рост производительности компьютеров, позволившие использовать эффективные графические средства и автоматизировать большинство этапов проектирования;
- внедрение сетевой технологии, предоставившей возможность объединения усилий отдельных исполнителей в единый процесс проектирования путем использования разделяемой базы данных, содержащей необходимую информацию о проекте.

CASE-технология представляет собой совокупность методов проектирования ПО, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту

модель на всех стадиях разработки и сопровождения ПО и разрабатывать приложения в соответствии с информационными потребностями пользователей. Большинство существующих CASE-средств основано на методах структурного или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

3.3 Визуальное моделирование

Под *моделью ПО* в общем случае понимается формализованное описание системы ПО на определенном уровне абстракции. Каждая модель определяет конкретный аспект системы, использует набор диаграмм и документов заданного формата, а также отражает точку зрения и является объектом деятельности различных людей с конкретными интересами, ролями или задачами.

Под термином «*моделирование*» понимается процесс создания формализованного описания системы в виде совокупности моделей. Особенно трудным оказывается описание систем средней сложности, таких, как система коммутаций в телефонных сетях, управление авиаперевозками или движением подводной лодки, сборка автомобилей, челночные космические рейсы, функционирование перерабатывающих предприятий (к таким системам относится и ПО). С точки зрения человека, эти системы описать достаточно трудно, потому что они настолько велики, что практически невозможно перечислить все их компоненты со своими взаимосвязями, и в то же время недостаточно велики для применения общих упрощающих предположений (как это принято в физике). Неспособность дать простое описание, а, следовательно, и обеспечить понимание таких систем делает их проектирование и создание трудоемким и дорогостоящим процессом и повышает степень их ненадежности. С ростом технического прогресса адекватное описание систем становится все более актуальной проблемой.

Модель должна давать полное, точное и адекватное описание системы, имеющее конкретное назначение. Это назначение, называемое целью модели, вытекает из формального определения модели: ***М есть модель системы S, если М может быть использована для получения ответов на вопросы относительно S с точностью.***

Таким образом, целью модели является получение ответов на некоторую совокупность вопросов. Эти вопросы неявно присутствуют (подразумеваются) в процессе анализа и, следовательно, они руководят созданием модели и направляют его. Это означает, что сама модель должна будет дать ответы на эти вопросы с заданной степенью точности. Если модель отвечает не на все вопросы или ее ответы недостаточно точны, то говорят, что модель не достигла своей цели.

По мнению авторитетных специалистов области проектирования ПО, моделирование является центральным звеном всей деятельности по созданию систем ПО. Модели строятся для того, чтобы понять и осмыслить структуру и поведение будущей системы, облегчить управление процессом ее создания и уменьшить возможный риск, а также документировать принимаемые проектные решения.

Визуальное моделирование – это способ восприятия проблем с помощью зримых абстракций, воспроизводящих понятия и объекты реального мира. Модели служат полезным инструментом анализа проблем, обмена информацией между всеми заинтересованными сторонами (пользователями, специалистами в предметной области, аналитиками, проектировщиками и т.д.), проектирования ПО, а также подготовки документации. Моделирование способствует более полному усвоению требований, улучшению качества системы и повышению степени ее управляемости.

С помощью модели можно упростить проблему, отбросив несущественные детали и сосредоточив внимание на главном. Способность к абстрагированию - это фундаментальное свойство человеческого интеллекта, помогающее справиться с феноменом сложности. На протяжении тысяч лет

художники, ремесленники и строители предпринимали попытки конструирования тех или иных моделей, предваряющие реальное воплощение творческих замыслов. Не составляет исключения и индустрия разработки ПО. Чтобы создать сложную программную систему, разработчики должны абстрагировать ее свойства с разных точек зрения, с помощью точных нотаций (систем обозначений) сконструировать адекватные модели, удостовериться, удовлетворяют ли они исходным требованиям, а затем реализовать модели на практике, постепенно пополняя систему новыми функциями.

К моделированию сложных систем приходится прибегать ввиду того, что мы не в состоянии постичь их одновременно во всей полноте. Способность человека к восприятию сложных вещей имеет свои естественные границы. В этом легко убедиться, обратившись к сфере строительства и архитектуры. Если нужно построить курятник, за работу можно приняться тотчас; если речь идет о новом доме, тогда, вероятно, потребуется хотя бы эскиз; наконец, при возведении небоскреба без точных расчетов и чертежей обойтись уже явно не удастся. Те же законы действуют и в мире программирования. Делая главную ставку на написание строк исходного кода или даже на использование форм Visual Basic, разработчик вряд ли сможет сколько-нибудь полно охватить структуру объемного проекта в целом. Предварительное моделирование, в свою очередь, позволяет проектировщику увидеть общую картину взаимодействий компонентов проекта без необходимости анализа особых свойств каждого компонента.

Графические (визуальные) модели представляют собой средства для визуализации, описания, проектирования и документирования архитектуры системы. Под архитектурой понимается набор основных правил, определяющих организацию системы:

- совокупность структурных элементов системы и связей между ними;
- поведение элементов системы в процессе их взаимодействия;
- иерархию подсистем, объединяющих структурные элементы;

– архитектурный стиль (используемые методы и средства описания архитектуры, а также архитектурные образцы).

Архитектура является многомерной, поскольку различные специалисты работают с различными ее аспектами. Например, при строительстве здания используются разные типы чертежей, представляющие различные аспекты архитектуры:

- планы этажей;
- вертикальные проекции;
- планы монтажа кабельной проводки;
- чертежи водопроводов, системы центрального отопления и вентиляции;
- вид здания на фоне местности (эскизы).

Архитектура ПО также предусматривает различные представления, служащие разным целям:

- представлению функциональных возможностей системы;
- отображению логической организации системы;
- описанию физической структуры программных компонентов в среде реализации;
- отображению структуры потоков управления и аспектов параллельной работы;
- описанию физического размещения программных компонентов на базовой платформе.

Архитектурное представление — это упрощенное описание (абстракция) системы с конкретной точки зрения, охватывающее определенный круг интересов и опускающее объекты, несущественные с данной точки зрения. Архитектурные представления концентрируют внимание только на элементах, значимых с точки зрения архитектуры. Архитектурно значимый элемент — это элемент, имеющий значительное влияние на структуру системы и ее производительность, надежность и возможность развития. Это элемент, важный для понимания системы.

Например, в состав архитектурно значимых элементов объектно-ориентированной архитектуры входят основные классы предметной области, подсистемы и их интерфейсы, основные процессы или потоки управления.

Архитектурные представления подобны сечениям различных моделей, выделяющим только важные, значимые элементы моделей.

Разработка модели архитектуры системы ПО промышленного характера на стадии, предшествующей ее реализации или обновлению, в такой же мере необходима, как и наличие проекта для строительства большого здания. Это утверждение справедливо как в случае разработки новой системы, так и при адаптации типовых продуктов класса R/3 или BAAN, в составе которых также имеются собственные средства моделирования. Хорошие модели являются основой взаимодействия участников проекта и гарантируют корректность архитектуры. Поскольку сложность систем повышается, важно располагать хорошими методами моделирования.

Хотя имеется много других факторов, от которых зависит успех проекта, но наличие строгого стандарта языка моделирования является весьма существенным.

Язык моделирования включает:

- элементы модели — фундаментальные концепции моделирования и их семантику;
- нотацию (систему обозначений) — визуальное представление элементов моделирования;
- руководство по использованию — правила применения элементов в рамках построения тех или иных типов моделей ПО.

Очевидно, что конечная цель разработки ПО - это не моделирование, а получение работающих приложений (кода). Диаграммы в конечном счете — это всего лишь наглядные изображения. Создание слишком большого количества диаграмм до начала программирования займет слишком много времени и не обеспечит построения готовой системы. Поэтому при использовании графических языков моделирования очень важно понимать,

чем это поможет, когда дело дойдет до написания кода. Можно привести следующие причины, побуждающие прибегать к их использованию:

– *получение общего представления о системе.* Графические модели помогают быстро получить общее представление о системе, сказать о том, какого рода абстракции существуют в системе и какие ее части нуждаются в дальнейшем уточнении;

– *общение с экспертами организации.* Графические модели об разуют внешнее представление системы и объясняют, что эта система будет делать;

– *изучение методов проектирования.* Множество людей отмечает наличие серьезных трудностей, связанных, например, с освоением объектно-ориентированных методов, и, в первую очередь, смену парадигмы. В некоторых случаях переход к объектно-ориентированным методам происходит относительно безболезненно. В других случаях при работе с объектами приходится сталкиваться с рядом препятствий, особенно в части максимального использования их потенциальных возможностей. Графические средства позволяют облегчить решение этой проблемы.

В процессе создания ПО, автоматизирующего деятельность некоторой организации, используются следующие виды моделей:

– модели деятельности организации (или модели бизнес-процессов):

– модели «AS-IS» («как есть»), отражающие существующее на момент обследования положение дел в организации и позволяющие понять, каким образом функционирует данная организация, а также выявить узкие места и сформулировать предложения по улучшению ситуации;

– модели «AS-TO-BE» («как должно быть»), отражающие представление о новых процессах и технологиях работы организации. Переход от модели «AS-IS» к модели «AS-TO-BE» может выполняться двумя способами: 1) совершенствованием существующих технологий на основе оценки их эффективности; 2) радикальным изменением

технологий и перепроектированием (реинжинирингом) бизнес-процессов.

– модели проектируемого ПО, которые строятся на основе модели «AS-TO-BE», уточняются и детализируются до необходимого уровня.

Состав моделей, используемых в каждом конкретном проекте, и степень их детальности в общем случае (как для структурного, так и для объектно-ориентированного подхода) зависят от следующих факторов:

- сложности проектируемой системы;
- необходимой полноты ее описания;
- знаний и навыков участников проекта;
- времени, отведенного на проектирование.

3.4 Структурные методы анализа и проектирования ПО

Структурные методы являются строгой дисциплиной системного анализа и проектирования. Методы структурного анализа и проектирования стремятся преодолеть сложность больших систем путем расчленения их на части («черные ящики») и иерархической организации этих «черных ящиков». Выгода в использовании «черных ящиков» заключается в том, что их пользователю не требуется знать, как они работают, необходимо знать лишь их входы и выходы, а также назначение (т.е. функции, которые они выполняет).

Таким образом, первым шагом упрощения сложной системы является ее разбиение на *«черные ящики»*, при этом такое разбиение должно удовлетворять следующим критериям:

- каждый «черный ящик» должен реализовывать единственную функцию системы;
- функция каждого «черного ящика» должна быть легко понимаема независимо от сложности ее реализации;
- связь между «черными ящиками» должна вводиться только при наличии связи между соответствующими функциями системы (например, в бухгалтерии один «черный ящик» необходим для расчета общей заработной

платы служащего, а другой для расчета налогов – необходима связь между этими «черными ящиками»: размер заработной платы требуется для расчета налогов);

– связи между «черными ящиками» должны быть простыми, насколько это возможно, для обеспечения независимости между ними.

Второй важной идеей, лежащей в основе структурных методов, является идея *иерархии*. Для понимания сложной системы недостаточно разбиения ее на части, необходимо эти части организовать определенным образом, а именно в виде иерархических структур. Все сложные системы Вселенной организованы в иерархии: от галактик до элементарных частиц. Человек при создании сложных систем также подражает природе. Любая организация имеет директора, заместителей по направлениям, иерархию руководителей подразделений, рядовых служащих.

Кроме того, структурные методы широко используют визуальное моделирование, служащее для облегчения понимания сложных систем.

Структурным анализом принято называть метод исследования системы, начинающий с ее общего обзора, который затем детализируется, приобретая иерархическую структуру со все большим числом уровней. Для таких методов характерно:

- разбиение системы на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 6—7);
- ограниченный контекст, включающий лишь существенные на каждом уровне детали;
- использование строгих формальных правил записи;
- последовательное приближение к конечному результату.

В структурном анализе основным методом разбиения на уровни абстракции является функциональная декомпозиция, заключающаяся в декомпозиции (разбиении) системы на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, те — на задачи и так далее до конкретных процедур.

