

**АВТНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»**

**ФАКУЛЬТЕТ БУХГАЛТЕРСКОГО УЧЁТА И ПРИКЛАДНОЙ ИНФОРМАТИКИ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**В.М. Вдовин
В.Н. Мазай**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ
РАБОТЫ**

специальность 080801.65 Прикладная информатика (в экономике)

Москва 2010

УДК 004.9
ББК 32.973
В 25

Вдовин В.М., Мазай В.Н. Методические указания по выполнению выпускной квалификационной работы. – М.: Российский университет кооперации, 2010. – 135 с.

Методические указания по выполнению выпускной квалификационной работы для специальности 080801.65 Прикладная информатика (в экономике) разработаны в соответствии с требованиями Государственного образовательного стандарта высшего профессионального образования, утвержденного 14.03.2000 г., и учебным планом от 29 марта 2005 г.

Рецензент: Суркова Л.Е., к.т.н., доцент.

В пособии рассмотрены вопросы: организации работы по выполнению дипломного проекта; структуры, содержания и требований дипломного проекта. Даны рекомендации по подготовке, выполнению защите квалификационной работы.

Методические указания по выполнению выпускной квалификационной работы рекомендованы к изданию решением информационных технологий от 01 сентября 2009 г., протокол №1.

© АНО ВПО ЦС РФ
«Российский университет кооперации», 2010
© Вдовин В.М., Мазай В.Н., 2010

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ОСНОВНЫЕ ПОЛОЖЕНИЯ ДИПЛОМНОГО ПРОЕКТИРОВАНИЯ	5
1.1. Общие требования, предъявляемые к дипломному проектированию	5
1.2. Цель и задачи дипломного проектирования	6
1.3. Основные требования к дипломному проекту	8
1.4. Преддипломная практика	11
2. ОРГАНИЗАЦИЯ РАЗРАБОТКИ И ЗАЩИТЫ ДИПЛОМНОГО ПРОЕКТА	12
2.1. Закрепление темы дипломного проекта за студентом	12
2.2. Руководство дипломным проектированием	14
2.3. Порядок защиты дипломного проекта	15
3. ПРИМЕРНОЕ СОДЕРЖАНИЕ И СТРУКТУРА ДИПЛОМНОГО ПРОЕКТА	16
3.1. Общие положения по структуре дипломного проекта	16
3.2. Содержание разделов структуры проекта по проектированию и созданию АИС	16
3.3. Содержание разделов структуры проекта по проектированию и созданию программных средств обработки экономической информации	99
3.4. Содержание разделов структуры проекта по модификации, адаптации, внедрению и сопровождению тиражируемых вариантов АИС и программных средств обработки экономической информации	118
3.5. Содержание разделов структуры проекта по проектированию и разработке WEB-ресурсов экономической деятельности	121
4. ОФОРМЛЕНИЕ ДИПЛОМНОГО ПРОЕКТА	123
5. ПРИМЕРНАЯ ТЕМАТИКА ДИПЛОМНОГО ПРОЕКТИРОВАНИЯ	128
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	131
ПРИЛОЖЕНИЯ	132

ВВЕДЕНИЕ

Настоящие методические указания по выполнению выпускной квалификационной работы (дипломного проекта) для специальности 080801.65 Прикладная информатика (в экономике) написаны в соответствии с требованиями Государственного образовательного стандарта высшего профессионального образования к минимуму содержания и уровню подготовки специалистов.

Квалификационная характеристика информатика-экономиста по специальности 080801.65 Прикладная информатика (в экономике) предусматривает подготовку студентов к проектно-конструкторской, производственно-управленческой, экспериментально-исследовательской и эксплуатационной деятельности, направленных на создание и применение систем сбора, хранения, передачи и обработки информации.

В результате подготовки и выполнения дипломного проекта студенты должны показать:

знания

- методов системного анализа для описания предметной области, объекта проектирования и постановки задачи, решаемой в проекте;
- формального аппарата для анализа организационной, функциональных, информационных, алгоритмических и технических структур информационных систем;
- методов системного моделирования при исследовании и проектировании информационных систем;
- математические модели для анализа, расчетов, оптимизации детерминированных и случайных явлений и процессов в информационных системах;
- методов теории информационных процессов и систем и средства информационных технологий при разработке и проектировании информационных систем;
- методов анализа и синтеза информационных систем и их компонентов;
- возможностей вычислительных систем и сетей при построении информационных систем;
- методов и средств разработки алгоритмов и программ, технологий программирования информационных систем;
- системные программные средства, операционные системы и оболочки, обслуживающие сервисные программы;
- модели представления знаний и формализации задач при разработке интеллектуальных компонентов информационных систем;

- инструментальные средства компьютерной графики и мультимедиа технологии;

умение

- формулировать основные технико - экономические требования к информационным системам;
- формулировать и решать задачи проектирования информационных систем и их компонентов с использованием информационных технологий;
- количественно и качественно оценивать эффективность функционирования информационных систем;
- формулировать задачи обеспечения информационной безопасности и защиты информации в проектируемых информационных системах;
- разрабатывать общий процесс проектирования информационных систем;
- разрабатывать структуры информационно-логических, функциональных и алгоритмических моделей информационных систем и их компонентов;
- разрабатывать принципы обеспечения безопасной жизнедеятельности при создании и эксплуатации информационных систем.

1. ОСНОВНЫЕ ПОЛОЖЕНИЯ ДИПЛОМНОГО ПРОЕКТИРОВАНИЯ

1.1. Общие требования, предъявляемые к дипломному проектированию

В соответствии с Законом РФ «Об образовании», а также с Государственными образовательными стандартами итоговая государственная аттестация выпускников, завершающих обучение по программам высшего профессионального образования в вузах является обязательной.

Согласно Положению об итоговой государственной аттестации выпускников высших учебных заведений в РФ итоговая государственная аттестация выпускников включает один или несколько аттестационных испытаний. Итоговая государственная аттестация состоит из:

- итогового междисциплинарного экзамена по специальности;
- защиты выпускной квалификационной работы (дипломного проекта).

Выпускные квалификационные работы выполняются в форме дипломного проекта, который является заключительным этапом обучения студентов в вузе.

До защиты работы студенты обязаны сдать междисциплинарный экзамен по специальности, который согласно Положению должен, наряду с требованиями и содержанием обязательных дисциплин, учитывать также общие требования к выпускнику, предусмотренные Государственным образовательным стандартом по данной специальности.

Выпускники, не прошедшие междисциплинарный экзамен, допускаются, согласно Положению, к защите дипломных проектов повторно. Студентам, не прошедшим аттестационных испытаний по уважительной причине, увеличивается срок обучения до следующего периода работы Государственной аттестационной комиссии (ГАК), но не более одного года.

1.2. Цель и задачи дипломного проектирования

Дипломное проектирование является заключительным этапом обучения студентов в университете, а дипломный проект – квалификационной работой, предъявляемой Государственной аттестационной комиссии (ГАК) при решении вопроса о присвоении студенту квалификации информатика-экономиста соответствующей специальности.

Целью дипломного проектирования является:

- систематизация, закрепление и расширение теоретических и практических знаний по проектированию, внедрению, модернизации, сопровождению автоматизированных информационных систем (АИС) с использованием современных информационных технологий на основе анализа экономико-информационной среды предметной области;
- развитие навыков ведения самостоятельной работы и разработки проектных решений по информационному, технологическому и программному обеспечению экономических задач;
- систематизация, закрепление и расширение теоретических и практических навыков студентов по избранной специальности;
- развитие навыков ведения самостоятельной работы и овладения методикой исследования при решении определенных проблем и вопросов дипломного проекта;
- установление уровня теоретических и практических знаний студентов, а также их умения применять полученные теоретические знания для решения конкретных практических задач.

В процессе выполнения дипломного проекта у студентов формируются следующие навыки:

- выстраивать логическую структуру проекта;
- анализировать экономико-информационную среду предметной об-

ласти и устанавливать структурное представление и взаимосвязи с другими компонентами информационного пространства;

- анализировать объект управления системой;
- классифицировать существующие АИС и определять направления создания АИС;
- анализировать информационные потоки, систематизировать документооборот, определять уровень автоматизации задач и состав автоматизированных и неавтоматизированных работ;
- анализировать особенности автоматизации процессов сбора, регистрации и передачи первичной информации;
- использовать экономико-математические модели и алгоритмы оптимизации процесса управления предметной областью;
- использовать экономико-математические модели и алгоритмы распределения вычислительных работ и информационных массивов по узлам локальной сети, на основе методик организации распределенной многоуровневой обработки информации;
- анализировать требования к концептуальному моделированию и выбирать инструментарий класса CASE;
- применять технологии концептуального моделирования экономико-информационной среды предметной области;
- разрабатывать состав и структуру функциональной части АИС с использованием современных методологий;
- производить информационное моделирование АИС на основе существующих методологий;
- разрабатывать организационную структуру предметной области;
- разрабатывать структуру АИС и электронную технологию функционирования подразделений предметной области;
- анализировать существующий рынок аппаратного и программного обеспечения;
- проектировать базовую топологию локальной вычислительной сети (ЛВС) предметной области с использованием современных технологий;
- проектировать технологию, обеспечивающую своевременный сбор, регистрацию, передачу, обработку, модификацию, хранение, анализ, защиту и выдачу необходимой информации пользователям;
- производить организацию баз данных, нормативно-справочной и оперативной информации АИС;
- анализировать особенности алгоритмизации предметной области и организации программного обеспечения системы;
- использовать современные языки программирования, СУБД при разработ-

ке АИС;

- использовать, современные обеспечивающие информационные технологии, такие как электронные таблицы, текстовые процессоры, графические редакторы и средства анимации, мультимедиа при подготовке дипломного проекта;
- разрабатывать и реализовывать проект в виде комплекса автоматизированных рабочих мест.

В соответствии с поставленной целью студент в процессе выполнения квалификационной работы должен решить перечисленные ниже задачи.

1. Изучить научную, справочную литературу и другие источники информации по рассматриваемой проблеме.

2. Изучить материально-технические и социально-экономические условия производства, характер их влияния на изменение технико-экономических показателей работы конкретного предприятия.

3. Дать подробный анализ собранной информации о работе предприятия или отдельных подразделений, увязывая его с выбранной темой проекта.

4. Разработать мероприятия, направленные на повышение эффективности работы предприятия с учетом выбранной темы.

5. Выполнить прогнозный расчет экономической эффективности предлагаемых мероприятий.

При выполнении дипломного проекта студент должен широко использовать специальные литературные источники, официальные документы правительства страны, нормативно-справочные материалы, результаты социологических исследований, фактический материал предприятия, материалы статистических органов, передовой отечественный и зарубежный опыт, современные экономико-математические методы исследований и обработки исходной информации.

1.3. Основные требования к дипломному проекту

Требования к дипломному проекту студента-дипломника специальности 080801.65 Прикладная информатика (в экономике) целиком основаны на Государственном стандарте этой специальности.

Дипломный проект – завершающий этап учебного процесса по подготовке специалиста, поэтому в работе должны быть сконцентрированы знания, полученные по теоретическим и профилирующим дисциплинам, дающие возможность оценить подготовку студентов к самостоятельной работе.

Несмотря на то, что дипломный проект является проектом учебным, к нему предъявляются следующие требования:

- актуальность и практическая значимость;
- широкое использование трудов известных ученых по рассматриваемой теме, существующих законодательных актов, решений директивных органов, данных специальной литературы, передовой отечественной и зарубежной практики;
- раскрытие теоретических аспектов темы, ее значение для повышения рыночной устойчивости предприятий;
- использование современных положений, инструкций, стандартов, указаний, методик и инструментов технико-экономического анализа и прогнозирования;
- применение критического подхода к действующей практике учета результатов производственно-финансовой деятельности предприятий, контроля и анализа;
- конкретное раскрытие темы с использованием фактической открытой информации реально работающих предприятий;
- освещение различных точек зрения по затронутым в дипломном проекте дискуссионным вопросам и обязательная формулировка аргументированной позиции студента по ним;
- наличие конкретных самостоятельных предложений по совершенствованию информационно-экономических процессов на предприятии;
- прогнозная оценка экономической эффективности и повышения деловой активности предприятия после использования разработанных студентом проектных предложений;
- применение при разработке дипломного проекта экономико-математических методов и компьютерной техники, типовых и собственных программных продуктов;
- четкое и грамотное изложение материала, оформление проекта в соответствии с существующими требованиями;
- дипломный проект должен представлять целостную, системную работу, все части которой органически взаимосвязаны.

При работе над дипломным проектом специалист, обучающийся по специальности «Прикладная информатика (в экономике)» должен:

показать осведомленность:

- в стандартах, распространяемых на процессы разработки и конечные продукты программного и аппаратного обеспечения информационных систем и организациях, разрабатывающих эти стандарты;
- в крупнейших производителях программного и аппаратного обеспечения в России и в мире, признаках классификации и критериях качества их продукции;

- в конъюнктуре рынка информационных технологий (ИТ), приоритетных и перспективных направлениях;
- в маркетинговой стратегии на рынке ИТ, источниках информации о новейших разработках;
- в вопросах управления проектами в области ИТ и менеджменте в целом;

проявить знания:

- предметных и функциональных технологий в области экономики по своей специализации в России и в мире, направления их развития с учетом зарубежных аналогов;
- современных средств и методов проектирования и разработки АИС на всех этапах реализации;
- экономического аспекта внедрения АИС на предприятии, способов сокращения затрат на их внедрение;
- принципов организации и работы компьютерных сетей, сетевого администрирования, защиты информации;
- принципов организации баз данных, баз знаний и экспертных систем;
- основ информационного бизнеса;

применить умения:

- проводить квалифицированное исследование предметной области, предлагать практические решения по реорганизации предметных технологий с целью улучшения экономических показателей деятельности;
- применять математические методы и модели для анализа объектов и процессов предметной области;
- проектировать однопользовательские и многопользовательские АИС; использовать средства автоматизированного проектирования АИС (CASE средства);
- уметь адаптировать готовые программные продукты и проектные решения к условиям конкретной предметной области;
- самостоятельно разрабатывать подсистемы АИС с использованием инструментальных средств разработки АИС;
- проводить тестирование, отладку, внедрение разработанной АИС, ее сопровождение, модернизацию или интеграцию с ней новых приобретенных программных продуктов;
- организовывать, планировать проект и руководить небольшой группой специалистов на всех этапах жизненного цикла АИС.

Как следует из требований, специалист должен не только уметь проектировать и разрабатывать информационные системы, используя новейшие методы и технологии, но и оценивать проектирование АИС с эконо-

мической точки зрения. Таким образом, согласно требованиям к дипломному проекту, студент-дипломник должен:

- сформулировать экономическую сущность задачи;
- провести формализацию расчета экономических показателей;
- обосновать проектные решения по автоматизации, поставленной задачи;
- выполнить структурно-функциональный анализ экономической системы, выделить объект автоматизации и выработать предложения по его совершенствованию;
- разработать информационную модель комплекса задач;
- разработать информационное обеспечение комплекса задач;
- разработать схему взаимосвязи модулей и файлов, диаграмму потоков данных, модель взаимодействия объектов;
- разработать схему технологического процесса решения задачи с использованием новой технологии;
- сформировать блок-схему алгоритма программного модуля.

Приведенные требования должны быть реализованы в расчетно-пояснительной записке и иллюстративных материалах к проекту (не менее 8), а в процессе защиты показаны, что является основанием для ГАК принятия положительного решения о присвоении студенту соответствующей квалификации специалиста. Однако единство общих требований не исключает широкой инициативы и проявления элементов творчества в разработке темы дипломного проекта. Оригинальность решения конкретных вопросов в соответствии с особенностями предприятия и другими факторами является одним из основных критериев оценки качества дипломного проекта.

1.4. Преддипломная практика

Учитывая то, что дипломные проекты, выполняются на материалах реально работающих предприятий, темы дипломных проектов должны быть предложены студентам выпускающей кафедрой на 4-м курсе и на этом же курсе (перед производственной практикой) каждому студенту необходимо определиться в части примерной темы.

Работа над дипломным проектом включает в себя ряд этапов:

- выбор и закрепление объектов преддипломной практики;
- выбор и закрепление темы дипломного проекта;
- разработку и утверждение задания на дипломный проект;
- сбор материала для проектирования на объекте практики;
- написание и оформление пояснительной записки и чертежей, входящих в дипломный проект;

- сдачу проекта на кафедру и подготовку выступления в ГАК;
- предварительную защиту дипломного проекта на кафедре;
- защиту в ГАК.

После прохождения производственной практики и сдачи отчета, студенты 5-го курса должны окончательно выбрать тему дипломного проекта и к середине 9-го семестра написать заявление на имя заведующего выпускающей кафедры с просьбой о закреплении темы и пожеланиями о руководителе дипломного проекта.