При этом система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы «снизу вверх» от отдельных задач ко всей системе целостность теряется, возникают проблемы при описании информационного взаимодействия отдельных компонентов.

Все наиболее распространенные методы структурного подхода базируются на ряде общих принципов. Базовыми принципами являются:

– принцип «разделяй и властвуй» — принцип решения трудных проблем путем разбиения их на множество меньших независимых задач, легких для понимания и решения;

– принцип иерархического упорядочения — принцип организации составных частей системы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

Выделение двух базовых принципов не означает, что остальные принципы являются второстепенными, поскольку игнорирование любого из них может привести к нежелательным последствиям (вплоть до неудачного завершения проекта). Основными из этих принципов являются:

– принцип абстрагирования — выделение существенных аспектов системы и отвлечение от несущественных;

– принцип непротиворечивости — обоснованность и согласованность элементов системы;

– принцип структурирования данных — данные должны быть структурированы и иерархически организованы.

В структурном анализе и проектировании используются различные модели, описывающие:

- 1) функциональную структуру системы;
- 2) последовательность выполняемых действий;
- 3) передачу информации между функциональными процессами;
- 4) отношения между данными.

Наиболее распространенными моделями первых трех групп являются:

- функциональная модель SADT (Structured Analysis and Design Technique);
- модель IDEF3;
- DFD (Data Flow Diagrams) - диаграммы потоков данных.

Модель «сущность — связь» (ERM — Entity-Relationship Model), описывающая отношения между данными, традиционно используется в структурном анализе и проектировании, однако, по существу, представляет собой подмножество объектной модели предметной области.

3.4.1. Метод функционального моделирования SADT (IDEF0)

Общие сведения

Метод SADT разработан Дугласом Россом (SoftTech, Inc.) в 1969 г. для моделирования искусственных систем средней сложности. Данный метод успешно использовался в военных, промышленных и коммерческих организациях США для решения широкого круга задач, таких, как долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, разработка ПО для оборонных систем, управление финансами и материально-техническим снабжением и др. Метод SADT поддерживается Министерством обороны США, которое было инициатором разработки семейства стандартов IDEF (Icam DEFinition), являющегося основной частью программы ICAM (интегрированная компьютеризация производства), проводимой по инициативе ВВС США. Метод SADT реализован в одном из стандартов этого семейства — IDEF0, который был утвержден в качестве федерального стандарта США в 1993 г., его подробные спецификации можно найти на сайте <http://www.idef.com>.

Метод SADT представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи

между этими действиями. Основные элементы этого метода основываются на следующих концепциях:

- графическое представление блочного моделирования. Графика блоков и дуг SADT-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описывается посредством интерфейсных дуг, выражающих «ограничения», которые, в свою очередь, определяют выполнение и управление функций;

- строгость и точность. Выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика. Правила SADT включают: ограничение количества блоков на каждом уровне декомпозиции (правило 3—6 блоков — ограничение мощности краткосрочной памяти человека), связность диаграмм (номера блоков), уникальность меток и наименований (отсутствие повторяющихся имен), синтаксические правила для графики (блоков и дуг), разделение входов и управлений (правило определения роли данных);

- отделение организации от функции, т.е. исключение влияния административной структуры организации на функциональную модель.

Метод SADT может использоваться для моделирования самых разнообразных процессов и систем. В существующих системах метод SADT может быть использован для анализа функций, выполняемых системой, и указания механизмов, посредством которых они осуществляются.

Состав функциональной модели

Результатом применения метода SADT является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга. Диаграммы — главные компоненты модели, все функции организации и интерфейсы на них представлены как блоки и дуги соответственно. Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация входит в блок сверху, в то время как входная

информация, которая подвергается обработке, показана с левой стороны блока, а результаты (выход) показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рис. 3.4).

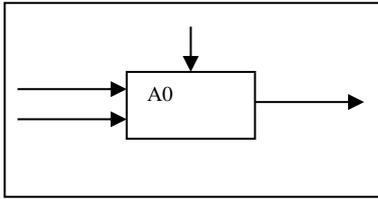


Рис 3.4 – Функциональный блок и интерфейсные дуги

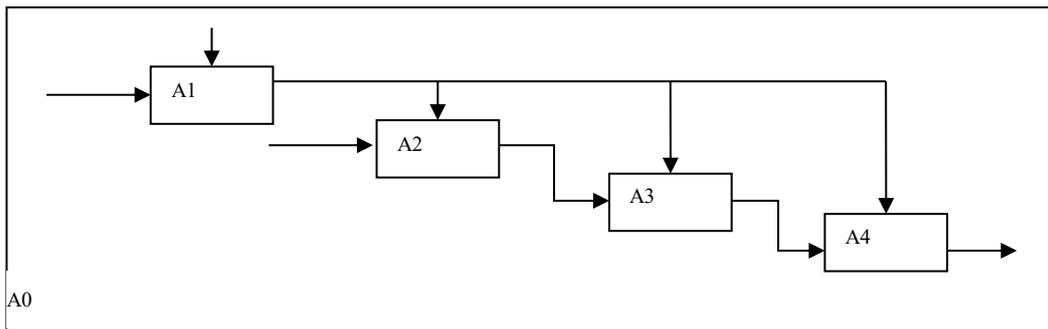
Одной из наиболее важных особенностей метода SADT является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель.

На рис. 3.5, где приведены четыре диаграммы и их взаимосвязи, показана структура SADT-модели. Каждый компонент модели может быть декомпозирован на другой диаграмме. Каждая диаграмма иллюстрирует «внутреннее строение» блока на родительской диаграмме.

Общее представление



Более детальное представление



Верхняя диаграмма является родительской для нижней диаграммы

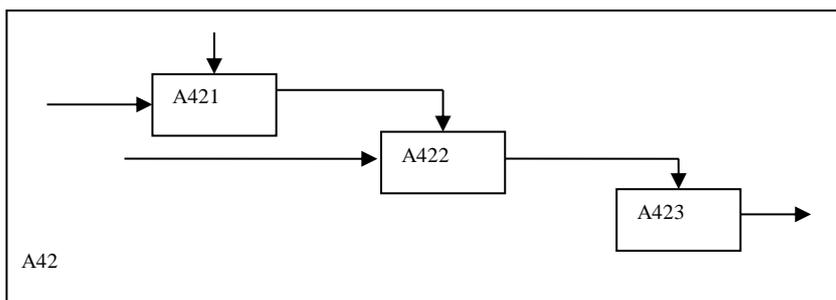
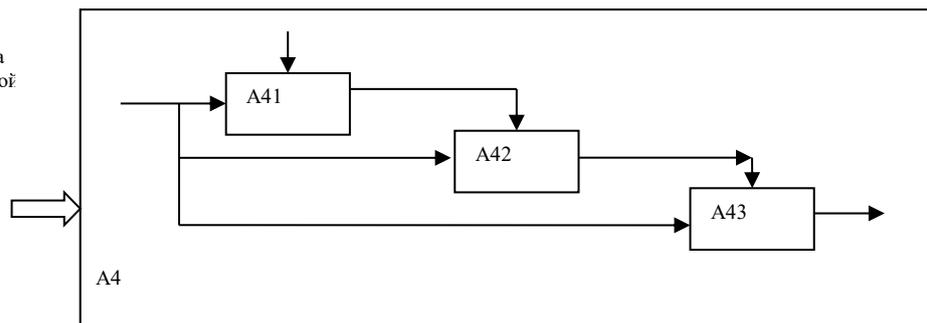


Рис. 3.5 – Структура SADT-модели. Декомпозиция диаграмм

Построение SADT-модели

Построение SADT-модели заключается в выполнении следующих действий:

- сбор информации об объекте, определение его границ;
- определение цели и точки зрения модели;

- построение, обобщение и декомпозиция диаграмм;
- критическая оценка, рецензирование и комментирование.

Построение диаграмм начинается с представления всей системы в виде простейшего компонента — одного блока и дуг, изображающих интерфейсы с функциями вне системы. Поскольку единственный блок отражает систему как единое целое, имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг — они также соответствуют полному набору внешних интерфейсов системы в целом.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки определяют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых показана как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом в целях большей детализации.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию. Кроме того, модель не может опустить какие-либо элементы, т.е., как уже отмечалось, родительский блок и его интерфейсы обеспечивают контекст. К нему нельзя ничего добавить и из него не может быть ничего удалено.

Модель SADT представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые изображены в виде блоков. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Каждая детальная диаграмма является декомпозицией блока из диаграммы предыдущего уровня. На каждом шаге декомпозиции диаграмма предыдущего уровня называется родительской для более детальной диаграммы.

Синтаксис диаграмм определяется следующими правилами:

- диаграммы содержат блоки и дуги;
- блоки представляют функции;

- блоки имеют доминирование (выражающееся в их ступенчатом расположении, причем доминирующий блок располагается в верхнем левом углу диаграммы);
- дуги изображают наборы объектов, передаваемых между блоками;
- дуги изображают взаимосвязи между блоками: выход-управление, выход-вход, обратная связь по управлению, обратная связь по входу, ВЫХОД-МЕХАНИЗМ.

На рисунках 3.6—3.8 приведены различные варианты выполнения функций и соединения дуг с блоками.

Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма изображают одну и ту же часть системы.

Некоторые дуги присоединены к блокам диаграммы обоими концами, у других же один конец остается неприсоединенным. Неприсоединенные дуги соответствуют входам, управлениям и выходам родительского блока. Источник или получатель этих пограничных дуг может быть обнаружен только на родительской диаграмме. Неприсоединенные концы должны соответствовать дугам на исходной диаграмме. Все граничные дуги должны продолжаться на родительской диаграмме, чтобы она была полной и непротиворечивой.

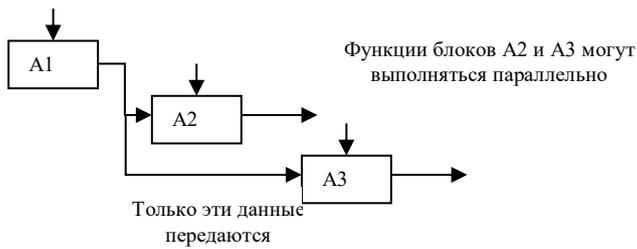


Рис. 3.6 – Одновременное выполнение функций.

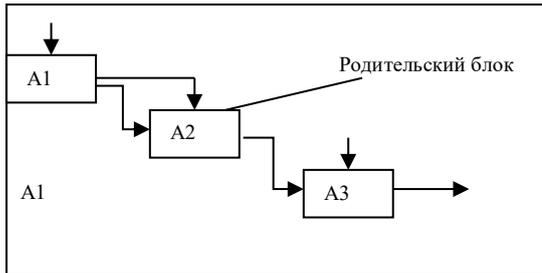
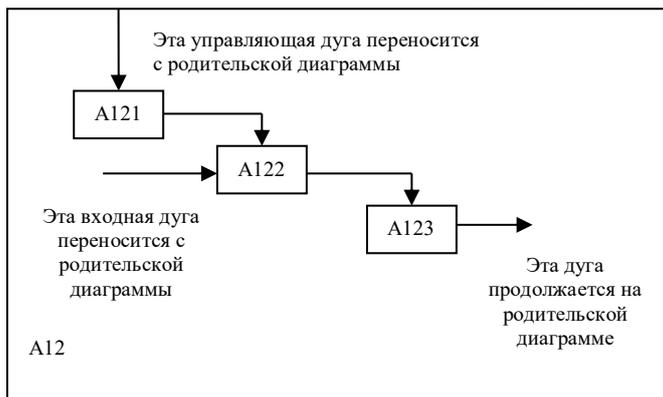


Рис. 3.7 – Пример родительского блока



Более детальное представление

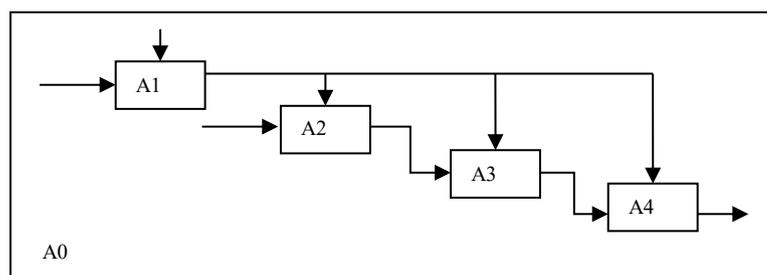


Рис. 3.8 – Соответствие интерфейсных дуг родительской (рис. 3.7.) и детальной диаграмм

На SADT-диаграммах не указаны явно ни последовательность, ни время. Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть изображены с помощью дуг. Обратные связи могут выступать в виде комментариев, замечаний, исправлений и т.д.).

Как было отмечено, механизмы (дуги с нижней стороны) показывают средства, с помощью которых осуществляется выполнение функций. Механизм может быть человеком, компьютером или любым другим устройством, которое помогает выполнять данную функцию.

Каждый блок на диаграмме имеет свой номер. Блок любой диаграммы может быть далее описан диаграммой нижнего уровня, которая, в свою очередь, может быть далее детализирована с помощью необходимого числа диаграмм. Таким образом, формируется иерархия диаграмм.

Для того чтобы указать положение любой диаграммы или блока в иерархии, используются номера диаграмм. Например, A21 является диаграммой, которая детализирует блок A21 на диаграмме A2. Аналогично, диаграмма A2 детализирует блок A2 на диаграмме A0, которая является самой верхней диаграммой модели.

Типы связей между функциями

Одним из важнейших моментов при моделировании бизнес-процессов организации с помощью метода SADT является точная согласованность типов связей между функциями. Различают по крайней мере связи семи типов (в порядке возрастания их относительной значимости):

- случайная;
- логическая;
- временная;
- процедурная;
- коммуникационная;
- последовательная;
- функциональная.

Случайная связь – показывает, что конкретная связь между функциями незначительна или полностью отсутствует.

Это относится к ситуации, когда имена данных на SADT – дугах в одной диаграмме имеют слабую связь друг с другом.

Логическая связь – данные и функции собираются вместе благодаря тому, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается.

Временная связь – представляет функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно.

Процедурная связь – функции сгруппированы вместе благодаря тому, что они выполняются в течение одной и той же части цикла или процесса.

Коммуникационная связь – функции группируются благодаря тому, что они используют одни и те же исходные данные и/или производят одни и те же выходные данные.

Последовательная связь – выход одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем в рассмотренных выше случаях, поскольку моделируются причинно-следственные зависимости.

Функциональная связь – все элементы функции влияют на выполнение одной и той же функции. Диаграмма, являющаяся чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связи. Одним из способов определения функционально связанных диаграмм является рассмотрение двух блоков, связанных через управляющие дуги.

В таблице 3.4 ниже представлены все типы связей. Важно отметить, что уровни 4-6 устанавливают типы связей, которые разработчики считают важнейшими для получения диаграмм хорошего качества.

Таблица 3.4 – Типы связей.

Уровень значимости	Тип связи	Характеристика типа связи	
		Для функций	Для данных
0	Случайная	Случайная	Случайная
1	Логическая	Функции одного и того же множества или типа (например, «редактировать все входы»)	Данные одного и того же множества или типа
2	Временная	Функции одного и того же периода времени (например, «операции инициализации»)	Данные, используемые в каком либо временном интервале
3	Процедурная	Функции, работающие в одной и той же фазе или итерации, например, «первый проход компилятора»	Данные используемые во время одной и той же фазы или итерации
4	Коммуникационная	Функции, использующие одни и те же данные	Данные, на которые воздействует одна и та же деятельность
5	Последовательная	Функции, выполняющие последовательное преобразование одних и тех же данных	Данные, преобразуемые последовательными функциями
6	Функциональная	Функции, объединяемые для выполнения одной функции	Данные, связанные с одной функцией

3.4.2 Моделирование потоков данных

Диаграммы потоков данных (DFD) являются основным средством моделирования функциональных требований к проектируемой системе.

С их помощью эти требования представляются в виде иерархии функциональных компонентов (процессов), связанных потоками данных. Главная цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Диаграммы потоков данных известны очень давно. В фольклоре упоминается пример использования DFD для реорганизации переполненного клерками офиса, относящийся к 20-м гг. осуществлявший реорганизацию консультант обозначил кружком каждого клерка, а стрелкой - каждый документ, передаваемый между ними. Используя такую диаграмму, он предложил схему реорганизации, в соответствии с которой два клерка, обменивающихся множеством документов, были посажены рядом, а клерки с

малым взаимодействием были посажены на большом расстоянии друг от друга. Так появилась первая модель, представляющая собой потоковую диаграмму – предвестника DFD.

Для построения DFD традиционно используются две различные нотации, соответствующие методам Йордана и Гейна - Сэрсона. Эти нотации незначительно отличаются друг от друга графическим изображением символов.

Далее при построении будет использоваться нотация Гейна –Сэрсона.

В соответствии с данными методами модель системы определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут уровень декомпозиции, на котором процессы становятся элементарными и детализировать их далее невозможно.

Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те, в свою очередь, преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям – потребителям информации.

Состав диаграмм потоков данных

Основными компонентами диаграмм потоков данных являются:

- внешние сущности (рис.3.9);
- системы и подсистемы (рис.3.10);
- процессы (рис. 3.11);
- накопители данных (рис.3.12);

– потоки данных (рис.3.13).

Внешняя сущность представляет собой материальный объект или физическое лицо, представляющие собой источник или приемник информации, например, заказчики, персонал, поставщики, клиенты, склад. Они находятся за пределами анализируемой системы.

Внешняя сущность обозначается квадратом, расположенным как бы над диаграммой и бросающим на нее тень для того, чтобы можно было выделить этот символ среди других обозначений.



Рис.3.9 – Графическое изображение внешней сущности

При построении модели сложной ЭИС она может быть представлена в общем виде на так называемой *контекстной диаграмме* в виде одной системы как единого целого либо может быть *декомпозирована* на ряд *подсистем*.

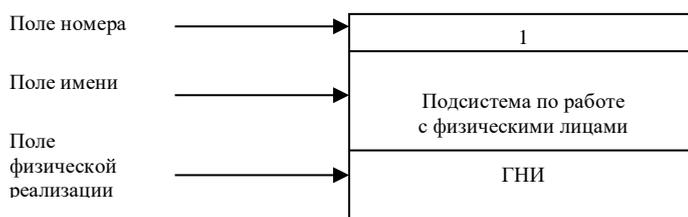


Рис.3 10 – Изображение системы (подсистемы)

Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.

Физически процесс может быть реализован различными способами: это может быть подразделение организации, выполняющее обработку входных

документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.

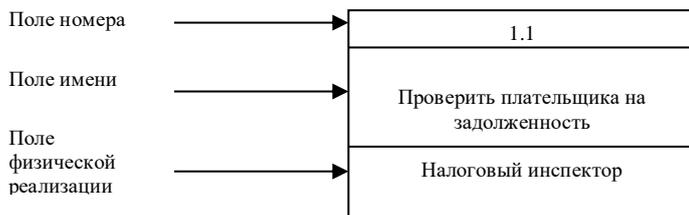


Рис.3.10 – Графическое изображение процесса

Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме, за которым следуют существительные в винительном падеже. Использование таких глаголов как «обработать», «модернизировать», «отредактировать» означает недостаточно глубокое понимание процесса и требует дальнейшего анализа.

Информация в поле физической реализации, показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

Накопитель данных – это абстрактное устройство, для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Накопитель данных может быть реализован физически в виде микрофиши, ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д. Накопитель данных на диаграмме потоков данных идентифицируется в виде буквы «D» и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Накопитель данных в общем случае является прообразом будущей базы данных, и описание хранящихся в нем данных может быть увязано с информационной моделью (ERD).

Поток данных определяет информацию, передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами и дискетами, переносимыми с одного компьютера на другой и т.д.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока. Каждый поток данных имеет свое имя, отражающее содержание.

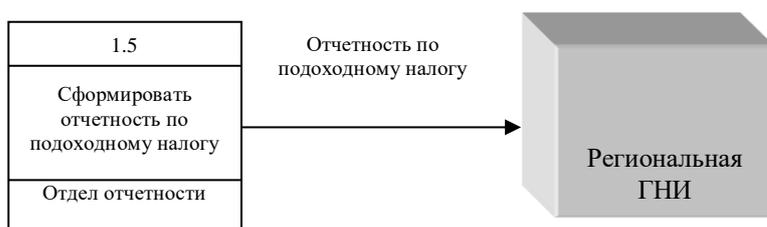


Рис. 3.11 – Поток данных между процессом и внешней сущностью

Построение иерархии потоков данных

Главная цель построения иерархии DFD заключается в том, чтобы сделать требования к системе ясными и понятными на каждом уровне детализации, а также разбить эти требования на части с точно определенными отношениями между ними. Для достижения этого целесообразно пользоваться следующими рекомендациями:

- размещать на каждой диаграмме от 3 до 6-7 процессов. Верхняя граница соответствует человеческим возможностям одновременного восприятия и понимания структуры сложной системы с множеством внутренних связей, нижняя граница выбрана по соображениям здравого смысла: нет необходимости детализировать процесс диаграммой, содержащей всего один или два процесса.
- не загромождать диаграммы не существенными на данном уровне деталями.

– декомпозицию потоков данных осуществлять параллельно с декомпозицией процессов. Эти две работы должны выполняться одновременно, а не после завершения другой.

Выбирать ясные, отражающие суть дела имена процессов и потоков, при этом стараться не использовать аббревиатуры.

Первым шагом при построении иерархии DFD является построение контекстных диаграмм. Обычно при проектировании относительно простых систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы. Перед построением контекстной DFD необходимо проанализировать внешние события (внешние сущности), оказывающие влияние на функционирование системы. Количество потоков на контекстной диаграмме должно быть по возможности небольшим, поскольку каждый из них может быть в дальнейшем разбит на несколько потоков на следующих уровнях диаграммы.

Для проверки контекстной диаграммы можно составить список событий. Список событий должен состоять из описаний действий внешних сущностей (событий) и соответствующих реакций на события системы. Каждое событие должно соответствовать одному (или более) потоку данных: входные потоки интерпретируются как воздействия, а выходные потоки - как реакция системы на входные потоки.

Для сложных систем строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистемы.

После построения контекстных диаграмм полученную модель следует проверить на полноту исходных данных об объектах системы и

изолированность объектов (отсутствие информационных связей с другими объектами).

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи DFD. Это можно сделать путем построения диаграммы для каждого события. Каждое событие представляется в виде процесса с соответствующими входными и выходными потоками, накопителями данных, внешними сущностями и ссылки на другие процессы для описания связей между этим процессом и его окружением. Затем все построенные диаграммы сводятся в одну диаграмму нулевого уровня.

Каждый процесс на DFD, в свою очередь, может быть детализирован при помощи DFD или (если процесс элементарный) спецификации. При детализации должны выполняться следующие правила:

- *правило балансировки* – при детализации подсистемы или процесса детализирующая диаграмма в качестве внешних источников или приемников данных может иметь только те компоненты (подсистемы, процессы, внешние сущности, накопители данных), с которыми имеют информационную связь детализируемая подсистема или процесс на родительской диаграмме.

- *правило нумерации* – при детализации процессов должна поддерживаться их иерархическая нумерация. Например, процессы, детализирующие процесс с номером 12, получают номера 12.1, 12.2, 12.3 и т.д.

Спецификация процесса должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.

Спецификация является конечной вершиной иерархии DFD. Решение о завершении детализации процесса и использовании спецификации принимается аналитиком исходя из следующих критериев:

- наличия у процесса относительно небольшого количества входных и выходных потоков данных (2-3).

- возможности описания преобразования данных процессом в виде последовательного алгоритма.

– выполнения процессом единственной логической функции преобразования входной информации в выходную.