Перед выходом на преддипломную практику студент составляет ориентировочный план и сроки выполнения проекта. План и сроки должны быть согласованы с руководителем и отражены в задании кафедры на дипломное проектирование.

Перед убытием на преддипломную практику выпускающая кафедра проводит со студентами собрание, на котором ставятся общие задачи, доводится установленный на кафедре график выполнения и предоставления дипломных проектов на рецензирование и защиту, порядок самой защиты, а также оформляются задания кафедры на выполнение дипломного проекта, которые утверждаются заведующим кафедрой.

В период прохождения преддипломной практики студент собирает материал, необходимый для выполнения дипломного проекта, изучает литературные источники по выбранной теме, материально-технические и социально-экономические условия производства, делает краткий анализ собранной информации, намечает и согласовывает с администрацией предприятия мероприятия, направленные на улучшение финансово-экономического положения предприятия. Эти материалы включаются в отчет по практике, который сдается на кафедру для проверки, а затем защищается перед началом непосредственного дипломного проектирования.

2. ОРГАНИЗАЦИЯ РАЗРАБОТКИ И ЗАЩИТЫ ДИПЛОМНОГО ПРОЕКТА

2.1. Закрепление темы дипломного проекта за студентом

Темы и примерное содержание выпускных квалификационных работ (дипломных проектов) определяются кафедрой и могут обновляться. Студенту предоставляется право выбора темы вплоть до предложения своей темы с необходимым обоснованием целесообразности её разработки.

Тематика дипломных проектов должна быть актуальна, соответствовать современному состоянию и перспективам развития АИС на базе различных классов ЭВМ и разнообразных средств сбора, передачи и отобра-

жения информации. При определении тем дипломных проектов следует исходить из реальных потребностей организаций, предприятий, банков, фирм в разработке и из возможности внедрения фрагментов будущего проекта в производство.

Окончательное заключение о целесообразности и актуальности темы дипломного проекта осуществляется преподавателем выпускающей кафедры и руководителем преддипломной практики.

В соответствии с квалификационной характеристикой специальности 080801.65 Прикладная информатика (в экономике) возможны следующие основные направления тематики дипломных проектов:

1. Проектирование и разработка АИС, решающих вопросы обработки экономической информации по комплексу задач и функций управления процессами и ресурсами различных сфер деятельности предметной области.

2. Разработка систем информационной поддержки принятия решения для менеджеров различного уровня.

3. Разработка информационных систем управления различными экономическими объектами.

4. Проектирование специализированных баз и банков данных и знаний экономической информации.

5. Создание экспертных систем.

6. Модернизация и модификация существующих АИС.

7. Разработка лабораторных практикумов, деловых игр и программных комплексов на основе использования различных классов ЭВМ и систем связи.

Название темы дипломного проекта должно быть краткими, отражать доминанту (основное содержание) проекта. В названии темы нужно указать технологии и современные методы проектирования.

Целесообразно, чтобы дипломным проектом был охвачен комплекс задач (не менее 5) предметной области. Под задачей следует понимать алгоритм или совокупность алгоритмов обработки и формирования результатной информации.

Закрепление темы дипломного проекта осуществляется кафедрой, а затем утверждается приказом ректора на основе поданного студентом заявления. После этого студент совместно с руководителем разрабатывает задание на дипломный проект, которое включает план проекта, содержание графических работ в листах, перечень основных литературных источников и др.

При выборе темы дипломного проекта следует руководствоваться интересом к проблеме, ее актуальностью, возможностью получения конкретных статистических данных, наличием специальной учебной и науч-

ной литературы. Студенты-дипломники при выборе темы руководствуются примерным перечнем тем дипломных проектов, имеющихся на кафедре, предварительно проконсультировавшись с будущим руководителем. После выбора темы студенту необходимо написать заявление закрепление темы и получить у руководителя задание на дипломное проектирование.

Руководители предприятий, где студенты-дипломники проходили преддипломную практику вправе обратиться в университет с письменной просьбой по разработке в дипломном проекте вопросов, интересующих данное предприятие.

2.2. Руководство дипломным проектированием

Руководителями дипломного проектирования назначаются наиболее опытные и квалифицированные преподаватели выпускающей кафедры.

После выдачи задания руководитель дипломного проекта обязан:

- оказать студенту помощь в разработке календарного графика на весь период выполнения дипломного проекта; выборе методов и приемов его реализации;
- проводить консультации по проблемным вопросам подбору основной литературы, справочных материалов и других источников информации по теме проекта;
- осуществлять систематический контроль за ходом проектирования в соответствии с утвержденным календарным планом;
- проводить обязательную предзащиту дипломного проекта с целью выявления степени готовности к публичной защите студентом своей работы;
- после предварительного заслушивания дать оценку подготовленности проекта и студента, отразив ее в письменном отзыве руководителя;
- организовать получение рецензии на дипломный проект, его доработку по замечаниям рецензента и получение допуска к публичной защите в ГАК;
- в обязательном порядке присутствовать на публичной защите дипломника в ГАК.

В случае необходимости выпускающей кафедре предоставляется право приглашать консультантов по отдельным разделам дипломного проекта за счет лимита времени, отведенного на руководство дипломным проектом. Консультанты консультируют и проверяют соответствующие разделы, выполненные студентом проекта и ставят свою подпись.

В согласованные с руководителем сроки студент отчитывается перед ним о ходе выполнения дипломного проекта.

На законченный дипломный проект должен быть представлен отзыв руководителя. В отзыве дается характеристика по всем разделам выполненной работы. Заведующий кафедрой на основании этих материалов решает вопрос о допуске студента к защите, делая об этом соответствующую запись на титульном листе дипломного проекта.

2.3. Порядок защиты дипломного проекта

Дипломный проект допускается к защите по результатам предварительной защиты проводимой на выпускающей кафедре согласно разработанному графику.

Дипломный проект, допущенный выпускающей кафедрой к защите, направляется деканатом на рецензию. Состав рецензентов утверждается деканом факультета по представлению заведующего кафедрой из числа преподавателей университета и специалистов производства. Рецензия в установленном порядке направляется вместе с отзывом руководителя и дипломным проектом в ГАК для защиты.

Защита дипломных проектов проводится на открытом заседании ГАК. При этом студент должен иллюстрировать доклад, основные выводы и предложения соответствующими материалами (таблицы, схемы, графики и т.д.). Результаты защиты дипломного проекта оцениваются оценками: отлично, хорошо, удовлетворительно, неудовлетворительно. Итоги защиты дипломных проектов объявляются в то же день после оформления протоколов заседания ГАК.

Студенту, защитившему дипломный проект, присваивается решением ГАК квалификация информатик-экономист по специальности 080801.65 Прикладная информатика в экономике, вручается диплом и нагрудный знак. Студенту, сдавшему курсовые экзамены с оценкой «отлично» не менее, чем по 75 % всех дисциплин учебного плана, а по остальным дисциплинам – с оценкой «хорошо» и защитившему дипломный проект с оценкой «отлично», а также проявившему себя в научной и общественной работе выдается диплом с отличием.

Дипломные проекты после защиты хранятся в архиве.

В тех случаях, когда защита дипломного проекта признается неудовлетворительной, ГАК устанавливает, может ли студент представить к повторной защите тот же проект с доработкой, определяемой комиссией, или же обязан разработать новую тему, которая устанавливается кафедрой. Студент, не защитивший дипломный проект, допускается к повторной защите ее в течение года после окончания университета.

3. ПРИМЕРНОЕ СОДЕРЖАНИЕ И СТРУКТУРА ДИПЛОМНОГО ПРОЕКТА

3.1. Общие положения по структуре дипломного проекта

Дипломный проект выполняется в соответствии с заданием на дипломное проектирование, которое включает *план структуры проекта*, построенный по одному из предложенных ниже вариантов, в который с разрешения руководителя, могут быть внесены изменения.

В соответствии с направлениями дипломного проектирования изложенными выше, возможна следующие направления дипломного проектирования:

1. Проектирование и создание АИС.
2. Проектирование и создание программных средств обработки экономической информации.
3. Модификация, адаптация, внедрение и сопровождение тиражируемых вариантов АИС и программных средств обработки экономической информации.
4. Проектирование и разработка WEB-ресурсов экономической деятельности.

По каждому изложенному направлению структура проекта будет различной.

3.2. Содержание разделов структуры проекта по проектированию и созданию АИС

Введение должно содержать общие сведения о дипломном проекте. В нем необходимо отразить актуальность выбранной темы, цели и задачи, решаемые в проекте, субъект (конкретное предприятие), объект (экономико-информационные процессы, происходящие на данном предприятии), используемые методики, практическую значимость полученных результатов.

Целью проекта может быть совершенствование процессов управления объектами экономической деятельности с использованием информационных технологий на базе:

- автоматизации процессов сбора, хранения, обработки и представления данных об объекте управления (функциональные подсистемы: управления складом, ассортиментом, продажами; CRM; производственный учет, маркетинг; бухгалтерский учет, планирование и т.д.);
- автоматизации процессов обработки данных об объекте управления с использованием экономико-математических, оптимизационных моделей и т.д.;

- совершенствования, адаптации процессов сбора, хранения, обработки и представления данных об объекте управления с использованием типовых экономических и информационных решений;
- расширения области экономической деятельности с использованием глобальной сети Интернет и интернет- технологий.

Во введении необходимо также перечислить задачи, которые будут рассмотрены в проекте, выделив вопросы, которые предполагается решить практически для достижения поставленной цели.

При этом нужно продумать новизну разработки и изложить перспективы развития объекта управления и проектируемой АИС.

Объем введения должен быть не более 3 страниц (здесь и далее указаны объемы машинописного текста).

Глава 1. Характеристика объекта внедрения АИС

1.1. Краткая характеристика предприятия должна включать краткую характеристику технико-экономических аспектов объекта управления.

Таковыми аспектами являются организационная структура предприятия, объект управления, тип производства, номенклатура готовой продукции, материалов и т.п.

Успешное выполнение этой части работы возможно только при соблюдении определённых требований, а именно: анализ должен учитывать особенности разнообразия форм собственности и применяемой модели хозяйствования, выявить резервы повышения эффективности производства, носить технико-экономический характер с учётом отраслевых особенностей, основываться на данных не только бухгалтерской отчётности, но и на плановой и нормативной информации, на данных статистической и специализированной ведомственной и внутрихозяйственной отчётности, другой информации.

В качестве основных производственных результатов необходимо оценить динамику объёмов производства, уровня себестоимости и реализации продукции, работ, услуг по номенклатуре и ассортименту. При этом следует обратить внимание на ритмичность производства и реализацию продукции, так как последняя в значительной степени влияет на формирование производственных результатов и задействованных ресурсов.

1.2. Анализ хозяйственной и финансовой деятельности объекта исследования

В качестве объекта исследования может выступать подразделение предприятия, фирмы, объединения и т.д., или отдельный вид деятельности, протекающий в нем, поэтому далее нужно привести краткую характеристику этого подразделения, в которой осуществляется рассматриваемая деятельность, и описать его структуру, перечень выполняемых в этом под-

разделении функций управления и его взаимодействие с другими подразделениями данного предприятия или подразделениями внешней среды.

Затем необходимо дать общее описание рассматриваемой деятельности, а также характеристику технико-экономических свойств ее как объекта управления.

Главными технико-экономическими свойствами объекта управления являются: цель и результаты деятельности, основные этапы и процессы рассматриваемой деятельности, используемые ресурсы и материалы. В ходе рассмотрения перечисленных свойств, для них, следует указать количественно-стоимостные оценки и ограничения.

Характеризуя подразделение предприятия, следует отразить особенности его функционирования, то есть принятые нормы и правила осуществления анализируемой деятельности, в условиях конкретной организации или предприятия.

В частности финансовый анализ рекомендуется проводить по следующим направлениям:

- Анализ обеспеченности предприятия оборотными средствами и эффективности их использования. Учитывая, что оборотный капитал является составной частью всего капитала предприятия, при анализе необходимо выяснить долю заёмных оборотных средств и их тенденцию изменения во времени. Для оценки эффективности использования оборотного капитала следует применить показатели: коэффициент оборачиваемости, длительность одного оборота и коэффициент закрепления оборотных средств в реализованной продукции. Наблюдаемые тенденции должны быть увязаны с ликвидностью оборотных средств и динамикой удельного веса наименее ликвидных.
- Экономический результат, производственные достижения предприятия, его имущественное и финансовое положение в значительной степени определяется уровнем использования производственных ресурсов, которыми оно располагает. Поэтому в дипломном проекте следует провести соответствующий анализ по видам используемых ресурсов, сделав акцент на обеспеченность материальными ресурсами и анализ производственных запасов в зависимости от специфики производственно-хозяйственной деятельности, анализ состояния и использования основных производственных фондов, трудовых ресурсов, в целом затрат на производство и реализацию продукции.
- Анализ обеспеченности материальными ресурсами и состояния производственных запасов. Обеспеченность материальными ресурсами и наличие запасов оценивается в сравнении с принятыми нормативами (планами), а также уровнем выполнения договорных обязательств поставщиками. При этом возможна проверка прогрессивности принятых

нормативов потребности в материальных ресурсах и установленного уровня производственных запасов. На предприятиях технического сервиса следует сосредоточить внимание на объёмах и номенклатуре обменного фонда, особенно на страховых запасах, являющихся компенсатором сезонного спроса на услуги, полнокомплектные машины, их узлы и агрегаты. В этом случае может быть оценена экономическая целесообразность изменения объёмов и номенклатуры обменного фонда.

- Анализ состояния и использования основных производственных фондов. Основной целью анализа производственных фондов является поиск резервов снижения их потребления в процессе производства продукции. Для этого необходимо рассмотреть динамику структуры и состояния фондов, при этом в проектах следует обратить особое внимание на активную часть фондов. Воспроизводство фондов может быть проанализировано с помощью следующих коэффициентов, характеризующих движение фондов: коэффициентов ввода фондов, обновления, выбытия и прироста. При оценке состояния фондов необходимо проанализировать их изношенность и прогрессивность обновления. Наблюдаемый процесс воспроизводства фондов должен быть рассмотрен во взаимосвязи с их использованием посредством анализа динамики фондоотдачи, а для активной части фондов необходимо проанализировать уровень загрузки в течение года и интенсивность использования в течение смены, рабочего дня.
- Анализ использования трудовых ресурсов. Основным показателем, оценивающим уровень использования трудовых ресурсов, является производительность труда, которая определяется по предприятию в целом через объёмы реализованной продукции, выполненных работ и услуг. Анализируя динамику производительности труда, необходимо рассмотреть влияние на нее таких факторов как: интенсивность использования фондов, фондовооружённость, уровень дохода работника предприятия с учетом оплаты труда, дивидендов, процентов и выплат по долевым паям, общих выплат социального характера за счет средств предприятия. Темпы изменения заработной платы необходимо сопоставить с темпами изменения производительности труда. Оценка эффективности использования трудовых ресурсов предприятия может быть произведена через прирост объёмов выручки от реализации продукции (работ, услуг) или снижения себестоимости ее производства. В рассматриваемом случае возможно использование, кроме стоимостных, натуральных показателей.
- Эффективность использования трудовых и материальных ресурсов. Эффективность использования трудовых и материальных ресурсов, привлекаемых в процессе производства продукции, находит отражение

в себестоимости, снижение которой является показателем положительных тенденций в экономике любого предприятия. Поэтому анализируя себестоимость необходимо:

- исследовать динамику изменения затрат;
- определить факторы, оказавшие влияние на динамику показателей себестоимости;
- установить причины отклонений фактических затрат от их уровня в бизнес-плане;
- выявить резервы возможного снижения себестоимости продукции (работ, услуг);
- разработать мероприятия и виды управленческих решений по практической реализации выявленных резервов повышения эффективности использования ресурсов.

С этой целью следует провести анализ затрат на производство и реализацию продукции, работ и услуг по экономическим элементам и калькуляционным статьям себестоимости; затрат на один рубль продукции (работ, услуг) по основному виду деятельности; себестоимости отдельных видов продукции (работ, услуг); расходов на обслуживание, производство и управление.

В качестве обобщающего показателя оценки изменений себестоимости необходимо использовать уровень затрат на один рубль продукции (работ, услуг). При анализе необходимо выявить степень влияния следующих факторов: себестоимости отдельных видов продукции (работ, услуг); цен на продукцию, работы, услуги, материалы, сырье, ремонтный фонд, запасные части, тарифы на электроэнергию и транспортные перевозки и т.д., что входит в качестве затрат в себестоимость; структурных сдвигов в составе продукции (ассортименте).