– возможности описания логики процесса при помощи спецификации небольшого объема (не более 20-30 строк).

Спецификации должны удовлетворять следующим требованиям:

– для каждого процесса нижнего уровня должна существовать одна и только одна спецификация.

– спецификация должна определять способ преобразования входных потоков в выходные.

– нет необходимости (по крайней мере на стадии формирования требований) определять метод реализации этого преобразования.

– спецификация должна стремиться к ограничению избыточности – не следует переопределять то, что уже было определено на диаграмме.

– набор конструкций для построения спецификаций должен быть простым и понятным.

Фактически спецификации представляют собой описания алгоритмов задач, выполняемых процессами. Спецификации содержат:

1. Номер и/или имя процесса.
2. Списки входных и выходных данных.
3. Тело (описание процесса), являющееся спецификацией алгоритма или операции, трансформирующей входные потоки данных в выходные.

Известно большое количество методов, позволяющих описать тело процесса, соответствующие этим методам языки могут варьироваться от структурированного естественного языка или псевдокода до визуальных языков проектирования.

Структурированный естественный язык применяется для читабельного, достаточно строгого описания спецификаций процессов. Он представляет собой разумное сочетание строгости языка программирования и читабельности естественного языка и состоит из подмножества слов,

организованных в определенные логические структуры, арифметических выражений и диаграмм.

В *состав языка* входят следующие основные символы:

- глаголы, ориентированные на действие и применяемые к объектам;
- термины, определенные на любой стадии проекта ПО (например, задачи, процедуры, символы данных и т.д.);
- предлоги и союзы, используемые в логических отношениях;
- общеупотребительные математические, физические и технические термины;
- арифметические уравнения;
- таблицы, диаграммы, графы;
- комментарии.

К *управляющим структурам* языка относятся последовательная конструкция, конструкция выбора, итерация (цикл).

При использовании структурированного естественного языка приняты следующие *соглашения*:

- логика процесса выражается в виде комбинации последовательных конструкций, конструкций выбора и итераций;
- глаголы должны быть активными, недвусмысленными и ориентированными на целевое действие (заполнить, вычислить, извлечь, а не модернизировать, обработать и т.д.);
- логика процесса должна быть выражена четко и недвусмысленно.

При построении иерархии DFD переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается с помощью структур данных. Для каждого потока данных формируется список всех его элементов данных, затем элементы данных объединяются в структуры данных, соответствующие более крупным объектам данных (например, строкам документов или объектам предметной области).

Каждый объект должен состоять из элементов, являющихся его атрибутами. Структуры данных могут содержать альтернативы, условные вхождения и итерации. *Условное вхождение* показывает, что данный компонент может отсутствовать в структуре (например, структура «данные о страховании» для объекта «служащий»). *Альтернатива* означает, что в структуру может входить один из перечисленных элементов. *Итерация* предусматривает вхождение любого числа элементов в указанном диапазоне (например, элемент «имя ребенка» для объекта «служащий»).

Для каждого элемента данных может указываться его тип (непрерывные и дискретные данные). Для *непрерывных данных* могут указываться единицы измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для *дискретных* данных может указываться таблица дискретных значений.

После построения законченной модели системы ее необходимо верифицировать (проверить на полноту и согласованность). В полной модели все ее объекты (подсистемы, процессы, потоки данных) должны быть подробно описаны и детализированы. Выявленные не детализированные объекты следует детализировать, вернувшись на предыдущие шаги разработки. В согласованной модели для всех потоков данных и накопителей данных должно выполняться правило сохранения информации: все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

3.4.3 Сравнительный анализ SADT-моделей и диаграмм потоков данных

Итак, практически во всех методах структурного подхода (структурного анализа) на стадии формирования требований к ПО используются две группы средств моделирования:

– диаграммы, иллюстрирующие функции, которые система должна выполнять, и связи между этими функциями - DFD или SADT (IDEF0);

- диаграммы, моделирующие данные и их отношения (ERD).

Таким образом, наиболее существенное различие между разновидностями структурного анализа заключается в средствах функционального моделирования. С этой точки зрения все разновидности структурного анализа могут быть разбиты на две группы – использующие DFD (в различных нотациях) и использующие SADT - модели. Соотношение применения этих двух разновидностей структурного анализа в существующих CASE - средствах составляет 90% для DFD и 10% для SADT.

Сравнительный анализ этих двух разновидностей методов структурного анализа проводится по следующим параметрам:

- адекватность средств решаемым задачам;
- согласованность с другими средствами структурного анализа;
- интеграция с последующими стадиями ЖЦ ПО (прежде всего со стадией проектирования).

Адекватность средств решаемым задачам. Модели SADT используются для моделирования организационных систем. С другой стороны, не существует никаких принципиальных ограничений на использование DFD в качестве средства построения статистических моделей деятельности организации. Метод SADT успешно работает только при описании стандартизированных бизнес-процессов в зарубежных корпорациях, поэтому он и принят в США в качестве типового.

Если же речь идет не о системах вообще, а о ЭИС, то здесь DFD вне конкуренции. SADT - диаграммы оказываются значительно менее выразительными и удобными при моделировании ЭИС.

Согласованность с другими средствами структурного анализа. Главным достоинством любых моделей является возможность их интеграции с моделями других типов. В данном случае речь идет о согласованности функциональных моделей со средствами моделирования данных. Согласование SADT - модели с ERD практически невозможно или носит искусственный характер. В свою очередь, DFD и ERD взаимно дополняют

друг друга и являются согласованными, поскольку в DFD присутствует описание структур данных, непосредственно используемое для построения ERD.

Интеграция с последующими стадиями жизненного цикла ПО. Важная характеристика модели - ее совместимость с моделями последующих стадий жизненного цикла ПО (прежде всего стадии проектирования, непосредственно следующей за стадией формирования требований и опирающейся на ее результаты).

DFD могут быть легко преобразованы в модели проектируемой системы.

Одним из основных критериев выбора того или иного метода является степень владения им со стороны консультанта или аналитика, грамотность выражения своих мыслей на языке моделирования. В противном случае в моделях, построенных с использованием любого метода, будет невозможно разобраться.

3.4.4 Функциональные модели, используемые на стадии проектирования

Функциональные модели, используемые на *стадии проектирования ПО*, предназначены для описания функциональной структуры проектируемой системы. Построенные ранее DFD при этом уточняются, расширяются и дополняются новыми конструкциями.

Так, например, для DFD переход от модели бизнес-процессов организации к *модели системных процессов* может происходить следующим образом:

- внешние сущности на контекстной диаграмме заменяются или дополняются техническими устройствами (например, рабочими станциями, принтерами);
- для каждого потока данных определяется, посредством каких технических устройств информация передается или производится;

- процессы на диаграмме нулевого уровня заменяются соответствующими процессорами – обрабатывающими устройствами (процессорами могут быть как технические устройства – ПК конечных пользователей, рабочие станции, серверы баз данных, так и служащие - исполнители);

- определяется и изображается на диаграмме тип связи между процессорами (например, локальная сеть - LAN);

- определяются задачи для каждого процессора (приложения, необходимые для работы системы), для них строятся соответствующие диаграммы. Определяется тип связи между задачами;

- устанавливаются ссылки между задачами и процессами диаграмм потоков данных следующих уровней.

Иерархия экранных форм моделируется с помощью **диаграмм последовательностей экранных форм**. Совокупность таких диаграмм представляет собой абстрактную модель пользовательского интерфейса системы, отражающую последовательность появления экранных форм в приложении. Построение диаграмм последовательностей экранных форм выполняется следующим образом:

- на DFD выбираются интерактивные процессы нижнего уровня. Интерактивные процессы нуждаются в пользовательском интерфейсе, поэтому нужно определить экранную форму для каждого такого процесса;

- форма диаграммы изображается в виде прямоугольника для каждого интерактивного процесса на нижнем уровне диаграммы;

- определяется структура меню. Для этого интерактивные процессы группируются в меню (либо так же как в модели процессов, либо другим способом - по функциональным признакам или в зависимости от принадлежности к определенным объектам);

- формы с меню изображаются над формами, соответствующими интерактивным процессам, и соединяются с ними переходами в виде стрелок, направленных от меню к формам;

– определяется верхняя форма (главная форма приложения), связывающая все формы с меню.

Техника структурных карт используется на стадии проектирования для описания структурных схем программ. При этом наиболее часто применяются две техники: структурные карты Константайна (для описания отношений между модулями) и структурные карты Джексона (для описания внутренней структуры модулей, являющихся базовыми строительными блоками программной системы). В настоящее время структурные карты применяются сравнительно редко.

3.5. Моделирование данных

Основные понятия

Цель моделирования данных состоит в обеспечении разработчика ЭИС концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD), нотация которых впервые была введена Питером Ченом в 1976 г. Базовым понятием ERD являются:

Сущность (Entity) – реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области.

Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности. Каждая сущность должна обладать некоторыми свойствами:

- иметь уникальное имя;
- обладать одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через связь;

– обладать одним или несколькими атрибутами, которые однозначно идентифицируют каждый экземпляр сущности.

Каждая сущность может обладать любым количеством связей с другими сущностями модели.

Связь (Relationship) - поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь – это ассоциация между сущностями, при которой каждый экземпляр одной сущности ассоциирован с произвольным (в том числе и нулевым) количеством экземпляров второй сущности, и наоборот.

Атрибут - любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Экземпляр атрибута – это определенная характеристика определенного элемента множества. Экземпляр атрибута определяется типом характеристики и ее значением, называемым значением атрибута. На диаграмме «сущность-связь» атрибуты ассоциируются с конкретными сущностями.

В учебной практике не является обязательным представление модели данных, достаточно использовать модели, построенные студентом ранее в рамках выполнения курсового проектирования, учебной исследовательской и научной работ.

По этой причине в данном пособии не приводится материал по проектированию баз данных.

4 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

В этот класс программного обеспечения входят программы и программные комплексы, обеспечивающие технологию разработки, отладки и внедрения создаваемых программных продуктов («программы для программ») – как относящихся к системному ПО, так и к прикладному.

Можно (с достаточной степенью условности) выделить два направления развития информационных технологий программирования (рис. 1.2):

– средства для создания программных приложений (RAD – Rapid Application Development), работающих в различных операционных системах (системы программирования и инструментальные среды разработки на базе различных языков программирования);

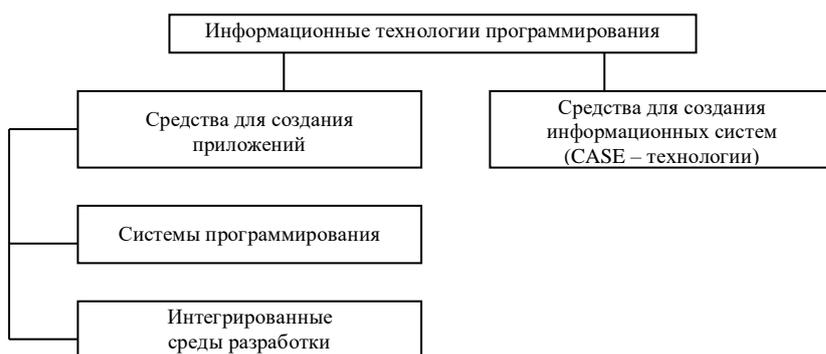


Рис. 4.1 – Классификация технологий программирования

– средства компьютерной поддержки разработки программного обеспечения и автоматизации создания сложных информационных систем (Computer Aided Software Engineering, CASE- технологии).

Системы программирования включают в себя:

- компилятор или интерпретатор языка программирования;
- отладчик и средства оптимизации кода программ;
- набор стандартных библиотек;
- редактор связей;
- справочные системы.

Примеры: Turbo Pascal, Borland Pascal (Borland), MS Visual Basic, MS Visual C++, Visual Fortran (сначала Microsoft, затем DIGITAL).