- Анализ рентабельности и доходность реализованной продукции. Учитывая экономический смысл показателей рентабельности продукции, их анализ проводится путём сравнения значений по годам анализируемого периода, показателями бизнес-плана и в разрезе видов производимой продукции, работ и услуг. При этом преследуется цель определения экономической целесообразности конкретных направлений (видов продукции) для инвестирования.
- Оценка рентабельности использования производственных ресурсов. В дипломных проектах целесообразно рассмотреть рентабельность основных производственных фондов и материальных оборотных средств, а также средств, направленных на оплату труда и социальные выплаты.

Исходя из полученных результатов, необходимо сделать выводы о «слабых» сторонах использования ресурсов (производственных фондов, оборотных средств, рабочей силы), что может служить обоснованием ре-

шения основной проблемы, сформулированной в теме дипломного проекта.

Проведение анализа основных производственно-экономических показателей возможно с использованием следующих направлений.

Анализ и определение показателей производственно-хозяйственной деятельности предприятия.

Производится составлением двух таблиц или одной комбинированной на основе определения соответствующих показателей в расчете на 100 га сельскохозяйственных угодий и 1 тыс. руб. денежной оценки пашни. Для заполнения таблицы использованы сходные данные, а также дополнительные показатели: валовой доход и чистый доход (прибыль). Валовой доход определяется как сумма прибыли от реализации продукции и размера заработной платы.

Прибыль в растениеводстве и животноводстве можно рассчитать как разность выручки от реализации продукции и полной себестоимости реализованной продукции.

Экономические показатели в расчете на 100 га сельскохозяйственных угодий определяются по формуле:

$$ВП_{100} = \frac{ВП}{0,01 \cdot S_{сх}},$$

где $ВП_{100}$ - валовая продукция на 100 га с.-х. угодий, тыс. руб.;

$ВП$ - валовая продукция сельскохозяйственного производства, тыс.руб.;

$S_{сх}$ - площадь с.-х. угодий предприятия, га;

$0,01$ - 100 га. с.-х. угодий.

При определении экономических показателей в расчете на 1 тыс. руб. денежной оценки необходимо использовать следующие методические положения.

Оценку земельных ресурсов проводят на основе земельно-оценочных работ по каждому району и с.-х. предприятию. Пашню и многолетние насаждения оценивают по окупаемости затрат, а природные кормовые угодья (сенокосы и пастбища) - по их продуктивности.

Для пашни и многолетних насаждений за 100 баллов принимается средняя по стране окупаемость 1 руб. затрат, равная 2,65 руб., которая соответствует стоимости продукции растениеводства по кадастровым ценам в расчете на 1 руб. затрат.

Денежная оценка 1 га пашни каждого хозяйства равна его баллу по окупаемости затрат, умноженному на норматив Денежной оценки 1 балла - 100 руб. Этот норматив принят на основе дифференцированного дохода, полученного в расчете на 1 га пашни.

Величину бальной оценки находят путем деления показателя окупаемости затрат данного хозяйства (в руб.) на такой же показатель в среднем по стране (26,5 руб.) и умножения полученной величины на 100.

На основе методических положений определяем денежную оценку пашни по формуле: $C_n = B \cdot C_b$,

где C_n - денежная оценка пашни, тыс. руб.;

B - качественная оценка почв (пашни) в баллах;

C_b - норматив денежной оценки 1 балла пашни площадью 1 га, тыс. руб.;

$S_{сх}$ - площадь пашни, га.

Экономические показатели в расчете на 1 тыс. руб. денежной оценки пашни определяются аналогично по формуле:

$$B_{np} = \frac{ВП}{C_n} \cdot 1000,$$

где B_{np} - валовая продукция на 1 тыс. руб. денежной оценки пашни.

Результаты расчетов сводятся в табл. 3.1.

Таблица 3.1

Показатели использования земли

Показатели	Значение показателей			
	20__	20__	20__	в среднем за три года
Валовой продукции, тыс. руб.				
в том числе				
- растениеводства				
- животноводства				
Зерна в массе после доработки, ц.				
Молока, ц.				
Валового дохода в целом по предприятию				
- растениеводства				
- животноводства				
Чистого дохода прибыли (убытка) от реализации продукции, тыс. руб.				
- растениеводства				
- животноводства				

Необходимо проанализировать полученные результаты, сделать выводы и сформулировать предложения по дальнейшему повышению эффективности земельных ресурсов.

Определение оснащенности предприятий фондами и энергетическими ресурсами, оценка эффективности использования фондов и капитальных вложений.

Оценку эффективности капитальных вложений, путей улучшения их использования строится на выполнении заданий:

первое - по фондам и энергетическим ресурсам;

второе - по капитальным вложениям.

Анализ сводится к определению значений показателей по предприятиям, их сравнительной оценке, анализу, формированию выводов и предложений по повышению эффективности использования фондов и энергетических ресурсов. При этом показатели обеспеченности определяются в расчете на 100 га сельскохозяйственных угодий по следующим зависимостям:

фондообеспеченность (Φ_0):

$$\Phi_0 = \frac{ОПФ}{S_{cx}} \cdot 100 = \frac{ОПФ}{0,01 \cdot S_{cx}};$$

энергообеспеченность (\mathcal{E}_0):

$$\mathcal{E}_0 = \frac{\mathcal{E}_m}{S_{cx}} \cdot 100 = \frac{\mathcal{E}_m}{0,01 \cdot S_{cx}};$$

электрообеспеченность ($\mathcal{E}_{до}$):

$$\mathcal{E}_{до} = \frac{\mathcal{E}_{лн}}{S_{cx}} \cdot 100 = \frac{\mathcal{E}_{лн}}{0,01 \cdot S_{cx}},$$

где ОПФ - основные производственные фонды, тыс.руб.;

\mathcal{E}_m - энергетические мощности предприятия, л. с.;

$\mathcal{E}_{лн}$ - годовой расход электроэнергии на производственные нужды, тыс. кВт. ч.;

S_{cx} - площадь с.-х. угодий, га;

0,01 - 100 га с.-х. угодий.

Показатели вооруженности оцениваются в расчете на среднегодового работника, соответственно:

фондовооруженность (Φ_v): $\Phi_v = \frac{ОПФ}{P};$

энерговооруженность (\mathcal{E}_v): $\mathcal{E}_v = \frac{\mathcal{E}_m}{P};$

электровооруженность (\mathcal{E}_{lv}): $\mathcal{E}_{lv} = \frac{\mathcal{E}_{лн}}{P},$

где P - среднегодовая численность работников предприятия, занятых в сельском хозяйстве.

Приведенные показатели связаны непосредственно с показателями обеспеченности и оказывают прямое влияние на производительность труда.

Экономическая эффективность использования производственных

фондов характеризуется системой показателей. Основными из них являются фондоотдача ($\Phi_{от}$) и фондоемкость ($\Phi_{с}$) продукции. Фондоотдача может быть определена по формулам:

$$\Phi_{вн} = \frac{ВП}{ОПФ} \quad (1); \quad \Phi_{вд} = \frac{ВД}{ОПФ} \quad (2); \quad \Phi_{д} = \frac{Д}{ОПФ} \quad (3),$$

где $\Phi_{вн}$, $\Phi_{вд}$, $\Phi_{д}$ - соответственно фондоотдача по валовой продукции, валовому доходу, чистому доходу (прибыли);

V_n , V_d , D - соответственно валовая продукция, валовой доход, чистый доход (прибыль).

При оценке эффективности использования основных фондов (формула 1) следует иметь в виду, что в состав валовой продукции включается перенесенная стоимость (в неизменных ценах) основных и оборотных фондов. Поэтому, чем больше она, тем выше стоимость валовой продукции, наблюдается двойной счет.

При определении фондоотдачи по валовому (формула 2) и чистому (формула 3) доходу отсутствуют затраты прошлого труда (семена, корма, топливо, электроэнергия, амортизация и др.).

Обратным показателем фондоотдачи является показатель фондоемкости, который показывает объем задействованных фондов в расчете на единицу валовой продукции, валового и чистого дохода.

Результаты расчетов сводятся в табл. 3.2.

Таблица 3.2

Оснащенность предприятий фондами и энергетическими ресурсами, эффективность их использования

№ пп	Показатели	Значение показателей			
		20__	20__	20__	в среднем за три года
1	Фондообеспеченность, тыс. руб.				
2	Энергообеспеченность, л.с.				
3	Электрообеспеченность, тыс.квт-ч				
4	Фондовооруженность, тыс. руб.				
5	Энерговооруженность, л.с.				
6	Электровооруженность, тыс.квт-ч				
7	Фондоотдача:				
	а) по валовой продукции				
	б) по валовому доходу руб.				
	в) по чистому доходу				
8	Фондоемкость:				
	а) валовой продукции				
	б) валового дохода руб.				
	в) чистого дохода				

Необходимо проанализировать полученные результаты, сделать выводы и сформулировать предложения по дальнейшему повышению эффективности земельных ресурсов.

Оценка уровня производительности труда, факторов и путей ее повышения.

Оценку уровня использования трудовых ресурсов производят по коэффициенту использования годового фонда рабочего времени (K_{up}), который можно определить из соотношения фактически отработанного времени к возможному (номинальному) фонду рабочего времени:

$$K_{up} = \frac{\Phi_o}{\Phi_n},$$

где $\Phi_o = \frac{T_p}{P}$ - фактически отработано часов в году, ч;

T_p - затраты труда на производство продукции, всего, чел.-ч;

P - среднегодовая численность работников, занятых в с.-х. производстве, чел.;

Φ_n - номинальный фонд рабочего времени, ч.

В расчетах принять $\Phi_n = 2016$ ч. (252 рабочих дня при 7 - часовой продолжительности рабочего дня).

Для определения производительности труда (Π_{mp}) необходимо использовать следующие аналитические зависимости:

$$\Pi_{mp}^n = \frac{B_n}{T_p}; \quad \Pi_{mp}^p = \frac{B_{np}}{T_{pp}}; \quad \Pi_{mp}^{жс} = \frac{B_{пжс}}{T_{ржс}},$$

где $\Pi_{mp}^n, \Pi_{mp}^p, \Pi_{mp}^{жс}$ - произведено валовой продукции (производительность труда) соответственно в целом по предприятию, в растениеводстве и животноводстве, тыс. руб. на 1 чел.-ч;

$B_n, B_{np}, B_{пжс}$ - объем валовой продукции, соответственно в целом по предприятию, в растениеводстве и животноводстве, тыс. руб.;

$T_p, T_{pp}, T_{ржс}$ - затраты труда на производство продукции, соответственно в целом по предприятию, в растениеводстве и животноводстве, чел.-ч.

Производительность труда может быть оценена количеством валовой продукции в расчете на среднегодового работника.

Этот показатель отражает не только уровень производительности труда, но и степень использования рабочей силы в течение года.

Дополнительным показателем оценки производительности живого труда является отношение валового дохода (стоимость валовой продукции за вычетом материальных затрат) к затратам рабочего времени. В связи с этим величину производительности труда необходимо определить по следующим формулам:

$$\Pi_{mp}^n = \frac{B_p}{T_p}; \quad \Pi_{mp}^p = \frac{B_\Delta^p}{T_{pp}^p}; \quad \Pi_{mp}^{жс} = \frac{B_\Delta^{жс}}{T_{pжс}},$$

где $B_\Delta, B_\Delta^p, B_\Delta^{жс}$ - величина валового дохода, соответственно полученного в целом по предприятию в растениеводстве и животноводстве, тыс. руб.

При определении производительности труда производственного коллектива целесообразно использовать вместо валового дохода хозрасчетный доход, который складывается из фонда заработной платы и прибыли, остающейся в распоряжении коллектива.

Принимая во внимание, что трудоемкость продукции - это есть показатель затрат рабочего времени на производство единицы продукции, в задании необходимо определить трудоемкость производства 1 ц зерна (T_p^z) и молока (T_p^m). Для этих целей используются аналитические зависимости вида:

$$T_p^z = \frac{T_{pz}}{B_{сз}}; \quad T_p^m = \frac{T_{pm}}{B_{нм}},$$

где T_{pz}, T_{pm} - прямые затраты труда, соответственно на производство зерна и молока, чел.-ч;

$B_{сз}, B_{нм}$ - валовой сбор зерна и валовой надой молока, ц.

Для сравнения роста производительности труда и заработной платы, которая определяется на 1 среднегодового работника по формуле :

$$Z_n^c = \frac{Z_n}{P},$$

где Z_n - фонд заработной платы предприятия, тыс. руб.

Решение задачи осуществляется путем выполнения расчетов, приведенных выше показателей и сравнительного анализа, полученных результатов по предприятиям. Величины показателей уровня производительности труда, использования трудовых ресурсов, а также факторов повышения производительности труда сводят в табл. 3.3.

Для формирования выводов и предложений при сравнении показателей таблицы 3.3 следует обратить внимание не только на разницу в производительности труда по предприятиям, но и на факторы ее, обеспечивающие. К их числу можно отнести показатели вооруженности, труда и обеспеченности предприятий фондами и энергетическими ресурсами, использования фонда рабочего времени. Изменение производительности труда в растениеводстве и животноводстве необходимо увязать с динамикой трудоемкости производства одного центнера зерна и молока. Сопоставляя изменения заработной платы одного работника, с производительностью труда следует иметь в виду, что опережающие темпы роста производительности труда над его оплатой являются объективной закономерностью нормально работающих предприятий.

Таблица 3.3

Результаты расчетов показателей производительности труда

№ пп	Показатели	Единицы измерения	Значение показателей		
			1-е предприятие	2-е предприятие	1-е ко 2-му, в %
1	Коэффициент использования годового фонда рабочего времени		1,11	1,01	
2	Произведено валовой продукции в целом по предприятию: а) в расчете на одного среднегодового работника б) в расчете на 1 чел.-ч. затраченного труда	тыс. руб.	24,8	47,80	
			11,1	23,53	
3	Произведено валовой продукции в расчете на 1 чел.-ч. затраченного труда: а) в животноводстве б) в растениеводстве	тыс. руб.	16,2	11,99	
			8,6	26,67	
4	Получено валового дохода в расчете на 1 чел.-ч затраченного труда: а) в целом по предприятию б) в растениеводстве в) в животноводстве	тыс. руб.			
5	Показатели: фондообеспеченность энергообеспеченность электрообеспеченность фондовооруженность энерговооруженность электровооруженность	тыс. руб.	443,3	465,9	
			л.с.	115,5	192,54
		кВт-ч	11,4	8,99	
			тыс. руб.	75,4	98,55
		л.с.	19,6	40,73	
			кВт-ч	1,9	1,9
6	Трудоемкости производства на 1 ц: а) зерна б) молока	чел.-ч	0,27	0,22	
			28,44	9,52	
7	Среднегодовая заработная плата работника	тыс. руб.	4,83	6,27	

Необходимо проанализировать полученные результаты, сделать выводы и сформулировать предложения по дальнейшему повышению эффективности земельных ресурсов.

Направление, содержание и методика анализа деятельности объекта должны определяться исходя из направления исследования и темы дипломного проекта по согласованию с дипломным руководителем.

При поиске резервов повышения эффективности производства наряду с традиционными приёмами анализа рекомендуется использовать экономико-математические методы. Причём в каждом конкретном случае следует тщательно оценить целесообразность и возможность применения того или иного метода.

1.3. Экономическая сущность комплекса решаемых задач

По результатам проведенного в предыдущем разделе анализа деятельности объекта необходимо выделить и сформулировать основные направления совершенствования управленческой деятельности по функциональным экономическим подсистемам либо их комплексам.

Основой направлений совершенствования следует принимать выявленные при анализе резервы повышения эффективности управления экономическим объектом. При этом направления совершенствования должны опираться на использование и внедрение методик решения экономических задач, рационализацию организационных и управленческих структур объектов исследования, оптимизационные и экономико-математические модели и т.д.

Реализация предложенных экономических задач может повлечь следующие трудности:

- невозможность расчета показателей, необходимых для управления объектом, из-за сложности вычислений или чрезмерного объема информации;
- большая трудоемкость обработки информации;
- низкая оперативность, снижающая качество управления объектом;
- невысокая достоверность результатов решения задачи из-за дублирования потоков информации;
- несовершенство организации сбора и регистрации исходной информации;
- несовершенство процессов сбора, передачи, обработки и выдачи информации.

В связи с чем, необходимо описать методы информационного обеспечения реализации экономических задач.

Глава 2. Теоретические основы решения комплекса задач

В главе раскрываются теоретические основы решения экономического и информационного комплекса задач дипломного проектирования.

2.1. Теоретические основы решения экономических задач

В разделе необходимо отразить теоретические основы решаемой экономической задачи, её место в системе экономических наук.

Раскрываются принципы процессов производства с.-х. продукции,

организации использования машин и оборудования в АПК, обеспечение их работоспособности, исследованию которых посвящён дипломный проект, показывается его народнохозяйственное значение, а отсюда и необходимость правильной организации процессов производства, учёта и анализа, раскрываются значение, задачи учёта и анализа объекта исследования. При этом следует изучить, аргументировано и логически изложить основные положения и методики организации и планирования производства, эксплуатации машин и оборудования в АПК и практической реализации других вопросов, относящихся к теме дипломного проекта.