Инструментальные интегрированные среды разработки (IDE – Integrated Development Environment) объединяют единым графическим интерфейсом набор средств коллективной разработки программ для комплексного их применения на всех технологических этапах создания программы. Основное назначение – повышение производительности труда программистов, автоматизация создания кодов программ, разработка приложений для архитектуры клиент-сервер, запросов и отчетов. Они обладают широкой функциональностью, совместимы со многими распространенными платформами и стандартами программирования.

Примеры: Oracle JDeveloper, Jbuilder 3 Enterprise (Borland), VisualAge for Java (IBM), VisualCafe for Java (Symantec), Forte for Java (Sun Microsystems), Kawa (Tec-Tools) – все на основе языка объектно-ориентированного программирования Java, Delphi (Borland) – на основе Object Pascal, C++ Builder (Borland) – на основе C++, MS Developer Studio.

Современное проектирование больших и сложных информационных систем невозможно без *CASE – технологий*. Дело в том, что самой большой проблемой, которую приходится решать программной инженерии, является сложность ПО. При увеличении размера ПО наблюдается нелинейный рост его сложности, появляются трудности в процессе общения между разработчиками, что ведет к ошибкам в продукте, превышению стоимости разработки, затягиванию выполнения графиков работ. Сложность структуры затрудняет развитие ПО и добавление новых функций.

CASE – технологии выделяют логические процессы преобразования информации, определяют группы элементов данных и их хранилища (базы данных), предоставляют специальные графические средства для изображения различного типа моделей, используемых при описании сложной системы: диаграмму потоков данных (DFD – Data Flow Diagrams), устанавливающую связь источников информации с потребителями, диаграмму «сущность–связь» (ERD – Entity Relationship Diagrams), являющуюся информационной моделью рассматриваемой предметной области и другие виды диаграмм.

CASE – технологии обеспечивают автоматическую генерацию кодов программ на основе их спецификаций, проверку корректности описания моделей данных и схем потоков данных, документирование программ в соответствии с принятыми стандартами, тестирование и отладку программ. В рамках CASE – технологий происходит поддержка полного жизненного цикла программного продукта с обеспечением эволюционности его развития, обеспечивается функциональная целостность проекта, поддержка одновременной работы групп разработчиков, возможность разработки приложений «клиент-сервер» требуемой конфигурации. Они обладают открытой архитектурой и возможностью экспорта/импорта.

Примеры: ProKit Workbench (McDonnell Douglas Information Systems), Design/IDEF (MetaSoftware), CASEАналитик («Эйтекс»), ERwin, BPwin (LogicWorks), Silverrun (Silverrun Technologies), Oracle Designer, Rational Enterprise Suite 2000 (Rational Software).

В заключение отметим, что современные CASE-системы – это средства разработки не только программных систем, но и организационно-управляющих систем, т.е. эту аббревиатуру можно расшифровать и как Computer Aided System Engineering. В рассматриваемом контексте CASE-технология фактически представляет собой совокупность методологий проектирования, моделирования, анализа и реорганизации бизнес-процессов (BPR – Business Process Reengineering, реинжиниринг бизнес-процессов) предприятий, поддержанную комплексом взаимоувязанных средств автоматизации.

В состав бизнес-модели должны входить следующие базовые компоненты:

- бизнес-функции, описывающие ЧТО делает бизнес;
- бизнес-процессы, описывающие КАК предприятие выполняет свои бизнес-функции;
- организационная структура, определяющая ГДЕ исполняются бизнес-функции и бизнес-процессы;

- роли, определяющие КТО исполняет бизнес-процессы;
- фразы, определяющие КОГДА (в какой последовательности) должны быть внедрены те или иные бизнес-функции;
- правила, определяющие связь между ЧТО, КАК, ГДЕ, КОГДА и КТО (рис. 4.2).

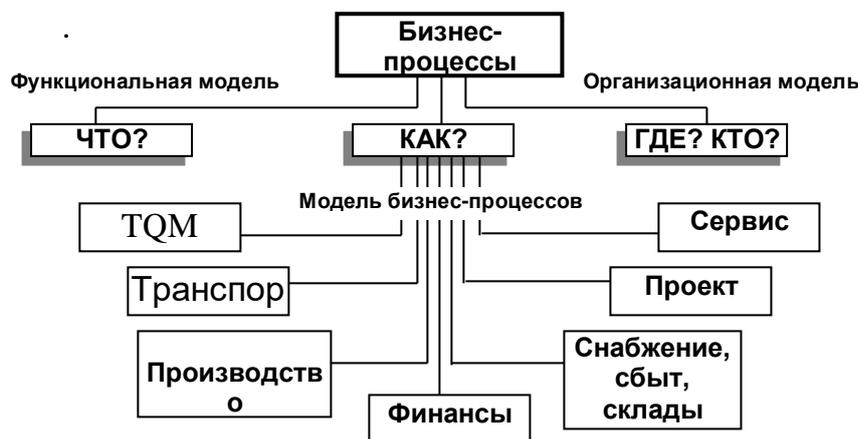


Рис. 4.2 – Структура бизнес-модели.

TQM (Total Quality Management) – управление качеством

Для бизнес-аналитиков CASE – это инструментарий, заменяющий им бумагу и карандаш на компьютер для автоматизации анализа и проектирования бизнес-процессов.

Принципиальное отличие CASE-среды для бизнес-процессов от соответствующей среды для программного обеспечения заключается в том, что хоть в обоих случаях и решаются задачи анализа и проектирования, задача генерации для бизнес-процесса гораздо сложнее, т.к. ПО является лишь одним из его компонентов.

В рамках учебной практики необходимо провести обзор программных средств различного назначения (из перечисленных выше в этом разделе) для целей выполнения задания по учебной практике.

5 ТЕМАТИКА ПРАКТИКИ

Учебная практика студентов заочной формы обучения выполняется по материалам, собранным студентом в течение предыдущих семестров во время выполнения учебной исследовательской и научной работ студента.

Студенты заочной формы обучения могут выбрать темы учебной практики самостоятельно в зависимости от своих профессиональных предпочтений.

В соответствии с квалификационной характеристикой бакалавра подготовки 09.03.03 «Прикладная информатика» возможны перечисленные ниже основные направления тематики практики.

1) Прикладная научная работа в области автоматизации информационных процессов и применении математических моделей и методов в управлении экономическими объектами.

2) Проектирование и разработка экономических информационных систем, обеспечивающих *обработку информации* по комплексу (комплексам) задач и функций управления процессами и ресурсами в различных предметных областях экономики.

3) Разработка систем *информационной поддержки принятия решения* для менеджеров различного уровня.

4) Разработка информационно-поисковых систем решения задач определения спроса и предложения на товары и услуги, производства товаров и решения актуальных задач экономики и менеджмента.

5) Разработка информационных *систем управления* различными экономическими объектами.

6) Разработка Web-приложений.

7) Разработка АРМ экономиста, бухгалтера, менеджера, маркетолога и др.

8) Разработка систем электронной торговли.

9) Разработка проекта внедрения ИС.

10) Разработка системы информационной безопасности для ИС.

При выполнении практики следует применять новые информационные технологии и современные методы проектирования.

Целесообразно, чтобы заданием на практику был охвачен *комплекс задач* предметной области не менее 3 и не более 5. Под задачей следует понимать алгоритм или совокупность алгоритмов обработки и формирования результатной информации. При этом объем охвата ИС и ее компонентов в качестве объектов проектирования может быть от автоматизации отдельного бизнес-процесса до ИС масштаба подразделения организации или небольшой компании. В первом случае большее внимание уделяется алгоритмам и программированию, во втором — системному проектированию.

Название темы учебной практики должно быть краткими и состоять из двух частей: в первой части указывается суть работы, во второй — объект работы. Например, «Автоматизация учета основных средств в ООО «Радиус».

Название темы учебной практики может состоять из трех или более частей при необходимости уточнить задачу, среду разработки и т. п.

Ниже приведены примеры тем учебной практики:

1. Автоматизированная система оценки инвестиционного проекта.
2. Разработка Web-приложения к подсистеме (с указанием конкретной подсистемы и системы).
3. Информационная система учета, контроля, анализа и принятия решений в области долгосрочных Областных Целевых Программ.
4. Проектирование и реализация доработки подсистемы (с указанием конкретной подсистемы) на предприятии (название предприятия).
5. Автоматизация учета предоставления услуг (с указанием вида услуг) на предприятии (название предприятия).
6. Автоматизация расчетов с бытовыми потребителями электроэнергии на платформе «1С:Предприятие 8.2».
7. Автоматизация расчета (вид расчета) в модуле (название модуля) системы R/3.

8. Информационная система учета (разновидность учета) на предприятии (название предприятия).
9. Автоматизация документооборота (с указанием конкретной подсистемы) на предприятии (название предприятия).
10. Автоматизация учета договоров (вид договоров) в среде (название среды программирования).
11. Автоматизация оценки инвестиционной привлекательности региона на примере (название региона).
12. Информационная система оценки риска экономической несостоятельности производственного предприятия.
13. Автоматизация мониторинга и прогнозирования параметров субфедерального долга (название региона).
14. Автоматизация работы с клиентами в CRM-системе (название системы и предприятия).
15. Автоматизация деятельности страхового агента (на примере конкретного предприятия).

6 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО СТРУКТУРЕ И ОФОРМЛЕНИЮ ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ

6.1 Структура отчета

Структура отчета по учебной практике включает набор следующих *обязательных* элементов:

- титульный лист;
- задание;
- оглавление;
- введение;
- основная часть документа;
- заключение;
- список использованных источников.

Факультативными (необязательными) элементами отчета являются приложение, перечни условных обозначений, сокращений и терминов.

Титульный лист служит обложкой документа. Пример оформления титульного листа приведен в Приложении А.

Задание на учебную практику, как правило, оформляется в стандартном, принятом на кафедре АСУ виде, его пример приведен в приложении Б.

Оглавление включает: введение, наименования всех глав разделов, подразделов, пунктов (если они имеют наименования), заключение, список использованных источников, приложения (при наличии). Строки оглавления заканчиваются указанием *номеров страниц*, на которых расположено *начало* соответствующей части документа.

Заголовок «Оглавление» (с прописной буквы) размещают в центре строки (симметрично тексту). Наименования, включенные в содержание, записывают строчными буквами, начиная с прописной буквы. Содержание включается в общее количество страниц документа.

Введение должно содержать общие сведения о проведенном исследовании во время учебной практики. В нем необходимо отразить

актуальность выбранной темы, цель и задачи, решаемые во время практики, субъект (конкретное предприятие), объект (экономико-информационные процессы, происходящие на данном предприятии), используемые методики, практическую значимость полученных результатов.

Также необходимо перечислить вопросы, которые будут рассмотрены в отчете, выделив вопросы, которые предполагается решить практически на основе теоретических исследований. Объем введения должен быть не более 2—3 страниц и не менее одной полной страницы.

В заключении необходимо сделать краткие выводы и оценку полученных во время учебной практики результатов, определить направления дальнейшей работы, четко сформулировать предполагаемую тему выпускной квалификационной работы.

Список использованных источников содержит библиографическое описание всех литературных источников, использованных в отчете по учебной практике. Сведения о каждом из источников располагают в порядке их упоминания в тексте или в алфавитном порядке. Образцы библиографического описания наиболее важных типов литературных источников (с учетом требований нормативных документов) приведены в приложении Ж. На каждый источник из списка в тексте отчета необходимо поставить ссылку, указывая после взятого из источника фрагмента номер источника из списка в квадратных скобках. Также, как это сделано в данном пособии в первом абзаце раздела 6.2.

В приложении могут быть примеры первичных экономических документов (входные и выходные формы и бланки или реальные заполненные документы); информация об объекте исследования (сложная структура предприятия, положения о видах деятельности, лицензии, фрагменты громоздких таблиц номенклатур продукции, различные справочники предприятий и т. д.), иллюстрации и таблицы, если они являются громоздкими и не вписываются в формат пояснительной записки.

6.2 Общие положения по оформлению отчета

Отчет по учебной практике необходимо оформить в соответствии с образовательным стандартом университета ОС ТУСУР-2013, ознакомиться с которым можно на официальном сайте университета (<http://tusur.ru>) в разделе *Студентам* → *Обучение* → *Правила оформления курсовых и выпускных работ* [6].