Здесь же следует остановиться на методах и организации планирования, учёта и анализа в конкретной организации, последовательности сбора и формирования информационной базы по избранному для исследования участку. Желательно, чтобы в дипломном проекте было кратко освещено положение дел по рассматриваемому кругу вопросов в передовых предприятиях страны с использованием статистических данных, материалов научных монографий и статей, опубликованных в печати.

2.2. Теоретические основы и технологии информационного обеспечения решения комплекса экономических задач

Обоснование необходимости и цели использования вычислительной техники для решения комплекса задач в процессе декомпозиции в зависимости от результатов решения комплекса из него можно на первом этапе выделить ряд отдельных задач. Например, если результатом решения является набор выходных документов, тогда технологический процесс формирования каждого выходного документа или группы однотипных выходных документов можно считать отдельной задачей. Далее следует перечислить выделенные задачи и для каждой отдельной задачи привести характеристику предметной технологии её решения, а также описать связи данной задачи с другими задачами, входящими в комплекс.

Характеризуя предметную технологию решения каждой задачи, следует отразить ее особенности, то есть принятые нормы и правила осуществления анализируемой деятельности, в условиях конкретной организации или предприятия. Например, в ходе декомпозиции комплекса задач по учету основных средств и анализа составляющих его задач следует указать порядок постановки на учет, выбытия, переоценки основных средств, используемые методы и нормы расчета амортизации, практикуемые бухгалтерские проводки.

При этом следует выделять последовательность решения задач комплекса и затем осуществлять разбиение каждой задачи на отдельные этапы и операции.

При этом необходимо произвести обоснованный выбор обеспечивающих технологий, определить функциональные информационные тех-

нологии, решения задач в зависимости от целей исследования.

Обоснование выбора обеспечивающих технологий включает в себя определение программных и аппаратных средств, необходимых для создания АИС.

При выполнении работы по обоснованию выбора аппаратной части проекта следует обратить внимание на ряд следующих специфических моментов:

- обоснованию выбора типа архитектуры: файл-сервер, клиент – сервер, многоуровневая архитектура;
- обоснованию выбора типа ЭВМ для клиентской и серверной части архитектуры;
- обоснованию выбора типа локальных сетей и сетевых операционных систем;
- определению способов технического обеспечения обработки распределенных транзакций и защиты хранимых данных и т.п.

Обоснование проектных решений по программному обеспечению комплекса задач заключается в формировании требований к системному и прикладному программному обеспечению и выборе на основе этих требований соответствующих компонентов программного обеспечения. Например, к большинству прикладного программного обеспечения можно выдвинуть требования надежности, эффективности, понятности пользователю, защиты информации, модифицируемости, мобильности, масштабируемости, минимизации затрат на сопровождение и поддержку и т.д. Кроме того, стоит выработать требования к оформлению экранных и печатных форм, эргономике программного обеспечения.

Формулировка требований к программному обеспечению должна происходить с учетом объема информационных потоков объекта управления, требований и особенностей существующей предметной технологии, структуры системы управления.

При обосновании проектных решений по программному обеспечению комплекса целесообразно:

- обосновать выбор средств проектирования и разработки информационного обеспечения (СУБД), прикладного программного обеспечения (методов и среды разработки прикладных программ, языков программирования, специализированных библиотек);
- определить возможности выбранных программных средств, при использовании которых достигаются требования к прикладному программному обеспечению (например, возможность организации удобного интерфейса, оптимизации запросов к данным и т.п.);
- определить состав разрабатываемых процедур обработки данных клиентской части корпоративной АИС.

Выбор методов и средств проектирования и разработки по возможности необходимо аргументировать, сравнивая их с аналогичными средствами, существующими на рынке.

Для определения состава обеспечивающих информационных технологий необходимо для каждого технологического этапа предметной технологии рассмотреть целесообразность и границы автоматизации данного этапа предметной технологии, а затем, в случае необходимости его автоматизации, обосновать выбор соответствующей обеспечивающей технологии.

В соответствие с организационной структурой субъекта управления следует рассмотреть необходимость создания комплексов программных средств для реализации целей и обязанностей сотрудников, входящих в систему управления. Далее необходимо обосновать состав каждого комплекса. При обосновании состава технологий важно отмечать недостатки и достоинства выбранных технологий и предложить подходы и решения для их устранения недостатков.

В ходе обоснования технологических решений необходимо отразить следующие вопросы:

- методы сбора исходной информации (централизованно, распределено), и анализ целесообразности использования специальных технических средств (датчиков, счетчиков и т.п.);
- формы передачи данных между АРМами и для внешней среды (на бумажных или машинных носителях), средства передачи (по локальной сети, модемной связи, по электронной почте и т.п.), порядок передачи данных (автоматически, после получения, по запросу, при определенных условиях);
- технологии обработки информации (централизованная, децентрализованная, распределенная), режим обработки данных (пакетный, диалоговый);
- формы и способы выдачи информации пользователю АРМ (на принтер, на экран монитора, в файл, централизованно, децентрализованно, распределено);
- способы обеспечения защиты целостности и секретности хранимых данных в базах данных;
- способы и методы обеспечения процедуры восстановления данных в базах данных и др.

2.3. Обзор существующих решений экономических задач

В данном разделе следует отметить, используются ли при существующей технологии решения комплекса задач какие-либо программные средства и, если используются, то каким образом.

Затем следует отметить, чем должна и будет отличаться технология

решения комплекса задач в проектируемой АИС от существующей, а также почему необходимо разрабатывать новое программное средство, и чем оно должно отличаться от существующих.

Если на рынке программных средств существуют готовые программные (программно - аппаратные) решения желательно дать краткое описание и провести анализ хотя бы одной такой разработки, указав основные характеристики (например, понятность пользователю, степень защиты информации, модифицируемость, мобильность, масштабируемость, затраты на сопровождение и поддержку и т.д.) и функциональные возможности.

Обзор рынка программных средств удобно проводить с помощью Internet. Адреса используемых при обзоре ресурсов следует добавить в список литературы дипломного проекта.

2.4. Обоснование методик и технологий решения комплекса задач

Обоснование выбора информационных средств для решения конкретных экономических задач представляет собой достаточно сложную проблему, так как современные вычислительные машины являются сложными системами.

Информационные средства *информационного обеспечения* обосновываются с точки зрения внемашинного и внутримашинного обеспечения и включают следующие вопросы:

- основные принципы проектирования информационного обеспечения комплекса задач;
- обоснование состава и содержания результатных массивов и выходных документов;
- обоснование состава, формы представления исходной информации в первичных документах и на машинных носителях;
- обоснование требований к системам классификации и кодирования информации.

В данной главе также необходимо уделить внимание обоснованию методов организации информационной базы. Здесь следует рассмотреть следующие вопросы:

- обоснование выбора формы хранения данных (база данных или совокупность локальных файлов);
- обоснование выбора модели логической структуры базы данных (иерархической, сетевой, реляционной);
- обоснование методов организации информационных массивов (преобразов файлов), ключей упорядочения и т.д.

При выборе информационного обеспечения создаваемой системы наиболее важными являются следующие узлы выбора альтернативных решений:

- определение целесообразности использования интегрированной базы данных;
- выбор СУБД;
- выбор структуры автономных файлов;
- использование диалога.

По каждому из названных узлов выбора альтернативных решений необходимо определить основные факторы, влияющие на этот выбор. Их ранжирование, определение удельного веса, получение интегрированной оценки и, следовательно, выбор альтернативного варианта определяется в каждом случае в соответствии с особенностями конкретной ситуации.

В качестве этих факторов выделим следующие:

1. Определение целесообразности использования интегрированной базы данных (БД): сложность информации; разнообразие запросов; объем информации; объем корректировок; возможности ЭВМ (память, программное обеспечение, надежность).

2. Использование диалога: требования пользователя; разнообразие запросов; объемы информации; возможности ЭВМ; надежность; время реакции на запрос; простота работы пользователя.

3. Выбор структуры автономных файлов: необходимая память; время на корректировку; надежность; время решения задачи.

4. Выбор СУБД: структура информации; возможности ЭВМ; наличие программного обеспечения; широта программного окружения СУБД; наличие сети ЭВМ; время реакции на запрос.

Обоснование проектных решений *по технологии сбора, передачи, обработки и выдачи информации* включают характеристику существующей технологии и подготовку предложений по ее совершенствованию, отражая:

- выбор способа сбора исходной информации на основе анализа целесообразности использования технических средств сбора (регистраторов производства, датчиков, счетчиков и т.д.);
- обоснование методов передачи информации в АИС (курьером, в форме документов, по каналам модемной связи, по каналам ЛВС, с использованием выделенных каналов, дискретным способом через дискеты, стримеры, оптические носители и т.п., в интерактивном режиме);
- обоснование методов обеспечения достоверности информации (верификация, счетный контроль и т.д.);
- обоснование технологии выдачи информации пользователю (централизованная, децентрализованная, распределенная, и т.д.; на принтер, на экран монитора, в файл).

Обоснование проектных решений *по программному обеспечению* комплекса задач заключается в формировании требований к системному,

специальному и прикладному программному обеспечению. Целесообразно:

- обосновать выбор соответствующего инструментального средства (языки программирования, специализированные библиотеки, СУБД, системы автоматизированного проектирования, системы класса CASE и др.) и среды, в которой предполагается использование разрабатываемой АИС;
- определить цель проектирования рациональной внутримашинной технологии обработки на основе выбранных инструментальных средств (например, сокращение времени обработки по сравнению с тем, что существует в настоящий момент за счет улучшенных механизмов поиска и сортировок, которые обеспечивает выбранный инструментарий; минимизация затрат на разработку и дальнейшее сопровождение ПО; обеспечение надежности АИС и защиты информации и т.д.);
- раскрыть сущность методов проектирования рациональной внутримашинной технологии (например, сокращение числа сортировок, использование эффективных методов поиска информации, процедурно-ориентированных подходов к выделению модулей, оптимизация алгоритмов и т.д.);
- определить функции управляющей программы;
- обосновать использование режимов обработки данных в проектируемой АИС. При каких обстоятельствах будет использоваться пакетный режим, в каких случаях диалоговый и т.д.;
- выработать требования к оформлению экранных и печатных форм, эргономике программного обеспечения.

Выбор одного из вариантов внутримашинной технологии обработки данных тесно связан с его обоснованием, при проведении которого в дипломном проекте целесообразно исходить из специфики проектируемого процесса.

В настоящее время широко используются пакетный и диалоговый режимы обработки данных, причем последний не является альтернативой первого, а может рассматриваться скорее как его развитие. Выбор того или иного режима вытекает из особенностей каждого из них и особенностей решаемой задачи.

Характеризуя пакетный режим обработки данных, необходимо отметить следующие его отличительные черты. Ввод потока заданий осуществляется с локальных устройств ввода. Выполнение режима включает три фазы обработки: подготовку, выполнение и завершение процесса. При этом первая фаза требует определения последовательности действий и ввода исходных данных. Вторая фаза предполагает логическое преобразование исходных файлов, создания и упорядочения рабочих файлов, обработку информации и формирование выходных данных, осуществляя кон-

троль результатов решения. На завершающей фазе выполняется печать. Эти особенности необходимо рассмотреть в связи со спецификой функциональной задачи.

Применение пакетного режима позволяет уменьшить вмешательство оператора в процесс решения задачи, требует только предварительного ввода данных, исключает возможность вмешательства пользователя и, таким образом, изменяет последовательность выполняемых действий. Однако, за счет этого появляется более полная загрузка оборудования, которое начинает работать по жесткому графику. В некоторых случаях для решения задачи выполняется и параллельная обработка данных. Пакетный режим более тесно связан с бумажной технологией.

Диалоговый режим, напротив, предполагает активное вмешательство пользователя в процесс работы комплекса и ориентацию на безбумажную технологию. В ходе его выполнения отсутствует заранее установленная последовательность операций обработки данных и дополнительного их ввода.

В процессе решения задачи удобство диалогового режима в полной мере проявляется в процессе общения с базой данных.

Среди них можно отметить:

- возможность перебора различных комбинаций поисковых признаков в запросе;
- обеспечение более быстрого поиска данных;
- улучшение характеристик выходных данных за счет оперативной коррекции запроса с терминала;
- возможность расширения, сужения или изменения направлений поиска сразу после получения результатов;
- множественность точек доступа;
- быстрый доступ к относительно редко используемой информации;
- оперативный анализ получаемых сведений.

В настоящее время в развитии вычислительной техники наметилась тенденция к рассредоточению вычислительных мощностей в пределах вычислительных систем. Все большее распространение приобретают вычислительные системы, в которых применяется распределенная обработка данных.

Это предопределило главную особенность тенденции - приближение ЭВМ непосредственно к местам возникновения и использования информации, их распределению по отдельным функциональным сферам деятельности, а, следовательно, и к изменению самой технологии обработки данных в направлении децентрализации. Структурно они реализуются как сети взаимосвязанных через каналы передачи данных ЭВМ, терминалов с одной или несколькими средними либо большими ЭВМ.

При обосновании применения распределенных систем обработки данных необходимо отметить их особенности: большое количество взаимодействующих вычислительных машин, выполняющих функции сбора, регистрации, хранения, передачи, обработки и выдачи информации; значительные вычислительные мощности; распределение обработки, хранения и использования данных; доступ пользователя к вычислительным и информационным ресурсам сети; симметричный интерфейс обмена данными между всеми узлами сети; возможность управления всеми элементами сети и ее расширяемость.

В связи с многообразием создаваемых сетей они классифицируются по ряду признаков: технологической структуре (централизованная, децентрализованная, кольцевая, радиально кольцевая и др.); организации связи (с коммутацией каналов, с коммутацией сообщений, с коммутацией пакетов и др.); функциональному назначению (универсальные и специализированные); организации данных (без банков данных, с локальными банками данных, с централизованным банком данных).

Далее необходимо рассмотреть организацию локальной сети на логическом уровне:

- рабочую систему, реализующую информационные процессы, связанные с организацией, хранением, поиском и вычислительной обработкой данных;
- терминальную систему, управляющую работой терминального оборудования и осуществляющая подготовку заданий пользователей, сопряжение пунктов съема данных;
- административную систему, управляющую процессами функционирования информационно-вычислительной сети;
- интерфейсную систему, реализующую функции, связанные с преобразованием процедур управления и передаваемой информации в условиях взаимодействия с другими сетями;
- коммуникационную, ориентированную на выполнение функций по обеспечению взаимодействия всех систем (управления потоками данных, их маршрутизация и коммутация).

Наметившаяся тенденция децентрализации средств вычислительной техники послужила предпосылкой развития на базе персональных микропроцессорных средств автоматизированных рабочих мест (АРМ).

Обоснование применения АРМ следует начать с рассмотрения их возможностей:

- информационно-справочное обслуживание;
- автоматизация делопроизводства;
- развитый диалог пользователя с ЭВМ;
- использование ресурсов, как ПЭВМ, так и центральной ЭВМ для решения различных задач;

- формирование и ведение локальных баз данных и использование централизованной базы данных при наличии вычислительной сети;
- представление сервиса пользователю на рабочем месте.

Далее необходимо рассмотреть такие преимущества АРМ, как надежность, низкая стоимость, сочетание автономного и многопользовательского режимов работы, возможность реализации интерфейса АРМ друг с другом и с большой ЭВМ, удобство подключения новых внешних устройств. Учитывая конкретику целевого назначения АРМ, необходимо исходить в обосновании из принципа максимальной ориентации на конечного пользователя, что обычно достигается адаптацией АРМ к уровню его подготовки и возможностям его обучения и самообучения. В свою очередь этот принцип тесно связан с принципом проблемной ориентации, то есть с ориентацией на решение определенного класса задач, объединенных общей технологией обработки данных, единством режимов эксплуатации. В узком смысле, проблемная ориентация заключается в ориентации на автоматизацию конкретных функций, выполняемых работниками экономических служб.

Следует отметить также уровень развития АРМ, среди которых выделяют: построение типовых (базовых) АРМ, ориентированных на группы конкретных пользователей; реализация на базе типовых АРМ специализированных (функциональных АРМ, например, АРМ бухгалтера, АРМ аналитика); объединение специализированных АРМ в проблемно-ориентированные комплексы в рамках локальных распределенных систем обработки данных.

Возможности АРМ обычно тесно связаны с их структуризацией и параметризацией, зависят от функциональных характеристик ПЭВМ, на которых они базируются. После рассмотрения этих вопросов нужно остановиться на обеспечивающей части АРМ: вопросах организации информационной базы; вопросах специфики программного обеспечения; вопросах обоснования общей технологии обработки данных; вопросах лингвистического обеспечения, диалога; вопросах методического обеспечения, ГОСТов.

Описанные выше разделы составляют первую часть проекта. Наиболее распространенной ошибкой при подготовке первой главы дипломного проекта является то, что студент пытается перемешать существующее положение дел на предприятии, выявленных в ходе обследования с открывающимися перспективами и технологией функционирования описываемой в дипломном проекте АИС.

Глава 3. Проектирование АИС

Проектная часть дипломного проекта является описанием решений, принятых по всей вертикали проектирования. Глава должна быть основана

на информации, представленной в аналитической части, обобщать ее. По сути, проектная часть является решением проблематики, изложенной в аналитической части, на языке информационных технологий. Поэтому недопустимо, если при проектировании используется информация об объекте управления, не описанная в первой главе.

3.1. Обоснование и выбор подхода к проектированию АИС

При проектировании АИС необходимо использовать один двух основных подходов:

1. Структурное проектирование;
2. Объектно-ориентированное проектирование.

Выбор подхода должен быть обоснован применительно к решению поставленных задач и аргументирован с точки зрения следующих аспектов:

- предметная область;
- модель жизненный цикл АИС;
- трудоемкость проектирования;
- коллективная работа над проектом;
- применимость автоматизированных средств проектирования и разработки.

3.2. Структурное проектирование

СУБД должна предоставлять доступ к данным любым пользователям, включая и тех, которые практически не имеют и (или) не хотят иметь представления о:

- физическом размещении в памяти данных и их описаний;
- механизмах поиска запрашиваемых данных;
- проблемах, возникающих при одновременном запросе одних и тех же данных многими пользователями (прикладными программами);
- способах обеспечения защиты данных от некорректных обновлений и (или) несанкционированного доступа;
- поддержании баз данных в актуальном состоянии и множестве других функций СУБД.

Проект базы данных надо начинать с анализа предметной области и выявления требований к ней отдельных пользователей (сотрудников организации, для которых создается база данных). Объединяя частные представления о содержимом базы данных, полученные в результате опроса пользователей, и свои представления о данных, которые могут потребоваться в будущих приложениях, сначала создается обобщенное неформальное описание создаваемой базы данных. Это описание, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающим над проектированием базы данных, называют инфологической моделью данных.

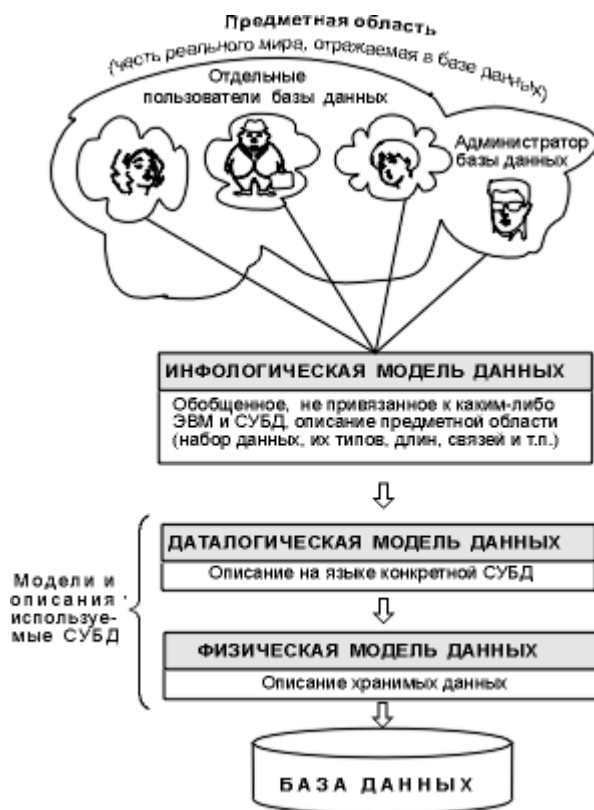


Рис. 3.1. Уровни моделей данных

Такая модель полностью независима от физических параметров среды хранения данных. Поэтому инфологическая модель не должна изменяться до тех пор, пока какие-то изменения в реальном мире не потребуют изменения в ней, чтобы эта модель продолжала отражать предметную область.

Остальные модели, показанные на рис. 3.1, являются компьютеро-ориентированными. С их помощью СУБД дает возможность программам и пользователям осуществлять доступ к хранимым данным лишь по их именам, не заботясь о физическом расположении этих данных. Нужные данные отыскиваются СУБД на внешних запоминающих устройствах по физической модели данных.

Так как указанный доступ осуществляется с помощью конкретной СУБД, то модели должны быть описаны на языке описания данных этой СУБД. Такое описание, создаваемое по инфологической модели данных, называют даталогической моделью данных.

Трехуровневая архитектура (инфологический, даталогический и физический уровни) позволяет обеспечить независимость хранимых данных от использующих их программ. При необходимости можно переписать хранимые данные на другие носители информации и (или) реорганизовать их физическую структуру, изменив лишь физическую модель данных, подключить к системе любое число новых пользователей (новых приложе-

ний), дополнив, если надо, даталогическую модель. Указанные изменения физической и даталогической моделей не будут замечены существующими пользователями системы, так же как не будут замечены и новые пользователи. Следовательно, независимость данных обеспечивает возможность развития системы баз данных без разрушения существующих приложений.

Инфологическая модель отображает реальный мир в некоторые понятные человеку концепции, полностью независимые от параметров среды хранения данных. Существует множество подходов к построению таких моделей: графовые модели, семантические сети, модель "сущность-связь" и т.д. Наиболее популярной из них на сегодняшний момент оказалась модель "сущность-связь".

Инфологическая модель должна быть отображена в компьютеро-ориентированную даталогическую модель, "понятную" СУБД. В процессе развития теории и практического использования баз данных, а также средств вычислительной техники создавались СУБД, поддерживающие различные даталогические модели.

Сначала стали использовать иерархические даталогические модели. Простота организации, наличие заранее заданных связей между сущностями, сходство с физическими моделями данных позволяли добиваться приемлемой производительности иерархических СУБД на первых ЭВМ с весьма ограниченными объемами памяти. Но, если данные не имели древовидной структуры, то возникала масса сложностей при построении иерархической модели и желании добиться нужной производительности.

Сетевые модели также создавались для мало ресурсных ЭВМ. Это достаточно сложные структуры, состоящие из "наборов" – поименованных двухуровневых деревьев. "Наборы" соединяются с помощью "записей-связок", образуя цепочки и т.д. При разработке сетевых моделей было выдумано множество "маленьких хитростей", позволяющих увеличить производительность СУБД, но существенно усложнивших последние. Прикладной программист должен знать массу терминов, изучить несколько внутренних языков СУБД, детально представлять логическую структуру базы данных для осуществления навигации среди различных экземпляров, наборов, записей и т.п.

Инфологическая модель данных "Сущность-связь"

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфоло-

гических моделей являются сущности, связи между ними и их свойства (атрибуты).

Сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе. Например, типом сущности может быть ГОРОД, а экземпляром – Москва, Киев и т.д.

Атрибут – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей (например, ЦВЕТ может быть определен для многих сущностей: ЯБЛОКО, АВТОМОБИЛЬ, ЦВЕТОК и т.д.). Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Примерами атрибутов для сущности АВТОМОБИЛЬ являются ТИП, МАРКА, НОМЕРНОЙ ЗНАК, ЦВЕТ и т.д. Здесь также существует различие между типом и экземпляром. Тип атрибута ЦВЕТ имеет много экземпляров или значений: Красный, Синий, Банановый, Белая ночь и т.д.; однако каждому экземпляру сущности присваивается только одно значение атрибута.

Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность. Например, для автомобильного завода цвет – это только атрибут продукта производства, а для лакокрасочной фабрики цвет – тип сущности.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.

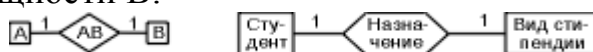
Связь – ассоциирование двух или более сущностей. Если бы назначением базы данных состояло только в хранении отдельных, не связанных между собой данных, то ее структура могла бы быть очень простой. Однако одно из основных требований к организации базы данных – это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи. А так как в реальных базах данных нередко содержатся сотни или даже тысячи сущностей, то теоретически между ними может быть установлено более миллиона связей. Наличие такого множества связей и определяет сложность инфологических моделей.

Характеристика связей и язык моделирования

При построении инфологических моделей можно использовать язык ER-диаграмм (от англ. Entity-Relationship, т.е. сущность-связь). В них сущности изображаются помеченными прямоугольниками, ассоциации – помеченными ромбами или шестиугольниками, атрибуты – помеченными овалами, а связи между ними – ненаправленными ребрами, над которыми может проставляться степень связи (1 или буква, заменяющая слово "много") и необходимое пояснение.

Между двумя сущностями, например, А и В возможны четыре вида связей.

Первый тип – связь ОДИН-К-ОДНОМУ (1:1): в каждый момент времени каждому представителю (экземпляру) сущности А соответствует 1 или 0 представителей сущности В:



Студент может не "заработать" стипендию, получить обычную или одну из повышенных стипендий.

Второй тип – связь ОДИН-КО-МНОГИМ (1:M): одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В.



Квартира может пустовать, в ней может жить один или несколько жильцов.

Так как между двумя сущностями возможны связи в обоих направлениях, то существует еще два типа связи МНОГИЕ-К-ОДНОМУ (M:1) и МНОГИЕ-КО-МНОГИМ (M:N).

В ER-диаграммах связь это линия, соединяющая геометрические фигуры, изображающие сущности, атрибуты, ассоциации и другие информационные объекты. В тексте же этот термин используется для указания на взаимозависимость сущностей. Если эта взаимозависимость имеет атрибуты, то она называется ассоциацией.

Классификация сущностей

Определяется три основных класса сущностей: *стержневые, ассоциативные и характеристические*, а также подкласс ассоциативных сущностей – *обозначения*.

Стержневая сущность (стержень) – это независимая сущность (подробнее она будет определена ниже), например "Студент", "Квартира", "Мужчины", "Врач" и т.д.

Ассоциативная сущность (ассоциация) – это связь вида "многие-ко-многим" ("1-ко-многим" и т.д.) между двумя или более сущностями или экземплярами сущности. Ассоциации рассматриваются как полноценные сущности: они могут участвовать в других ассоциациях и обозначениях точно так же, как стержневые сущности, а также могут обладать свойствами

ми, т.е. иметь не только набор ключевых атрибутов, необходимых для указания связей, но и любое число других атрибутов, характеризующих связь.

Характеристическая сущность (характеристика) – это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями (частный случай ассоциации). Единственная цель характеристики в рамках рассматриваемой предметной области состоит в описании или уточнении некоторой другой сущности. Необходимость в них возникает в связи с тем, что сущности реального мира имеют иногда многозначные свойства (книга – несколько характеристик переиздания - исправленное, дополненное, переработанное, и т.д.)

Существование характеристики полностью зависит от характеризующей сущности: (глава не может существовать без книги и т.д.).

Для описания характеристики используется изображения трапеции (рис. 3.2).

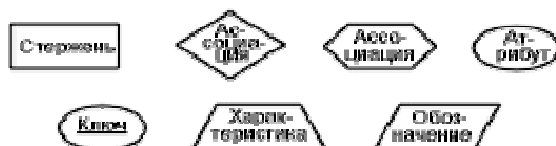


Рис. 3.2. Элементы расширенного языка ER-диаграмм

Обозначающая сущность или *обозначение* – это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями и отличается от характеристики тем, что не зависит от обозначаемой сущности.

Обозначения используют для хранения повторяющихся значений больших текстовых атрибутов: "кодификаторы" изучаемых студентами дисциплин, наименований организаций и их отделов, перечней товаров и т.п.

Обозначения и характеристики не являются полностью независимыми сущностями, поскольку они предполагают наличие некоторой другой сущности, которая будет "обозначаться" или "характеризоваться". Однако они все же представляют собой частные случаи сущности и могут, конечно, иметь свойства, могут участвовать в ассоциациях, обозначениях и иметь свои собственные (более низкого уровня) характеристики. Необходимо представлять, что все экземпляры характеристики должны быть обязательно связаны с каким-либо экземпляром характеризующей сущности. Однако допускается, чтобы некоторые экземпляры характеризующей сущности не имели связей.

Первичные и внешние ключи

Ключ или *возможный ключ* – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждая сущность обладает хотя бы одним возможным ключом. Один из них принима-

ется за *первичный ключ*. При выборе первичного ключа следует отдавать предпочтение несоставным ключам или ключам, составленным из минимального числа атрибутов. Нецелесообразно также использовать ключи с длинными текстовыми значениями (предпочтительнее использовать целочисленные атрибуты). Так, для идентификации студента можно использовать либо уникальный номер зачетной книжки, либо набор из фамилии, имени, отчества, номера группы и может быть дополнительных атрибутов, так как не исключено появление в группе двух студентов с одинаковыми фамилиями, именами и отчествами.

Не допускается, чтобы первичный ключ стержневой сущности (любой атрибут, участвующий в первичном ключе) принимал неопределенное значение. Иначе возникнет противоречивая ситуация: появится не обладающий индивидуальностью, и, следовательно не существующий экземпляр стержневой сущности. По тем же причинам необходимо обеспечить уникальность *первичного ключа*.

Теперь о *внешних ключах*:

- Если сущность С связывает сущности А и В, то она должна включать внешние ключи, соответствующие первичным ключам сущностей А и В.
- Если сущность В обозначает сущность А, то она должна включать внешний ключ, соответствующий первичному ключу сущности А.

Связь между первичными и внешними ключами сущностей иллюстрируется рис.3.3.



Рис. 3.3. Структуры: а - ассоциации; б - обозначения (характеристики)

Здесь для обозначения любой из ассоциируемых сущностей (стержневой, характеристик, обозначений или даже ассоциаций) используется новый обобщающий термин "Цель" или "Целевая сущность".

Таким образом, при рассмотрении проблемы выбора способа представления ассоциаций и обозначений в базе данных основной вопрос, на

который следует получить ответ: "Каковы внешние ключи?". И далее, для каждого внешнего ключа необходимо решить три вопроса:

1. Может ли данный внешний ключ принимать неопределенные значения (NULL-значения)? Иначе говоря, может ли существовать некоторый экземпляр сущности данного типа, для которого неизвестна целевая сущность, указываемая внешним ключом? В случае поставок это, вероятно, невозможно – поставка, осуществляемая неизвестным поставщиком, или поставка неизвестного продукта не имеют смысла. Но в случае с сотрудниками такая ситуация однако могла бы иметь смысл – вполне возможно, что какой-либо сотрудник в данный момент не зачислен вообще ни в какой отдел. Заметим, что ответ на данный вопрос не зависит от прихоти проектировщика базы данных, а определяется фактическим образом действий, принятым в той части реального мира, которая должна быть представлена в рассматриваемой базе данных. Подобные замечания имеют отношение и к вопросам, обсуждаемым ниже.

2. Что должно случиться при попытке УДАЛЕНИЯ целевой сущности, на которую ссылается внешний ключ? Например, при удалении поставщика, который осуществил по крайней мере одну поставку. Существует три возможности:

- КАСКАДИРУЕТСЯ – операция удаления "каскадируется" с тем, чтобы удалить также поставки этого поставщика.
- ОГРАНИЧИВАЕТСЯ - удаляются лишь те поставщики, которые еще не осуществляли поставок. Иначе операция удаления отвергается.
- УСТАНОВЛИВАЕТСЯ - для всех поставок удаляемого поставщика NULL-значение внешний ключ устанавливается в неопределенное значение, а затем этот поставщик удаляется. Такая возможность, конечно, неприменима, если данный внешний ключ не должен содержать NULL-значений.

3. Что должно происходить при попытке ОБНОВЛЕНИЯ первичного ключа целевой сущности, на которую ссылается некоторый внешний ключ? Например, может быть предпринята попытка обновить номер такого поставщика, для которого имеется, по крайней мере, одна соответствующая поставка. Этот случай для определенности снова рассмотрим подробнее. Имеются те же три возможности, как и при удалении:

- КАСКАДИРУЕТСЯ - операция обновления "каскадируется" с тем, чтобы обновить также и внешний ключ в поставках этого поставщика.
- ОГРАНИЧИВАЕТСЯ - обновляются первичные ключи лишь тех поставщиков, которые еще не осуществляли поставок. Иначе операция обновления отвергается.
- УСТАНОВЛИВАЕТСЯ - для всех поставок такого поставщика

NULL-значение внешний ключ устанавливается в неопределенное значение, а затем обновляется первичный ключ поставщика. Такая возможность, конечно, неприменима, если данный внешний ключ не должен содержать NULL-значений.

Таким образом, для каждого внешнего ключа в проекте вы должны специфицировать не только поле или комбинацию полей, составляющих этот внешний ключ, и целевую таблицу, которая идентифицируется этим ключом, но также и ответы на указанные выше вопроса (три ограничения, которые относятся к этому внешнему ключу).