В этом разделе приведены основные требования по оформлению отчета.

При оформлении отчета по учебной практике на персональном компьютере предъявляются следующие общие требования:

1) Общий **объем** машинописного текста, без приложений, должен составлять 25—30 страниц.

2) Текст отчета должен быть напечатан шрифтом Times New Roman 12—14 размера с **интервалом 1,5** на одной стороне стандартного листа белой односортной бумаги формата А4 размером 210×297 мм.

3) Текст следует выполнять, устанавливая следующие размеры полей: левое — 30 мм, правое — 10 мм, верхнее — 20 мм, нижнее — 20 мм. Размер абзацного отступа должен быть одинаковым по всему тексту работы, рекомендуемый размер абзацного отступа 12,5 мм, выравнивание текста — по ширине.

4) Контуры букв и знаков должны быть без ореола и расплывающейся краски. Насыщенность букв должна быть ровной в пределах строки, страницы и всей работы.

5) Таблицы, рисунки, схемы, графики, фотографии и др. в тексте пояснительной записки, и в приложении должны быть выполнены на стандартных листах формата А4. Таблицы и иллюстрации большого размера допускается выполнять на других стандартных форматах, при этом они должны быть сложены на формат А4 «гармоникой» по ГОСТ 2.501.

6) Все страницы, включая иллюстрации и приложения, нумеруются по порядку. Первой страницей считается титульный лист, на нем цифра «1» не

ставится (также на листе «Задание на учебную практику» и странице «Оглавление» номера страниц могут отсутствовать), но учитывается при общей нумерации. Порядковый номер печатается **по центру верхнего поля** страницы, начиная со следующей страницы после оглавления.

6.3 Деление текста отчета

Текст работы разделяют на разделы и подразделы. Внутри подразделов выделяют пункты, которые при необходимости могут быть разделены на подпункты. Каждый раздел рекомендуется начинать с нового листа (страницы).

Разделы (за исключением структурных элементов работы «Оглавление», «Сокращения, обозначения, термины и определения» и «Список использованных источников») должны иметь порядковые номера, обозначенные арабскими цифрами и записанные перед соответствующим заголовком.

Подразделы и пункты должны иметь нумерацию в пределах каждого раздела или подраздела; подпункты — в пределах пункта. Отдельные разделы могут не иметь подразделов и состоять непосредственно из пунктов. Если раздел или подраздел состоит из одного пункта, то нумеровать его не следует. Точка в конце номеров разделов, подразделов, пунктов, подпунктов не ставится.

Разделы и подразделы (главы и параграфы) должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны четко и кратко отражать содержание разделов и подразделов (глав и параграфов). Оформление заголовков должно соответствовать единому стилю форматирования, принятому в работе. Допускается выделение заголовков размером и (или) жирностью шрифта.

Заголовки следует выполнять посередине страницы с прописной буквы без точки в конце, не подчеркивая. В нумеруемых разделах перед заголовком помещают номер соответствующего раздела или подраздела. Переносы слов в

заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Расстояние по вертикали от текста до заголовка и между заголовком и текстом, а также между заголовками раздела и подраздела должно быть равно удвоенному межстрочному расстоянию (вставка «пустой» строки с одинарным интервалом).

Пример оформления фрагмента работы приведен в приложении Г.

6.4 Таблицы, иллюстрации, формулы

Таблицы применяют для лучшей наглядности и удобства сравнения показателей.

Таблицы слева, справа и снизу, как правило, ограничивают линиями. Головка таблицы должна быть отделена линией от остальной части таблицы. Разделять заголовки и подзаголовки боковика и граф диагональными линиями не допускается. Рекомендуемая высота строк таблицы — не менее 8 мм.

Графу «№ п/п (номер по порядку)» в таблицу включать не допускается. При необходимости нумерации параметров порядковые номера указывают в боковике таблицы перед их наименованием.

Нумерация граф таблицы арабскими цифрами допускается в тех случаях, когда в тексте работы имеются ссылки на них, при делении таблицы на части, а также при переносе части таблицы на следующий лист (страницу).

Все таблицы нумеруют в пределах раздела арабскими цифрами.

Над левым верхним углом таблицы помещают надпись «Таблица» с указанием номера таблицы, например: «Таблица 2.1» (первая таблица второго раздела), «Таблица В.5» (пятая таблица приложения В).

Если строки или графы таблицы выходят за формат страницы, ее делят на части, помещая одну часть под другой или рядом, при этом в каждой части таблицы повторяют ее головку и боковик.

Слово «Таблица» указывают один раз слева над первой частью таблицы, над другими частями пишут слова «Продолжение таблицы» с указанием номера таблицы.

Если в конце страницы таблица прерывается и ее продолжение будет на следующей странице, в первой части таблицы нижнюю горизонтальную линию, ограничивающую таблицу, не проводят.

Таблица может иметь название. Название таблицы должно отражать содержание, быть точным, кратким. Если таблица имеет название, то его помещают после номера таблицы через тире, с прописной буквы.

На все таблицы должны быть ссылки в тексте работы.

Таблицу следует располагать в работе непосредственно после абзаца, где она упоминается впервые, или на следующем листе (странице), а при необходимости — в приложении к работе.

Пример оформления таблицы и ссылки на неё приведен в приложении И.

Иллюстрации (фотографии, схемы, чертежи, графики, фрагменты листинга программы и пр.) помещаются в работе для наглядного пояснения текста.

В тексте работы все иллюстрации именуется рисунками.

Рисунки нумеруются в пределах раздела (приложения) арабскими цифрами, например: «Рисунок 1.2» (второй рисунок первого раздела); «Рисунок В.3» (третий рисунок приложения В).

Рисунок может иметь тематическое наименование и пояснительные данные (подрисуночный текст). Слово «рисунок», его номер и тематическое наименование (при наличии) помещают ниже изображения и пояснительных данных симметрично иллюстрации.

На все иллюстрации должны быть ссылки в тексте работы. Иллюстрации должны размещаться сразу после ссылки или на следующем листе (странице).

Ссылки в тексте на таблицы и иллюстрации оформляют по типу: (таблица 4.3); «... в таблице 1.1, графа 4»; (рисунок 2.11); «...в соответствии с рисунком 1.2»; «... как показано на рисунке Г.7, поз. 12 и 13».

Примеры оформления иллюстраций даны в приложении И.

Формулы следует выделять из текста в отдельную строку.

Значения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Значение каждого символа дают с новой строки в той последовательности, в какой они приведены в формуле. Первая строка расшифровки должна начинаться со слова «где» без двоеточия после него.

Формулы, следующие одна за другой и не разделенные текстом, отделяют запятой.

Формулы, на которые имеются ссылки в тексте работы, должны быть пронумерованы в пределах раздела (приложения) арабскими цифрами. Номер формулы должен состоять из номера раздела и порядкового номера формулы, разделенных точкой, например: «(1.4)». Номер указывают с правой стороны листа на уровне формулы в круглых скобках.

Пример

Плотность в килограммах на кубический метр вычисляют по формуле:

$$\rho = m/V, \quad (5.1)$$

где m — масса образца, кг;

V — объем образца, м³.

... ..

... ..

$$R = U/I, \quad (5.2)$$

$$P = UI. \quad (5.3)$$

СПИСОК ЛИТЕРАТУРЫ

1. Золотов, С. Ю. Проектирование информационных систем: Учебное пособие [Электрон. ресурс] / Золотов С. Ю. — Томск: ТУСУР, 2016. — 117 с. — Режим доступа: <https://edu.tusur.ru/publications/6478> (дата обращения 23.10.17).
2. Золотов, С. Ю. Проектирование информационных систем: Методические рекомендации для выполнения курсового проекта, лабораторных работ и практических занятий для студентов направления бакалавриата 09.03.03 «Прикладная информатика» [Электрон. ресурс] / Золотов С. Ю. — Томск: ТУСУР, 2016. — 34 с. — Режим доступа: <https://edu.tusur.ru/publications/6479> (дата обращения 23.10.17).
3. Сенченко, П. В. Организация баз данных: Учебное пособие [Электрон. ресурс] / Сенченко П. В. — Томск: ТУСУР, 2015. — 170 с. — Режим доступа: <https://edu.tusur.ru/publications/5179> (дата обращения 23.10.17).
4. Сенченко, П. В. Организация баз данных: Учебно-методическое пособие к выполнению самостоятельной и лабораторных работ и курсового проекта [Электронный ресурс] / Сенченко П. В. — Томск: ТУСУР, 2017. — 80 с. — Режим доступа: <https://edu.tusur.ru/publications/6671> (дата обращения 23.10.17).
5. Романенко, В. В. Объектно-ориентированное программирование: Учебное пособие [Электронный ресурс] / Романенко В. В. — Томск: ТУСУР, 2014. — 475 с. — Режим доступа: <https://edu.tusur.ru/publications/4872> (дата обращения 23.10.17).
6. Маклаков С. В. BPWin и ERWin. CASE-средства разработки информационных систем / С. В. Маклаков. — М. : Диалог-МИФИ, 2000. — 256 с.
7. Вендров, Александр Михайлович. Проектирование программного обеспечения экономических информационных систем : учебник для вузов / А. М. Вендров. - 2-е изд., перераб. и доп. - М. : Финансы и статистика, 2006. - 543[1] с.

8. Образовательный стандарт вуза ОС ТУСУР 01-2013. Работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления. Приказ ректора от 03.12.2013 г. №14103 [электр. ресурс]. — <https://regulations.tusur.ru/documents/70> (дата обращения 23.10.17).

ПРИЛОЖЕНИЕ А
Пример оформления
титульного листа отчета по учебной практике

Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Кафедра автоматизированных систем управления (АСУ)

РАЗРАБОТКА ПРОТОТИПА
ИНФОРМАЦИОННОЙ СИСТЕМЫ
МАРКЕТИНГОВЫХ ИССЛЕДОВАНИЙ
НА ОАО «МАНТОМЬ»

Отчет по учебной практике

Студент гр. 3-94э

_____ И. О. Фамилия

«__» _____ 20__ г.

Руководитель от университета

(должность, ученая степень, звание)

_____ И. О. Фамилия

«__» _____ 20__ г.

(оценка)

Томск 2017

ПРИЛОЖЕНИЕ Б

Пример формы задания

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Кафедра автоматизированных систем управления (АСУ)

Задание

на учебную практику
«Практика по получению первичных профессиональных умений и навыков,
в том числе первичных умений
и навыков научно-исследовательской деятельности»

Студенту факультета дистанционного образования группы 3447-а Любиной
Ирине Павловне

Тема работы: Автоматизация экономического анализа и мониторинга
предприятий банком «РРР-банк»

Индивидуальное задание: Разработать прототип информационной системы
для экономического анализа и мониторинга предприятий банком «Б-банк»

Время прохождения практики: с 1.09.2017 по 14.09.2017

Руководитель практики от университета,
доцент каф. АСУ, кандидат технических наук
М.В.Григорьева

Томск 2017

ПРИЛОЖЕНИЕ В

Пример введения отчета по учебной практике

ВВЕДЕНИЕ

Целью учебной практики является разработка прототипа информационной системы учета выездных съемок для ТК "53 канал".

Объектом исследования является телекомпания "53 канал" города Нск.

Предмет исследования – автоматизация учета и планирования выездных съемок телеканала.

Для достижения поставленной цели во время учебной решены следующие *задачи*:

- анализ предметной области;
- обзор состояния информационных технологий объекта исследования;
- формализованная постановка задачи;
- обзорный анализ готовых решений по автоматизации предметной области;
- разработка информационной и функциональной модели задачи;
- разработка прототипа пользовательского интерфейса и базы данных информационной системы.

Актуальность темы. Современное телевидение представляет собой высокотехнологическое производство, складывающееся из множества отдельных бизнес-процессов. Как и при промышленном производстве, в телекомпаниях существует разделение участков на специализации с целью получения конечного продукта производства (телевизионной передачи).

Переход на цифровое телевидение обуславливает огромные объемы различной медиа-информации, которые с каждым годом растут. Эффективная работа с такими объемами практически невозможна без автоматизации и применения компьютерной техники.

Одним из важных участков выступает учет выездных съемок: от приема заявок операторов и водителей (с соответствующими предпочтениями), формирования расписания, и до предоставления ведомости отработанного

времени в бухгалтерию телекомпании. Выполнение этих подпроцессов связано с необходимостью полной и оперативной информации обо всех участниках и объектах съемок. Такая информация может быть получена при автоматизированном ведении учета выездных съемок, что, в свою очередь, благоприятно сказывается на продуктивности всего телевизионного процесса.

Отсутствие автоматизации не позволяет в полном объеме проводить своевременный учет и контроль работы съемочных групп, ограниченные аналитические возможности бумажного (ручного) учета лишает руководство телеканала возможности аргументировано формировать съемочные команды.

Методы, используемые в работе: структурное проектирование в нотациях IDEF0, IDEF1x, визуальное объектное программирование, прототипирование.

ПРИЛОЖЕНИЕ Г

Пример оформления фрагмента работы

2 УПРАВЛЕНИЕ ЗАПАСАМИ НА ПРЕДПРИЯТИИ

2.1 Организационно-экономическая характеристика ЗАО «Градиент»

ЗАО «Градиент» является юридическим лицом и осуществляет свою деятельность на основании законодательства, действующего на территории РФ и Устава. Срок деятельности ЗАО «Градиент» не ограничен.

.....

2.1.1 Цели и задачи предприятия

.....

2.1.2 Масштаб деятельности предприятия и его место на рынке

ЗАО «Градиент» работает уже более 10 лет на рынке г. Омска и области и зарекомендовало себя как надежный партнер. Постоянными клиентами являются Омскэнергоремонт, ХимПром, Пенсионный фонд, Центральный банк РФ, Горэлектротранс, ОмскГипродорНИИ, ОМтелеком и многие другие.

Товары и услуги, предоставляемые фирмой:

– компьютерная и офисная техника российских и зарубежных производителей;

– копировально-множительная техника;

– канцелярские товары (около 1500 наименований);

– услуги гарантийного и послегарантийного обслуживания.

....

ПРИЛОЖЕНИЕ Д

Пример заключения отчета по учебной практике

ЗАКЛЮЧЕНИЕ

В период учебной практики разработан прототип информационной система учета выездных съемок для ТК "51 канал" г. Нск.

В отчете по практике проведен анализ предметной области и используемые информационные технологии. В ходе формализованной постановки задачи описаны основные бизнес-процессы телеканала посредством IDEF0 диаграмм с декомпозицией, обозначены основные функции информационной системы, её входные и выходные документы и отчеты.

На основании обзора готовых решений по автоматизации телепроизводства обоснована необходимость разработки собственной информационной системы учета выездных съемок.

В практической части обоснован выбор среды разработки и базы данных (C++ Builder и Microsoft Access). Построена функциональная модель информационной системы учета выездных съемок, структурно состоящая из трех модулей. Посредством DFD-диаграмм описаны потоки данных в системе и хранилища, а также внешние сущности.

Физическая база данных состоит из десяти таблиц, объединенных пятнадцатью связями "один-ко-многим".

В ходе тестирования разработанного прототипа информационной системы в базу данных внесена тестовая информация (справочники, планы, заявки), на основе которой сформировано расписание выездных съемок и построены статистические отчеты.

В дальнейшем предполагается разработка полноценной информационной системы путём доработки данного прототипа.

ПРИЛОЖЕНИЕ Е

Пример описания бизнес-процесса предметной области автоматизации и формализованной постановки задачи

2.1 Описание бизнес-процессов автоматизации

Компьютерная техника требует профессионального обслуживания специалиста, знакомого с особенностями этой техники, поэтому большинство ее владельцев предпочитает обслуживать компьютерную технику в специализированных фирмах, занимающихся ремонтом и сервисным обслуживанием. Сервисный центр «IT-сервис» заключает подобные договоры на обслуживание компьютерной техники. Непосредственно этим занимается служба сервисного обслуживания, которая делится на участки по территориальному расположению клиентов. Первоначально диагностику техники производят специалисты этих участков. При невозможности отремонтировать технику на удаленном участке, инженеры привозят технику на участок ремонта технических средств.

Схема информационных потоков при работе с клиентами в службе сервисного обслуживания (СО) с указанием направления и описания данных представлена на рисунке 2.2.

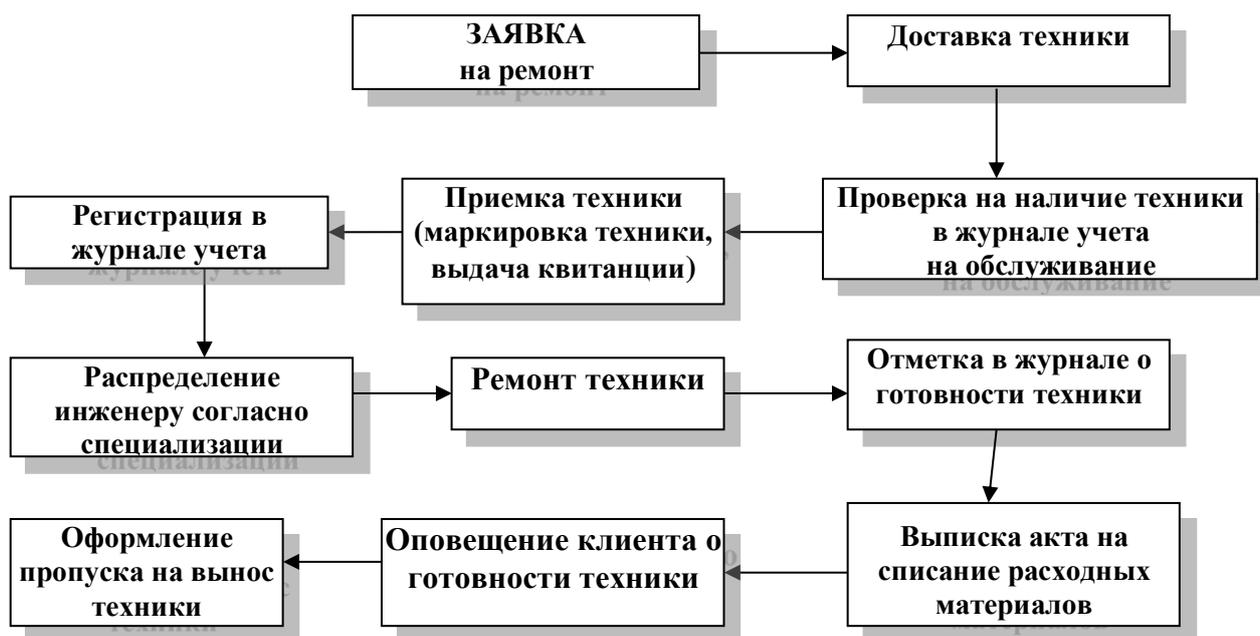


Рис. 2.2 – Порядок взаимодействия в службе сервисного обслуживания

Сервисный центр «IT-сервис» заключает договор на обслуживание определенного количества техники. Заказчик подписывает договор на обслуживание определенного количества техники с серийными номерами и соответствующими марками. Заказчик доставляет свою технику для ремонта, и, если этот аппарат есть в списках на обслуживание, его принимают в ремонт. Для этого делается соответствующая запись в журнал по учету техники на ремонт, куда записывается марка, серийный номер техники, фамилия представителя заказчика и наименование предприятия.

При наличии на складе расходных материалов, необходимых для ремонта, техника ремонтируется и выдается клиенту, а если их нет, то инженер делает заявку на эти материалы и техника ожидает прихода на склад необходимых комплектующих частей.

Также инженеры периодически выезжают для обслуживания техники, транспортировка которой невозможна или нецелесообразна, и производят ремонт, а также профилактические работы (чистка, смазка, настройка), заправку тонером – красящим порошком на месте у заказчика.

Ранее весь учет ремонта компьютерной техники в сервисном центре «IT-сервис» осуществлялся вручную с помощью журналов учета. С каждым годом количество обслуживаемой техники возрастало, и наконец, назрела необходимость автоматизации этого учета в службе сервисного обслуживания.

Из-за больших объемов ручной работы при ведении учета ремонта, расходных частей и комплектующих, возникают следующие проблемы:

- низкая скорость оперативного учета, сбора, хранения и обработки информации о клиентах и их компьютерной технике;
- высокая трудоемкость обработки информации;
- низкая оперативность, снижающая качество управления объектом;
- несовершенство организации сбора и регистрации исходной информации;

- низкая скорость оперативности при оказании информационно-справочных услуг клиентам, прием заявок от клиентов;
- некорректный учет результатов деятельности службы сервисного обслуживания;
- некорректный анализ сервиса работы службы сервисного обслуживания;
- замедление документооборота.

С помощью автоматизации управленческой функции возможно решение всех вышеперечисленных проблем. Автоматизированная система должна выполнять оперативный учет компьютерной техники и запчастей и оформление необходимых сопроводительных документов.

2.3 Формализованная постановка задачи

Требуется создать информационную систему учета заказов на ремонт компьютерной техники и расхода комплектующих для службы сервисного обслуживания. Необходимо рассмотреть следующие процессы:

- движения технических средств (от заказчика, в ремонт, из ремонта, заказчику);
- учета расхода запасных частей;
- ведение документации по движению заказа.

Задача учета движения заказа и запасных частей касается процессов поступления в ремонт от заказчика, выполнения работ по ремонту, выдачи техники из ремонта, расхода и списания запасных частей.

К **функциям**, которые должны быть реализованы в рассматриваемой задаче, относятся:

- учет поступления техники на ремонт в службу СО;
- формирование документов, связанных с поступлением техники в ремонт;
- учет выполнения работ по ремонту техники;

- формирование документов, связанных со списанием израсходованных материалов при ремонте;

- учет выдачи отремонтированной техники заказчику;

- формирование документов, связанных с выдачей техники из ремонта.

Из общего описания предметной области известен ряд особенностей, существенных для процессов, относящихся к рассматриваемой задаче.

Каждому объекту техники присваивается свой номер учета в журнале. Отсутствие маркировки на оборудовании свидетельствует о том, что данный объект не принят к учету. Система также должна хранить данные о выданной из ремонта технике.

Информационная система должна формировать 3 документа, назначение и состав которых рассмотрены ниже. Все эти документы содержат **входную информацию** для информационной системы.

1) Заявка на обслуживание – в этом документе содержится информация о том, кто подал заявку, какое устройство вышло из строя.

2) Журнал учета обслуживаемой техники – в этом документе содержится информация об объектах ремонта, т.е. это список техники по предприятиям, с которыми сервисный центр заключил договора на обслуживание этой техники, документ включает в себя следующую информацию:

- марка устройства (SONY, NEC; CANON и т.д.);
- тип устройства (например, монитор, принтер и т.д.);
- серийный номер устройства.

3) Заявка на получение материалов со склада сервисного центра – в этом документе содержится информация о том, какие расходные материалы или комплектующие требуются для ремонта оборудования и количество этих материалов.

Выходной информацией задачи для пользователя являются следующие документы (1-3) и отчеты (4-6), в которых отражается:

1) Справка о приеме техники в ремонт – оформляется при поступлении техники на участок ремонта. Документ включает с себя следующую информацию:

- номер заказа;
- дату поступления заказа;
- наименование техники, модель;
- серийный номер;
- предприятие – заказчик;

2) Акт на списание израсходованных запчастей – данный отчет выводит перечень и количество запчастей, израсходованных на ремонт определенного устройства. Данный отчет содержит следующую информацию:

- номер заказа;
- тип запчасти (диод, микросхема, резистор и т.д.);
- марка запчасти;
- дату исполнения;
- количество затраченных запчастей;
- наименование предприятия (для которого выполнялся ремонт).