Ограничения целостности

Целостность (от англ. integrity – нетронутость, неприкосновенность, сохранность, целостность) – понимается как правильность данных в любой момент времени. Но эта цель может быть достигнута лишь в определенных пределах: СУБД не может контролировать правильность каждого отдельного значения, вводимого в базу данных (хотя каждое значение можно проверить на правдоподобность). Например, нельзя обнаружить, что вводимое значение 5 (представляющее номер дня недели) в действительности должно быть равно 3. С другой стороны, значение 9 явно будет ошибочным и СУБД должна его отвергнуть. Однако для этого ей следует сообщить, что номера дней недели должны принадлежать набору (1,2,3,4,5,6,7).

Поддержание целостности базы данных может рассматриваться как защита данных от неверных изменений или разрушения. Современные СУБД имеют ряд средств для обеспечения поддержания целостности.

Выделяют три группы правил целостности:

1. Целостность по сущностям - не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе, принимал неопределенное значение.
2. Целостность по ссылкам - значение внешнего ключа должно либо:
 - быть равным значению первичного ключа цели;
 - быть полностью неопределенным, т.е. каждое значение атрибута, участвующего во внешнем ключе должно быть неопределенным.
3. Целостность, определяемая пользователем - для любой конкретной базы данных существует ряд дополнительных специфических правил, которые относятся к ней одной и определяются разработчиком. Чаще всего контролируется:
 - уникальность тех или иных атрибутов;
 - диапазон значений (экзаменационная оценка от 2 до 5);
 - принадлежность набору значений (пол "М" или "Ж").

Построение инфологической модели

Для определения перечня и структуры хранимых данных надо собрать информацию о реальных и потенциальных приложениях, а также о

пользователях базы данных, а при построении инфологической модели следует заботиться лишь о надежности хранения этих данных, совершенно забывая о приложениях и пользователях, для которых создается база данных.

Это связано с абсолютно различающимися требованиями к базе данных прикладных программ и администрирования базы данных. В первом случае, например, проще иметь в одном месте (в одной таблице) все данные, необходимые им для реализации запроса из прикладной программы. Во втором же, необходимо заботится об исключении возможных искажений хранимых данных при вводе в базу данных новой информации и обновлении или удалении существующей. Так как многолетний мировой опыт использования информационных систем, построенных на основе баз данных, показывает, что недостатки проекта невозможно устранить любыми ухищрениями в программах приложений, то опытные проектировщики не позволяют себе идти на поводу у прикладных программ, в ущерб обеспечению целостности данных.

Поэтому приведем несколько советов:

1. четко разграничивать такие понятия как запрос на данные и ведение данных (ввод, изменение и удаление);
2. помнить, что, как правило, база данных является информационной основой не одного, а нескольких приложений, часть их которых появится в будущем;
3. плохой проект базы данных не может быть исправлен с помощью любых (даже самых изошренных) приложений.

Реляционная структура данных

В конце 60-х годов появились работы, в которых обсуждались возможности применения различных табличных даталогических моделей данных, т.е. возможности использования привычных и естественных способов представления данных. Наиболее значительной из них была статья сотрудника фирмы IBM д-ра Э.Кодда (Codd E.F., A Relational Model of Data for Large Shared Data Banks. SACM 13: 6, June 1970), где, вероятно, впервые был применен термин "реляционная модель данных".

Будучи математиком по образованию Э.Кодд предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность, декартово произведение). Он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как отношение – relation (англ.).

Наименьшая единица данных реляционной модели – это отдельное атомарное (неразложимое) для данной модели значение данных. Так, в одной предметной области фамилия, имя и отчество могут рассматриваться как единое значение, а в другой – как три различных значения.

Доменом называется множество атомарных значений одного и того же типа. Например - домен пунктов отправления (назначения) – множество названий населенных пунктов, а домен номеров рейса – множество целых положительных чисел.

Смысл доменов состоит в следующем. Если значения двух атрибутов берутся из одного и того же домена, то, вероятно, имеют смысл сравнения, использующие эти два атрибута (например, для организации транзитного рейса можно дать запрос "Выдать рейсы, в которых время вылета из Москвы в Сочи больше времени прибытия из Архангельска в Москву"). Если же значения двух атрибутов берутся из различных доменов, то их сравнение, вероятно, лишено смысла: не имеет смысла сравнивать номер рейса со стоимостью билета.

Отношение на доменах D_1, D_2, \dots, D_n (не обязательно, чтобы все они были различны) состоит из заголовка и тела.

Заголовок (он называется интерпретацией) состоит из такого фиксированного множества атрибутов A_1, A_2, \dots, A_n , что существует взаимно однозначное соответствие между этими атрибутами A_i и определяющими их доменами D_i ($i=1,2,\dots,n$).

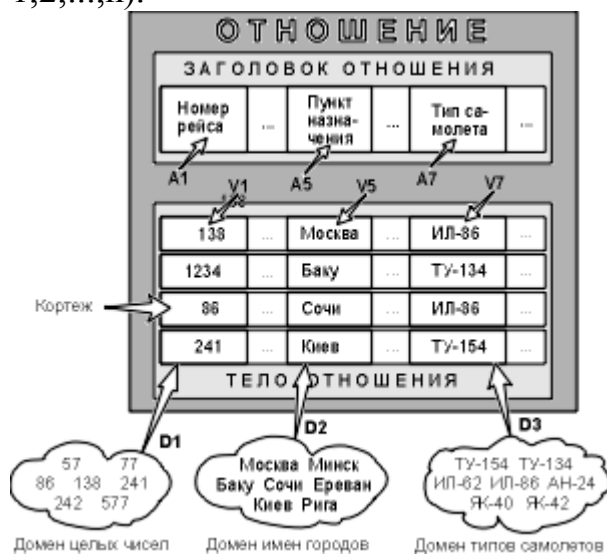


Рис. 3.4. Отношение с математической точки зрения (A_i - атрибуты, V_i - значения атрибутов)

Тело состоит из меняющегося во времени множества кортежей, где каждый кортеж состоит в свою очередь из множества пар атрибут-значение ($A_i:V_i$), ($i=1,2,\dots,n$), по одной такой паре для каждого атрибута A_i в заголовке. Для любой заданной пары атрибут-значение ($A_i:V_i$) V_i является значением из единственного домена D_i , который связан с атрибутом A_i .

Степень отношения – это число его атрибутов. Отношение степени один называют унарным, степени два – бинарным, степени три – тернарным, ..., а степени n – n -арным.

Кардинальное число или *мощность отношения* – это число его кор-

тежей. Кардинальное число отношения изменяется во времени в отличие от его степени.

Поскольку отношение – это множество, а множества по определению не содержат совпадающих элементов, то никакие два кортежа отношения не могут быть дубликатами друг друга в любой произвольно-заданный момент времени. Пусть R – отношение с атрибутами A_1, A_2, \dots, A_n . Говорят, что множество атрибутов $K=(A_i, A_j, \dots, A_k)$ отношения R является возможным ключом R тогда и только тогда, когда удовлетворяются два независимых от времени условия:

1. Уникальность: в произвольный заданный момент времени никакие два различных кортежа R не имеют одного и того же значения для A_i, A_j, \dots, A_k .
2. Минимальность: ни один из атрибутов A_i, A_j, \dots, A_k не может быть исключен из K без нарушения уникальности.

Каждое отношение обладает хотя бы одним возможным ключом, поскольку, по меньшей мере, комбинация всех его атрибутов удовлетворяет условию уникальности. Один из возможных ключей (выбранный произвольным образом) принимается за его первичный ключ. Остальные возможные ключи, если они есть, называются альтернативными ключами.

Вышеупомянутые и некоторые другие математические понятия явились теоретической базой для создания реляционных СУБД, разработки соответствующих языковых средств и программных систем, обеспечивающих их высокую производительность, и создания основ теории проектирования баз данных. Однако для массового пользователя реляционных СУБД можно с успехом использовать неформальные эквиваленты этих понятий:

Отношение – Таблица (иногда Файл),

Кортеж – Строка (иногда Запись),

Атрибут – Столбец, Поле.

При этом принимается, что "запись" означает "экземпляр записи", а "поле" означает "имя и тип поля".

Реляционная база данных

Реляционная база данных – это совокупность отношений, содержащих всю информацию, которая должна храниться в БД. Однако пользователи могут воспринимать такую базу данных как совокупность таблиц.

При этом:

1. Каждая таблица состоит из однотипных строк и имеет уникальное имя.
2. Строки имеют фиксированное число полей (столбцов) и значений (множественные поля и повторяющиеся группы недопустимы). Иначе говоря, в каждой позиции таблицы на пересечении строки и столбца

всегда имеется в точности одно значение или ничего.

3. Строки таблицы обязательно отличаются друг от друга хотя бы единственным значением, что позволяет однозначно идентифицировать любую строку такой таблицы.
4. Столбцам таблицы однозначно присваиваются имена, и в каждом из них размещаются однородные значения данных (даты, фамилии, целые числа или денежные суммы).
5. Полное информационное содержание базы данных представляется в виде явных значений данных и такой метод представления является единственным. В частности, не существует каких-либо специальных "связей" или указателей, соединяющих одну таблицу с другой. Так, связи между строкой с БЛ = 2 таблицы "Блюда" на рис. 3.2 и строкой с ПР = 7 таблицы продукты (для приготовления Харчо нужен Рис), представляется не с помощью указателей, а благодаря существованию в таблице "Состав" строки, в которой номер блюда равен 2, а номер продукта – 7.
6. При выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке безотносительно к их информационному содержанию. Этому способствует наличие имен таблиц и их столбцов, а также возможность выделения любой их строки или любого набора строк с указанными признаками (например, рейсов с пунктом назначения "Париж" и временем прибытия до 12 часов).

Манипулирование реляционными данными

Предложив реляционную модель данных, Э.Ф. Кодд создал и инструмент для удобной работы с отношениями – реляционную алгебру. Каждая операция этой алгебры использует одну или несколько таблиц (отношений) в качестве ее операндов и продуцирует в результате новую таблицу, т.е. позволяет "разрезать" или "склеивать" таблицы (рис. 3.5).

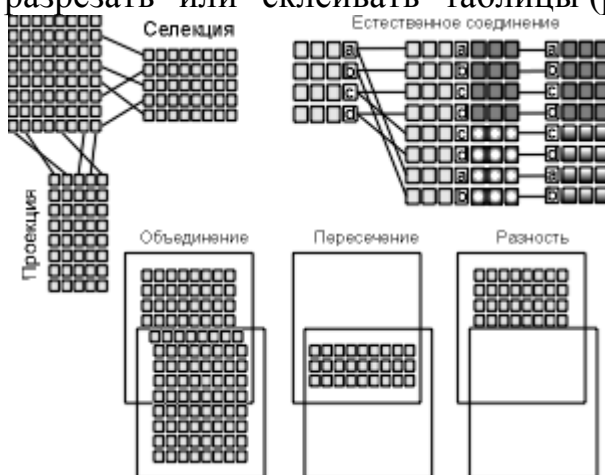


Рис. 3.5. Некоторые операции реляционной алгебры

Созданы языки манипулирования данными, позволяющие реализовать все операции реляционной алгебры и практически любые их сочетания. Среди них наиболее распространены SQL (Structured Query Language – *структурированный язык запросов*) и QBE (Query-By-Example – *запросы по образцу*). Оба относятся к языкам очень высокого уровня, с помощью которых пользователь указывает, какие данные необходимо получить, не уточняя процедуру их получения.

С помощью единственного запроса на любом из этих языков можно соединить несколько таблиц во временную таблицу и вырезать из нее требуемые строки и столбцы (селекция и проекция).

Проектирование реляционных баз данных, цели проектирования

Только небольшие организации могут обобществить данные в одной полностью интегрированной базе данных. Чаще всего практически не возможно охватить и осмыслить все информационные требования сотрудников организации (т.е. будущих пользователей системы). Поэтому информационные системы больших организаций содержат несколько десятков БД, нередко распределенных между несколькими взаимосвязанными ЭВМ различных подразделений. (Так в больших городах создается не одна, а несколько овощных баз, расположенных в разных районах.)

Отдельные БД могут объединять все данные, необходимые для решения одной или нескольких прикладных задач, или данные, относящиеся к какой-либо предметной области (например, финансам, студентам, преподавателям, кулинарии и т.п.). Первые обычно называют *прикладными БД*, а вторые – *предметными БД* (соотносящимся с предметами организации, а не с ее информационными приложениями). (Первые можно сравнить с базами материально-технического снабжения или отдыха, а вторые – с овощными и вещевыми базами.)

Предметные БД позволяют обеспечить поддержку любых текущих и будущих приложений, поскольку набор их элементов данных включает в себя наборы элементов данных прикладных БД. Вследствие этого предметные БД создают основу для обработки неформализованных, изменяющихся и неизвестных запросов и приложений (приложений, для которых невозможно заранее определить требования к данным). Такая гибкость и приспособляемость позволяет создавать на основе предметных БД достаточно стабильные информационные системы, т.е. системы, в которых большинство изменений можно осуществить без вынужденного переписывания старых приложений.

Основывая же проектирование БД на текущих и предвидимых приложениях, можно существенно ускорить создание высокоэффективной информационной системы, т.е. системы, структура которой учитывает наиболее часто встречающиеся пути доступа к данным. Поэтому приклад-

ное проектирование до сих пор привлекает некоторых разработчиков. Однако по мере роста числа приложений таких информационных систем быстро увеличивается число прикладных БД, резко возрастает уровень дублирования данных и повышается стоимость их ведения.

Таким образом, каждый из рассмотренных подходов к проектированию воздействует на результаты проектирования в разных направлениях. Желание достичь и гибкости, и эффективности привело к формированию методологии проектирования, использующей как предметный, так и прикладной подходы. В общем случае предметный подход используется для построения первоначальной информационной структуры, а прикладной – для ее совершенствования с целью повышения эффективности обработки данных.

При проектировании информационной системы необходимо провести анализ целей этой системы и выявить требования к ней отдельных пользователей (сотрудников организации). Сбор данных начинается с изучения сущностей организации и процессов, использующих эти сущности. Сущности группируются по "сходству" (частоте их использования для выполнения тех или иных действий) и по количеству ассоциативных связей между ними (самолет – пассажир, преподаватель – дисциплина, студент – сессия и т.д.). Сущности или группы сущностей, обладающие наибольшим сходством и (или) с наибольшей частотой ассоциативных связей объединяются в предметные БД. (Нередко сущности объединяются в предметные БД без использования формальных методик – по "здравому смыслу").

Далее будут рассматриваться вопросы, связанные с проектированием отдельных реляционных предметных БД.

Основная цель проектирования БД – это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте. Так называемый, "чистый" проект БД ("Каждый факт в одном месте") можно создать, используя методологию нормализации отношений.

Нормализация – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении данных. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

Каждая таблица в реляционной БД удовлетворяет условию, в соответствии с которым в позиции на пересечении каждой строки и столбца

таблицы всегда находится единственное атомарное значение, и никогда не может быть множества таких значений. Любая таблица, удовлетворяющая этому условию, называется нормализованной. Фактически, ненормализованные таблицы, т.е. таблицы, содержащие повторяющиеся группы, даже не допускаются в реляционной БД.

Всякая нормализованная таблица автоматически считается таблицей в *первой нормальной форме*, сокращенно *1НФ*. Таким образом, строго говоря, "нормализованная" и "находящаяся в 1НФ" означают одно и то же. Однако на практике термин "нормализованная" часто используется в более узком смысле – "полностью нормализованная", который означает, что в проекте не нарушаются никакие принципы нормализации.

Теперь в дополнение к 1НФ можно определить дальнейшие уровни нормализации – *вторую нормальную форму (2НФ)*, *третью нормальную форму (3НФ)* и т.д. По существу, таблица находится в 2НФ, если она находится в 1НФ и удовлетворяет, кроме того, некоторому дополнительному условию, суть которого будет рассмотрена ниже. Таблица находится в 3НФ, если она находится в 2НФ и, помимо этого, удовлетворяет еще другому дополнительному условию и т.д.

Таким образом, каждая нормальная форма является в некотором смысле более ограниченной, но и более *желательной*, чем предшествующая. Это связано с тем, что "(N+1)-я нормальная форма" не обладает некоторыми непривлекательными особенностями, свойственным "N-й нормальной форме". Общий смысл дополнительного условия, налагаемого на (N+1)-ю нормальную форму по отношению к N-й нормальной форме, состоит в исключении этих непривлекательных особенностей.

Теория нормализации основывается на наличии той или иной зависимости между полями таблицы. Определены два вида таких зависимостей: *функциональные* и *многозначные*.

Функциональная зависимость. Поле В таблицы функционально зависит от поля А той же таблицы в том и только в том случае, когда в любой заданный момент времени для каждого из различных значений поля А обязательно существует только одно из различных значений поля В. Отметим, что здесь допускается, что поля А и В могут быть составными.

Полная функциональная зависимость. Поле В находится в полной функциональной зависимости от составного поля А, если оно функционально зависит от А и не зависит функционально от любого подмножества поля А.

Многозначная зависимость. Поле А многозначно определяет поле В той же таблицы, если для каждого значения поля А существует хорошо определенное множество соответствующих значений В.

Нормальные формы

Ранее было дано определение первой нормальной формы (1НФ). Приведем здесь более строгое её определение, а также определения других нормальных форм.

Таблица находится в первой нормальной форме (1НФ) тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица находится во второй нормальной форме (2НФ), если она удовлетворяет определению 1НФ и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом.

Для упрощения нормализации подобных таблиц целесообразно использовать следующую рекомендацию. *При проведении нормализации таблиц, в которые введены цифровые (или другие) заменители составных и (или) текстовых первичных и внешних ключей, следует хотя бы мысленно подменять их на исходные ключи, а после окончания нормализации снова восстанавливать.*

Таблица находится в третьей нормальной форме (3НФ), если она удовлетворяет определению 2НФ и не одно из ее не ключевых полей не зависит функционально от любого другого не ключевого поля.

Таблица находится в *нормальной форме Бойса-Кодда (НФБК)*, если и только если любая функциональная зависимость между его полями сводится к полной функциональной зависимости от *возможного* ключа.

В следующих нормальных формах (4НФ и 5НФ) учитываются не только функциональные, но и многозначные зависимости между полями таблицы. Для их описания познакомимся с понятием полной декомпозиции таблицы.

Полной декомпозицией таблицы называют такую совокупность произвольного числа ее проекций, соединение которых полностью совпадает с содержимым таблицы.

Таблица находится в *пятой нормальной форме (5НФ)* тогда и только тогда, когда в каждой ее полной декомпозиции все проекции содержат возможный ключ. Таблица, не имеющая ни одной полной декомпозиции, также находится в 5НФ.

Четвертая нормальная форма (4НФ) является частным случаем 5НФ, когда полная декомпозиция должна быть соединением ровно двух проекций. Весьма не просто подобрать реальную таблицу, которая находилась бы в 4НФ, но не была бы в 5НФ.

Как уже говорилось, нормализация – это разбиение таблицы на несколько, обладающих лучшими свойствами при обновлении, включении и удалении данных. Теперь можно дать и другое определение: нормализация

– это процесс последовательной замены таблицы ее полными декомпозициями до тех пор, пока все они не будут находиться в 5НФ. На практике же достаточно привести таблицы к НФБК и с большой гарантией считать, что они находятся в 5НФ. Разумеется, этот факт нуждается в проверке, однако пока не существует эффективного алгоритма такой проверки. Поэтому остановимся лишь на процедуре приведения таблиц к НФБК.

Эта процедура основывается на том, что единственными функциональными зависимостями в любой таблице должны быть зависимости вида $K \rightarrow F$, где K – первичный ключ, а F – некоторое другое поле. Заметим, что это следует из определения первичного ключа таблицы, в соответствии с которым $K \rightarrow F$ всегда имеет место для всех полей данной таблицы. "Один факт в одном месте" говорит о том, что не имеют силы никакие другие функциональные зависимости. Цель нормализации состоит именно в том, чтобы избавиться от всех этих "других" функциональных зависимостей, т.е. таких, которые имеют иной вид, чем $K \rightarrow F$.

Если воспользоваться приведенной ранее рекомендацией и подменить на время нормализации коды первичных (внешних) ключей на исходные ключи, то, по существу, следует рассмотреть лишь два случая:

1. Таблица имеет составной первичный ключ вида, скажем, $(K1, K2)$, и включает также поле F , которое функционально зависит от части этого ключа, например, от $K2$, но не от полного ключа. В этом случае рекомендуется сформировать другую таблицу, содержащую $K2$ и F (первичный ключ – $K2$), и удалить F из первоначальной таблицы:

Заменить $T(K1, K2, F)$, первичный ключ $(K1, K2)$, ФЗ $K2 \rightarrow F$
на $T1(K1, K2)$, первичный ключ $(K1, K2)$,
и $T2(K2, F)$, первичный ключ $K2$.

2. Таблица имеет первичный (возможный) ключ K , не являющееся возможным ключом поле $F1$, которое, конечно, функционально зависит от K , и другое не ключевое поле $F2$, которое функционально зависит от $F1$. Решение здесь, по существу, то же самое, что и прежде – формируется другая таблица, содержащая $F1$ и $F2$, с первичным ключом $F1$, и $F2$ удаляется из первоначальной таблицы:

Заменить $T(K, F1, F2)$, первичный ключ K , ФЗ $F1 \rightarrow F2$
на $T1(K, F1)$, первичный ключ K ,
и $T2(F1, F2)$, первичный ключ $F1$.

Для любой заданной таблицы, повторяя применение двух рассмотренных правил, почти во всех практических ситуациях можно получить в конечном счете множество таблиц, которые находятся в "окончательной" нормальной форме и, таким образом, не содержат каких-либо функциональных зависимостей вида, отличного от $K \rightarrow F$.

Для выполнения этих операций необходимо первоначально иметь в

качестве входных данных какие-либо "большие" таблицы (например, универсальные отношения). Но нормализация ничего не говорит о том, как получить эти большие таблицы. Процедура проектирования как раз и дает возможность формирования исходных таблиц.

Процедура проектирования

Процесс проектирования информационных систем является достаточно сложной задачей. Он начинается с построения инфологической модели данных, т.е. идентификации сущностей. Затем необходимо выполнить следующие шаги процедуры проектирования даталогической модели.

1. Представить каждый стержень (независимую сущность) таблицей базы данных (базовой таблицей) и специфицировать первичный ключ этой базовой таблицы.

2. Представить каждую ассоциацию (связь вида "многие-ко-многим" или "многие-ко-многим-ко-многим" и т.д. между сущностями) как базовую таблицу. Использовать в этой таблице внешние ключи для идентификации участников ассоциации и специфицировать ограничения, связанные с каждым из этих внешних ключей.

3. Представить каждую характеристику как базовую таблицу с внешним ключом, идентифицирующим сущность, описываемую этой характеристикой. Специфицировать ограничения на внешний ключ этой таблицы и ее первичный ключ – по всей вероятности, комбинации этого внешнего ключа и свойства, которое гарантирует "уникальность в рамках описываемой сущности".

4. Представить каждое обозначение, которое не рассматривалось в предыдущем пункте, как базовую таблицу с внешним ключом, идентифицирующим обозначаемую сущность. Специфицировать связанные с каждым таким внешним ключом ограничения.

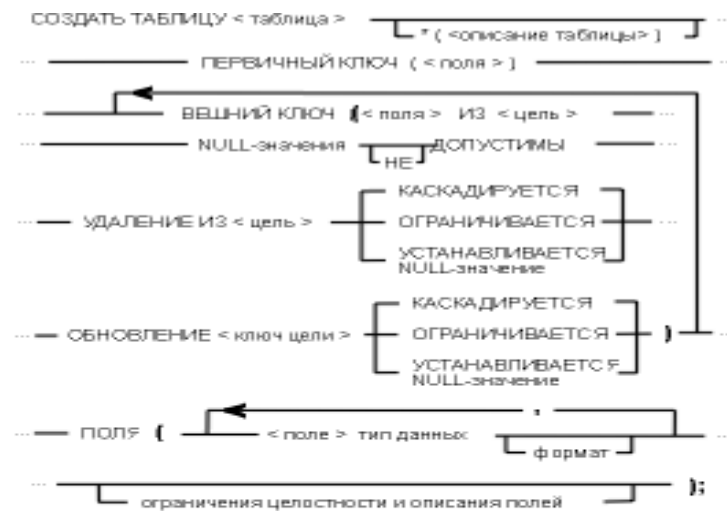
5. Представить каждое свойство как поле в базовой таблице, представляющей сущность, которая непосредственно описывается этим свойством.

6. Для того чтобы исключить в проекте непреднамеренные нарушения каких-либо принципов нормализации, выполнить описанную в п. 4.6 процедуру нормализации.

7. Если в процессе нормализации было произведено разделение каких-либо таблиц, то следует модифицировать инфологическую модель базы данных и повторить перечисленные шаги.

8. Указать ограничения целостности проектируемой базы данных и дать (если это необходимо) краткое описание полученных таблиц и их полей.

На рис. 3.6 показан синтаксис предложения, предлагаемого для регистрации принимаемых проектных решений.



В этом описании:	
"таблица"	– имя проектируемой базовой таблицы;
"описание таблицы"	– любой текст, характеризующий назначение и (или) содержание таблицы;
"поля"	– список имен полей, разделенных запятыми;
"цель"	– имя таблицы, первичный ключ которой используется как внешний ключ проектируемой;
"ключ цели"	– специфицирует "первичный ключ" цели;
"поле"	– имя поля проектируемой таблицы (включая и имена внешних ключей).

Рис.3.6. Синтаксис описания проектных решений

Рассмотренный язык описания данных, основанный на языке SQL, позволяет дать удобное и полное описание любой сущности и, следовательно, всей базы данных. Однако такое описание, как и любое подробное описание, не отличается наглядностью. Для достижения большей иллюстративности целесообразно дополнять проект инфологической моделью.

Для наиболее распространенных реляционных баз данных можно предложить язык инфологического моделирования "Таблица-связь". В нем все сущности изображаются одностолбцовыми таблицами с заголовками, состоящими из имени и типа сущности. Строки таблицы – это перечень атрибутов сущности, а те из них, которые составляют первичный ключ, располагаются рядом и обводятся рамкой. Связи между сущностями указываются стрелками, направленными от первичных ключей или их составляющих.

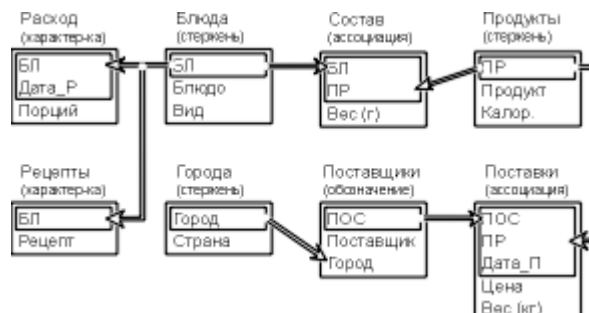


Рис. 3.7. Инфологическая модель базы данных "Питание", построенная с помощью языка "Таблицы-связи"

Будьте очень внимательны с неопределенными (NULL) значениями. В поведении неопределенных значений проявляется много произвола и противоречивости. В разных СУБД при выполнении различных операций (сравнение, объединение, сортировка, группирование и другие) два неопределенных значения могут быть или не быть равными друг другу. Они могут по разному влиять на результат выполнения операций по определению средних значений и нахождения количества значений. Для исключения ошибок в ряде СУБД существует возможность замены NULL-значения нулем при выполнении расчетов, объявление всех NULL-значений равными друг другу и т.п.

Пример проектирования базы данных

База данных предназначена для хранения данных о приобретенных библиотекой изданиях (монографиях, справочниках, сборниках статей и т.п.), информации о местонахождении отдельных экземпляров (переплетов) каждого издания и сведений о читателях, рис.3.8

Д27 Дейт К. Руководство по реляционной СУБД DB2 / Пер. с англ. и предисл. М.Р.Когаловского. – М.: Финансы и статистика, 1988. – 320 с.: ил.

ISBN 5-279-00063-9

Книга американского специалиста в области реляционных баз данных К.Дейта, автора популярной в СССР монографии "Введение в системы баз данных" (М.: Наука, 1981), представляет собой руководство по перспективной СУБД фирмы ИБМ DB2, сочетающей возможности широко известной системы IMS/VS и реляционной СУБД.

Для специалистов по программному обеспечению информационных систем и студентов вузов.

ББК 32.973

Рис.3.8. Макет аннотированной каталожной карточки

Для ведения библиотечных каталогов, организации поиска требуемых изданий и библиотечной статистики в базе должны храниться сведения, большая часть которых размещаются в аннотированных каталожных карточках (рис. 3.8). Анализ запросов на литературу (как читателями, так и сотрудниками библиотек) показывает, что для поиска подходящих изданий (по тематике, автору, художнику, издательству и т.п.) и отбора нужного (например, по аннотации) следует выделить следующие атрибуты каталожной карточки:

1. Автор (фамилия и имена (инициалы) или псевдоним каждого автора издания).

2. Название (заглавие) издания.
3. Номер тома (части, книги, выпуска).
4. Вид издания (сборник, справочник, монография, ...).
5. Составитель (фамилия и имена (инициалы) каждого из составителей издания).
6. Язык, с которого выполнен перевод издания.
7. Переводчик (фамилия и инициалы каждого переводчика).
8. Под чьей редакцией (фамилия и имена (инициалы) каждого из титульных редакторов).
9. Художник (фамилия и имена (инициалы) каждого художника-иллюстратора) - для художественных изданий, иллюстрируемых оригинальными рисунками.
10. Повторность издания (второе, одиннадцатое и т.п.).
11. Характер переиздания (исправленное, дополненное, переработанное, стереотипное и т.п.).
12. Место издания (город).
13. Издательство (название издательства).
14. Год выпуска издания.
15. Издательская аннотация или реферат.
16. Библиотечный шифр (например, ББК 32.973).
17. Авторский знак (например, Д27).

Библиотечный шифр и авторский знак используются при составлении каталогов и организации расстановки изданий на полках: по содержанию (в соответствии с библиотечным шифром) и алфавиту (в соответствии с авторским знаком).

Библиотечно-библиографическая классификация (ББК) распределяет издания по отраслям знания в соответствии с их содержанием. В ней используется цифро-буквенные индексы ступенчатой структуры.

Каждый из девяти классов (1. Марксизм-ленинизм; 2. Естественные науки; 3. Техника. Технические науки; 4. Сельское и лесное хозяйство; 5. Здравоохранение; 6/8. Общественные и гуманитарные науки; 9. Библиографические пособия. Справочные издания. Журналы.) делится на подклассы и следующие ступени деления:

Техника. Технические науки.

32 Радиоэлектроника.

32.97 Вычислительная техника.

32.973 Электронные вычислительные машины и устройства.

32.973.2 Электронно-вычислительные машины и устройства дискретного действия.

Шифр ББК используется при выделении хранимым изданиям определенных комнат, стеллажей и полок, а также для составления каталогов и статистических отчетов.

Авторский знак, состоящий из первой буквы фамилии (псевдонима) автора или названия издания (для изданий без автора) и числа, соответствующего слогу, наиболее приближающегося по написанию к первым буквам фамилии (названия), упрощает расстановку книг на полках в алфавитном порядке.

К объектам и атрибутам, позволяющим охарактеризовать отдельные экземпляры изданий (переплеты), места их хранения и читателей, можно отнести:

18. Номер комнаты (помещения для хранения переплетов).
19. Номер стеллажа в комнате.
20. Номер полки на стеллаже.
21. Номер (инвентарный номер) переплета.
22. Дата приобретения конкретного переплета.
23. Цена конкретного переплета.
24. Дата размещения конкретного переплета на конкретном месте.
25. Дата изъятия переплета с установленного места.
26. Номер читательского билета (формуляра).
27. Фамилия читателя.
28. Имя читателя.
29. Отчество читателя.
30. Адрес читателя.
31. Телефон читателя.
32. Дата выдачи читателю конкретного переплета.
33. Срок, на который конкретный переплет выдан читателю.
34. Дата возврата переплета.

Построение инфологической модели

Анализ определенных выше объектов и атрибутов позволяет выделить сущности проектируемой базы данных и, приняв решение о создании реляционной базы данных, построить ее инфологическую модель на языке "Таблицы-связи".

К стержневым сущностям можно отнести:

1. Создатели (Код создателя, Создатель).

Эта сущность отводится для хранения сведений об основных людях, принимавших участие в подготовке рукописи издания (авторах, составителях, титульных редакторах, переводчиках и художниках). Такое объединение допустимо, так как данные о разных создателях выбираются из одного домена (фамилия и имена) и исключает дублирование данных (один и тот же человек может играть разные роли в подготовке разных изданий). Например, С.Я.Маршак писал стихи (Сказка о глупом мышонке) и пьесы (Двенадцать месяцев), переводил Дж.Байрона, Р.Бернса, Г.Гейне и составлял сборники стихов.

Так как фамилия и имена (инициалы) создателя могут быть достаточно громоздкими (М.Е. Салтыков-Щедрин, Франсуа Рене де Шатобриан, Остен Жюль Жан-Батист Ипполит и т.п.) и будут многократно встречаться в разных изданиях, то их целесообразно нумеровать и ссылаться на эти номера. Для этого вводится целочисленный атрибут "Код_создателя", который будет автоматически наращиваться на единицу при вводе в базу данных нового автора, переводчика или другого создателя.