3) Акт о завершении проведения сервисного обслуживания – заполняется при выдаче устройств из ремонта. Содержит следующие данные:

- номер заказа;
- дату поступления заказа;
- дату выполнения заказа
- наименование техники, модель;
- серийный номер;
- тип израсходованных запчастей;
- марка израсходованных запчастей;
- количество израсходованных запчастей;
- дату выдачи заказа;

– предприятие – заказчик.

4) Расходные материалы и комплектующие, израсходованные за определенный период – формируют отчет, содержащий все запчасти, сгруппированные по типу и марке, и их количество, израсходованные за заданный пользователем период времени. Содержит следующие данные:

- период (даты с... по...);
- тип запчасти;
- марка запчасти;
- количество.

5) Техника, находившаяся в ремонте за определенный период – выводит информацию об устройствах, находившихся в ремонте за заданный пользователем период времени:

- период (даты с... по...);
- дата поступления;
- тип оборудования;
- модель оборудования.

6) Техника, находившаяся в ремонте за определенный период по предприятиям – выводит перечень техники, находившуюся в ремонт за указанный пользователем период времени сгруппированную по определенному предприятию. Содержит следующую информацию:

- период (даты с... по....);
- устройство (тип);
- предприятие – заказчик;
- количество техники.

В шапке всех выходных документов должно присутствовать название отчета и период отчетности. Шрифт Arial 14, полужирный, черный.

В нижней части отчета необходимо указать ФИО ответственных за формирование документов и отчетов должностных лиц.

ПРИЛОЖЕНИЕ Ж

Примеры библиографического описания книг и других изданий

В этом приложении приводятся примеры описания источников информации в зависимости от вида источника. Описание источника может начинаться с фамилии автора (Описание под заглавием) или с названия источника (под заглавием). И в обоих случаях после названия источника необходимо указать автора.

1. Описание под заголовком (фамилией автора)

Давыдов, В. Г. Программирование и основы алгоритмизации: учеб. пособие / В. Г. Давыдов. — 2-е изд., стер. — М. : Высш. шк., 2005. — 447 с.

Шипова, В. М. Технология разработки программного обеспечения / В. М. Шипова, О. А. Дзукаев, О. А. Антонова. — М. : Грантъ, 2002. — 242 с.

2. Описание под заглавием

Технология разработки программного обеспечения / А. В. Степанов, В. Л. Цепелев, О. Д. Аюшиев. — Чита : Поиск, 2002. — 160 с.

Метрология, стандартизация и сертификация / под ред. А. Н. Кокосова. — СПб. : Лань, 2002. — 288 с.

3. Описание сборников трудов

Информационные технологии : сб. науч. тр. / под ред. А. А. Фокина, А. В. Важенина. — Челябинск, 2002. — 124 с.

4. Описание многотомного издания

Энциклопедия Си : в 2 т. / под ред. М. Р. Сапина. — М. : Высш. шк., 1986. — Т. 2.

5. Описание отдельного тома

Венгеровский, А.И. Поиск и сортировки / А. И. Венгеровский. — Томск : Изд-во ТГУ, 1996. — 260 с. — (Лекции по структурам и алгоритмам данным : В 2 частях. Ч. 1)

6. Описание автореферата диссертации

Коняева, Т. П. Автоматизация управленческого учета внебюджетной образовательной деятельности : автореф. дис. ... канд. мед. наук / Т. П. Коняева. — Кемерово, 2002. — 23 с.

7. Описание диссертации

Дзюиан, А. Н. Управление ресурсами в вузе: методология, технологии, практика : дис. ... д-ра техн. наук / А. Н. Дзюман. — Томск, 2002. — 172 с.

8. Описание сборника трудов

Доклады Томского государственного университета систем управления и радиоэлектроники : сб. науч. тр. / под ред. А. М. Корикова. — Томск, 2002. — 124 с.

9. Описание многотомного издания

Операционные системы : в 2 т. / под ред. М. Р. Сапина. — М. : Высшая школа, 1986. — Т. 2.

10. Описание отдельного тома

Венгеровский, А.И. Философия С++ / А. И. Венгеровский. — Томск : Изд-во ТГУ, 1996. — 260 с. — (Лекции по объектно-ориентированному программированию : В 2 частях. Ч. 1)

11. Описание статьи из журнала

Иванов, В. Т. Нелинейные производственные функции // Тер. архив. — 1999. — Т. 50, № 2. — С. 21—32.

12. Описание статьи из сборника

Нежувака, А. К. Итоги научно-исследовательской работы кафедры информатики / А. К. Нежувака, В. Н. Жданов // Актуальные вопросы подготовки инженерных кадров. — Томск : Томский политехнический ин-т, 2000. — С. 12—13.

13. Описание статей из энциклопедий:

Благообразов, В. А. Тянь-Шань / В. А. Благообразов, Н. А. Гвоздецкий, В. С. Буртман // БСЭ. — 3-е изд. — М., 1997. — Т. 26. — С. 428—431.

14. Описание электронных ресурсов.

Ресурсов локального доступа:

Джефферсон, Р. Сборник упражнений и задач для студентов по математической экономике [Электронный ресурс] / Р. Джефферсон. — Электрон. Дан. — 1999. — 1 электрон. опт. диск (CD-ROM).

Справочник C++ [Электронный ресурс] / ред. О. П. Фомина. — М., 2002. — 1 электрон. опт. диск (CD-ROM).

Ресурсов удаленного доступа:

Исследовано в России [Электронный ресурс] / Моск. физ.-тех. ин-т. — Электрон. журн. — Долгопрудный : МФТИ, — 1998. — Режим доступа к журн. : <http://zhurnal.mipt.rssi.ru>.

Faulkner, A. Разработка и стандартизация программного обеспечения ИС / Alison Faulkner, Phil Thomas // Обзор современных методологий [Электронный ресурс]. — Электрон. журн. — 2002. — Вып. 16. — Режим доступа к журн. : <http://www.psyobsor.org/>

15. Описание нормативных документов:

О несостоятельности (банкротстве) кредитных организаций : федер. закон РФ от 25 февр. 1999 г. № 40 // Рос. газ. — 4 марта.

О таможенной политике РФ : постановление Правительства РФ от 3 февр. 1997 г. № 37 // Рос. газ. — 1997. — 5 февр.

ПРИЛОЖЕНИЕ И

Примеры оформления таблиц и рисунков

Пример И.1

Таким образом, в ходе проделанной работы найдены все необходимые данные, доказывающие целесообразность и эффективность данной разработки. Приведем эти данные в сводной таблице 4.12.

Таблица 4.12 — Результаты экономического обоснования проекта

Характеристика проекта	Значение
Затраты на разработку и внедрение проекта, руб.	119 134,79
Общие эксплуатационные затраты, руб.	62 700,96
Экономический эффект, руб.	79 754,74
Коэффициент экономической эффективности	0,67
Срок окупаемости, лет	1,50

Пример И.2

На основе полученных данных был сформирован план-график выполнения работ по проекту для программиста и руководителя в виде диаграмм Ганта, которые приведены на рисунке 4.1 и рисунке 4.2 соответственно.

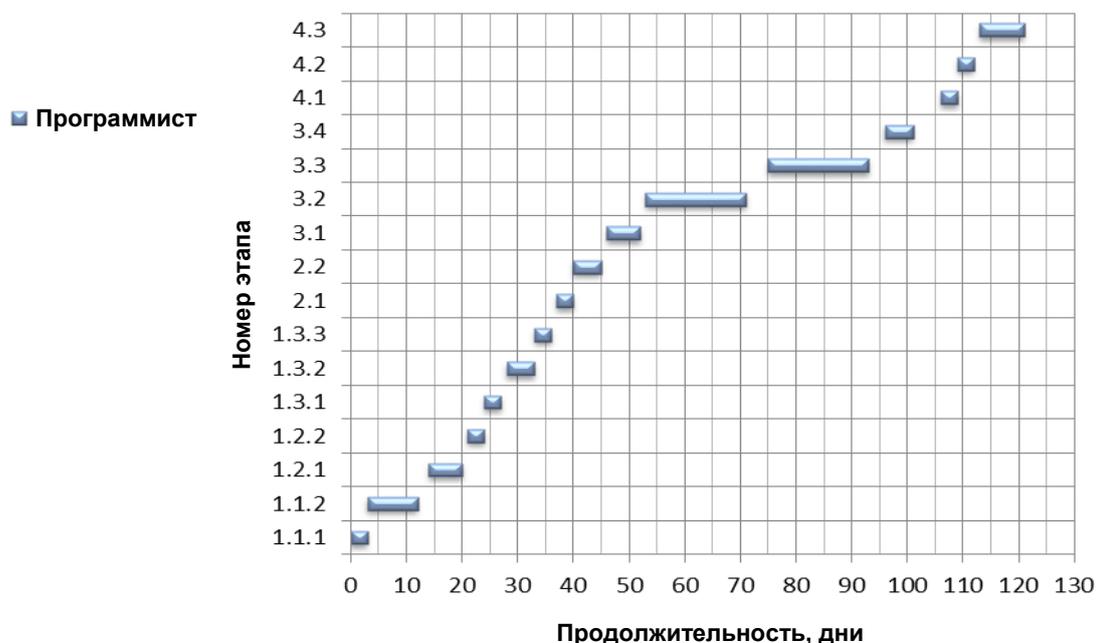


Рисунок 4.1 — План-график выполнения работ программиста

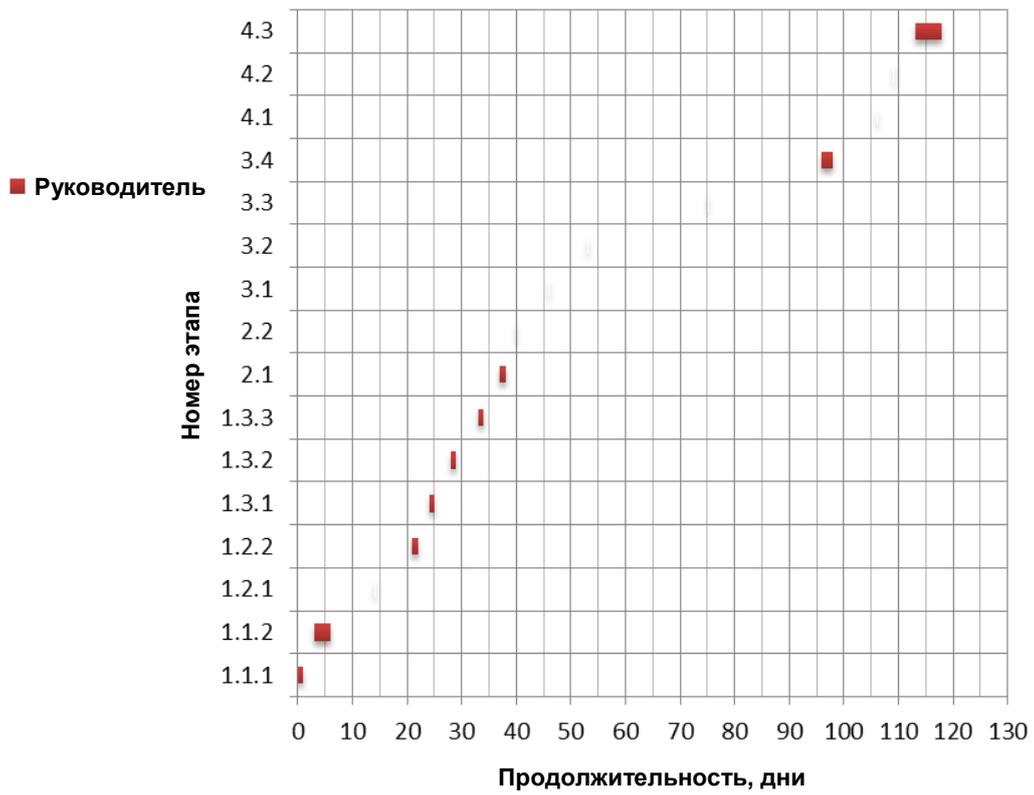


Рисунок 4.2 — План-график выполнения работ руководителя