Аналогично создаются: Код_издательства, Код_заглавия, Вид_издания, Код_характера, Код_языка, Номер_билета, Номер_переплета, Код_места и Код_издания, замещающие от одного до девяти атрибутов.

2. Издательства (Код_издательства, Название, Город).
3. Заглавия (Код_заглавия, Заглавие).

Выделение этой сущности позволит сократить объем данных и снизить вероятность возникновения противоречивости (исключается необходимость ввода длинных текстовых названий для различных томов собраний сочинений, повторных изданий, учебников и т.п.).

4. Вид_издания (Вид_издания, Название_вида).
5. Характеры (Код_характера, Характер_переиздания).
6. Языки (Код_языка, Язык, Сокращение).

Кроме названия языка хранится его общепринятое сокращение (англ., исп., нем., фр.), если оно существует.

7. Места (Код_места, Номер_комнаты, Номер_стеллажа, Номер_полки).

Один из кодов этой сущности (например, "-1") отведен для описания обобщенного места, находящегося за стенами хранилища книг (издание выдано читателю, временно передано другой библиотеке или организации).

8. Читатели (Номер_билета, Фамилия, Имя, Отчество, Адрес, Телефон).

Две ключевые сущности, описывающие издание и его конкретные экземпляры, оказываются зависимыми от других сущностей и попадают в класс обозначений:

1. Издание (Код_издания, Код_заглавия, Вид_издания, Номер_тома, Авторский_знак, Библиотечн_шифр, Повторность, Код_издательства, Год_издания, Аннотация) [Заглавия, Вид_издания, Издательства];
2. Переплеты (Номер_переплета, Код_издания, Цена, Дата_приобретения)[Издания];

Стержневые сущности и обозначения связаны между собой ассоциациями:

1. Авторы [Создатели М, Издание N] (Код_создателя, Код_издания).
2. Составители [Создатели М, Издания N] (Код_создателя, Код_издания).
3. Редакторы [Создатели М, Издания N] (Код_создателя, Код_издания).
4. Художники [Создатели М, Издания N] (Код_создателя, Код_издания).
5. Переводчики [Создатели М, Издания N] (Код_создателя, Код_издания, Язык).
6. Переиздания [Характеры М, Издания N] (Код_характера, Код_издания).
7. Размещение [Места М, Переплеты N] (Код_места, Номер_переплета, Дата_размещения, Дата_изъятия).
8. Выдача [Читатели М, Переплеты N] (Номер_билета, Номер_переплета, Дата_выдачи, Срок, Дата_возврата).

И, наконец, для уменьшения объема часто используемого обозначения "Издания" из него выделена характеристика:

1. Аннотации (Код_издания, Аннотация) {Издание}.

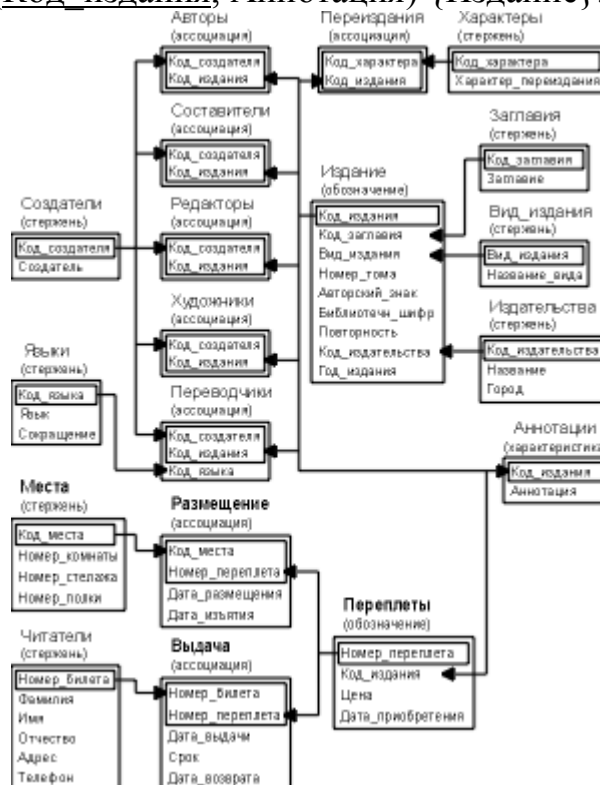


Рис. 3.9. Инфологическая модель базы данных "Библиотека", построенная с помощью языка "Таблицы-связи"

Проектирование базы данных

В соответствии с процедурой проектирования каждая из полученных сущностей должна быть представлена базовой таблицей. Первый вариант этих таблиц описывается так:

СОЗДАТЬ ТАБЛИЦУ Создатели *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Код_создат)

ПОЛЯ (Код_создат Целое, Фам_ИО Текст 30);

СОЗДАТЬ ТАБЛИЦУ Издательства *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Код_издательства)

ПОЛЯ (Код_издательства Целое, Название
Текст 40, Город Текст 25);

СОЗДАТЬ ТАБЛИЦУ Заглавия *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Код_заглавия)

ПОЛЯ (Код_заглавия Целое, Заглавие Запись);

СОЗДАТЬ ТАБЛИЦУ Вид_издания *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Вид_издания)

ПОЛЯ (Вид_издания Целое, Название_вида Текст 16);

СОЗДАТЬ ТАБЛИЦУ Характеры *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Код_характера)

ПОЛЯ (Код_характера Целое, Характер_переиздания
Текст 16);

СОЗДАТЬ ТАБЛИЦУ Языки *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Код_языка)

ПОЛЯ (Код_языка Целое, Язык Текст 16, Сокращение
Текст 6);

СОЗДАТЬ ТАБЛИЦУ Места *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Код_места)

ПОЛЯ (Код_места Целое, Номер_комнаты Целое,
Номер_стелажа Целое, Номер_полки Целое);

СОЗДАТЬ ТАБЛИЦУ Читатели *(Стержневая сущность)

ПЕРВИЧНЫЙ КЛЮЧ (Ном_билета)

ПОЛЯ (Ном_билета Целое, Фамилия Текст 20, Имя Текст 16,
Отчество Текст 20, Адрес Текст 60, Телефон Текст 9);

СОЗДАТЬ ТАБЛИЦУ Издание *(Обозначение)

ПЕРВИЧНЫЙ КЛЮЧ (Код_издания)

ВНЕШНИЙ КЛЮЧ (Код_заглавия ИЗ Заглавия

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Заглавия ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Заглавия.Код_заглавия ОГРАНИЧИВАЕТСЯ)

ВНЕШНИЙ КЛЮЧ (Вид_издания ИЗ Вид_издания

NULL-значения ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Вид_издания ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Вид_издания.Вид_издания КАСКАДИРУЕТСЯ)

ВНЕШНИЙ КЛЮЧ (Код_издательства ИЗ Издательства

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Издательства ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Издательства.Код_издательства КАСКАДИРУЕТСЯ)

ПОЛЯ (Код_издания Целое, Код_заглавия Целое,

Вид_издания Текст 16, Номер_тома Целое,

Авторский_знак Текст 3, Библиотечн_шифр Текст 12,

Повторность Целое, Код_издательст- ва Целое,

Год_издания Целое)

ОГРАНИЧЕНИЯ (1. Значения полей Код_заглавия, Вид_издания и Код_издательства должны принадлежать набору значений соответствующих полей таблиц Заглавия, Вид_издания и Издательства; при нарушении вывод сообщения "Такого заглавия нет", "Такого вида издания нет" или "Такого издательства нет".);

СОЗДАТЬ ТАБЛИЦУ Переплеты *(Обозначение)

ПЕРВИЧНЫЙ КЛЮЧ (Номер_переплета)

ВНЕШНИЙ КЛЮЧ (Код_издания ИЗ Издания

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)

ПОЛЯ (Номер_переплета Целое, Код_издания Целое,

Цена Деньги, Дата_приобретения Дата)

ОГРАНИЧЕНИЯ (Значения поля Код_издания должны принадлежать набору значений соответствующего поля таблицы Издания; при нарушении вывод сообщения "Такого издания нет");

СОЗДАТЬ ТАБЛИЦУ Аннотации *(Характеризует Издания)
ПЕРВИЧНЫЙ КЛЮЧ (Код_издания)
ВНЕШНИЙ КЛЮЧ (Код_издания ИЗ Издания
NULL-значения ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ
ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)
ПОЛЯ (Код_издания Целое, Аннотация Запись)
ОГРАНИЧЕНИЯ (Значения поля Код_издания должны принадлежать
набору значений соответствующего поля таблицы
Издания; при нарушении вывод сообщения "Такого
издания нет");

СОЗДАТЬ ТАБЛИЦУ Авторы *(Связывает Создатели и Издания)
ПЕРВИЧНЫЙ КЛЮЧ (Код_создателя, Код_издания)
ВНЕШНИЙ КЛЮЧ (Код_создателя ИЗ Создатели
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Создатели ОГРАНИЧИВАЕТСЯ
ОБНОВЛЕНИЕ Создатели.Код_создателя КАСКАДИРУЕТСЯ)
ВНЕШНИЙ КЛЮЧ (Код_издания ИЗ Издания
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ
ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)
ПОЛЯ (Код_создателя Целое, Код_издания Целое)
ОГРАНИЧЕНИЯ (Значения полей Код_создателя и Код_издания должны
принадлежать набору значений соответствующих полей
таблиц Создатели и Издание; при нарушении вывод
сообщения"Такого автора нет"или"Такого издания нет");

Аналогичное содержание имеют описания таблиц Составители, Редакторы, Художники и Переиздания. Остальные же таблицы проектируемой базы данных описываются так:

СОЗДАТЬ ТАБЛИЦУ Переводчики *(Связывает Создатели, Издания и Языки)
ПЕРВИЧНЫЙ КЛЮЧ (Код_создателя, Код_издания)
ВНЕШНИЙ КЛЮЧ (Код_создателя ИЗ Создатели
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Создатели ОГРАНИЧИВАЕТСЯ
ОБНОВЛЕНИЕ Создатели.Код_создателя КАСКАДИРУЕТСЯ)
ВНЕШНИЙ КЛЮЧ (Код_издания ИЗ Издания
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Издания ОГРАНИЧИВАЕТСЯ
ОБНОВЛЕНИЕ Издания.Код_издания КАСКАДИРУЕТСЯ)

ВНЕШНИЙ КЛЮЧ (Код_языка ИЗ Языки
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Языки ОГРАНИЧИВАЕТСЯ
ОБНОВЛЕНИЕ Языки.Код_языка КАСКАДИРУЕТСЯ)

ПОЛЯ (Код_создателя Целое, Код_издания Целое)

ОГРАНИЧЕНИЯ (Значения полей Код_создателя, Код_издания и
Код_языка должны принадлежать набору значений
соответствующих полей таблиц Создателя, Издание
и Языки; при нарушении вывод сообщения "Такого
автора нет" или "Такого издания нет" или "Такого
языка нет");

СОЗДАТЬ ТАБЛИЦУ Размещение *(Связывает Места и Переплеты)

ПЕРВИЧНЫЙ КЛЮЧ (Код_места, Номер_переплета)

ВНЕШНИЙ КЛЮЧ (Код_места ИЗ Места
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Места ОГРАНИЧИВАЕТСЯ
ОБНОВЛЕНИЕ Места.Код_места КАСКАДИРУЕТСЯ)

ВНЕШНИЙ КЛЮЧ (Номер_переплета ИЗ Переплеты
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Переплеты КАСКАДИРУЕТСЯ
ОБНОВЛЕНИЕ Переплеты.Ном_переплета КАСКАДИРУЕТСЯ)

ПОЛЯ (Код_места Целое, Номер_переплета Целое,
Дата_размещения Дата, Дата_изъятия Дата)

ОГРАНИЧЕНИЯ (Значения полей Код_места и Номер_переплета
должны принадлежать набору значений
соответствующих полей таблиц Переплеты и Места;
при нарушении вывод сообщения "Такого переплета
нет" или "Такого места нет");

СОЗДАТЬ ТАБЛИЦУ Выдача *(Связывает Читатели и Переплеты)

ПЕРВИЧНЫЙ КЛЮЧ (Ном_билета, Ном_переплета)

ВНЕШНИЙ КЛЮЧ (Ном_билета ИЗ Читатели
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Читатели КАСКАДИРУЕТСЯ
ОБНОВЛЕНИЕ Читатели.Ном_билета КАСКАДИРУЕТСЯ)

ВНЕШНИЙ КЛЮЧ (Ном_переплета ИЗ Переплеты
NULL-значения НЕ ДОПУСТИМЫ
УДАЛЕНИЕ ИЗ Переплеты КАСКАДИРУЕТСЯ
ОБНОВЛЕНИЕ Переплеты.Ном_переплета КАСКАДИРУЕТСЯ)

ПОЛЯ (Ном_билета Целое, Ном_переплета Целое,
Дата_выдачи Дата, Срок Целое, Дата_возврата Дата)
ОГРАНИЧЕНИЯ (Значения полей Ном_билета и Ном_переплета должны
принадлежать набору значений соответствующих полей
таблиц Читатели и Переплеты; при нарушении вывод
сообщения "Такого читателя нет" или "Такого переплета
нет").

Теперь следует проверить, не нарушены ли в данном проекте какие-либо принципы нормализации, т.е. что любое не ключевое поле каждой таблицы:

- функционально зависит от полного первичного ключа, а не от его части (если ключ составной);
- не имеет функциональной зависимости от другого неключевого поля.
- Сущности Авторы, Составители, Редакторы, Художники и Переиздания, не имеющие неключевых полей, безусловно нормализованы. Нормализованы и сущности Создатели, Характеры, Заглавия, Вид_издания и Аннотации, состоящие из несоставного ключа и единственного неключевого поля.

Анализ сущностей Переводчики, Размещение и Выдача, состоящих из составного ключа и не ключевых полей, показал, что в них нет функциональных связей между не ключевыми полями. Последние же не зависят функционально от какой-либо части составного ключа.

Наконец, анализ сущностей Издания, Переплеты, Места, Читатели и Языки, показал, что единственной "подозрительной" сущностью является стержень Языки, имеющий два функционально связанных неключевых поля: Язык и Сокращение.

Поле Язык стало не ключевым из-за ввода цифрового первичного ключа Код_языка, заменяющего текстовый возможный ключ Язык. Это позволило уменьшить объем хранимых данных в таблице Переводчики, затраты труда на ввод множества текстовых значений и возможной противоречивости, которая часто возникает из-за ввода в разные поля ошибочных дубликатов (например, "Английский", "Англиский", "Анлийский", "Англиский" и т.п.). Если мы вспомним рекомендации о замене на время нормализации цифровых заменителей первичных ключей (Код_языка) на исходный ключ (Язык) или воспользуемся формулировкой НФБК, то окажется, что таблица Языки – нормализована.

Для завершения проекта необходимо ввести в описания таблиц дополнительные сведения об ограничениях целостности (выше указан лишь минимальный их набор) и дать описание некоторых таблиц.

3.3. Объектно-ориентированное проектирование

Концептуальной основой объектно-ориентированного анализа и проектирования ПО (ООАП) является объектная модель. Ее основные принципы (абстрагирование, инкапсуляция, модульность и иерархия) и понятия (объект, класс, атрибут, операция, интерфейс и др.) наиболее четко сформулированы Гради Бучем в его фундаментальной книге [Г. Буч -- Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд.: Пер. с англ. -- М.: Издательство Бином, СПб.: Невский диалект, 1999] и последующих работах.

Большинство современных методов ООАП основаны на использовании языка UML. Унифицированный язык моделирования UML (Unified Modeling Language) представляет собой язык для определения, представления, проектирования и документирования программных систем, организационно-экономических систем, технических систем и других систем различной природы. UML содержит стандартный набор диаграмм и нотаций самых разнообразных видов.

UML - это преемник того поколения методов ООАП, которые появились в конце 1980-х и начале 1990-х годов. Создание UML фактически началось в конце 1994 г., когда Гради Буч и Джеймс Рамбо начали работу по объединению их методов Booch и OMT (Object Modeling Technique) под эгидой компании Rational Software. К концу 1995 г. они создали первую спецификацию объединенного метода, названного ими Unified Method, версия 0.8. Тогда же в 1995 г. к ним присоединился создатель метода OOSE (Object-Oriented Software Engineering) Ивар Якобсон. Таким образом, UML является прямым объединением и унификацией методов Буча, Рамбо и Якобсона, однако дополняет их новыми возможностями. Главными в разработке UML были следующие цели:

- предоставить пользователям готовый к использованию выразительный язык визуального моделирования, позволяющий им разрабатывать осмысленные модели и обмениваться ими;
- предусмотреть механизмы расширяемости и специализации для расширения базовых концепций;
- обеспечить независимость от конкретных языков программирования и процессов разработки.
- обеспечить формальную основу для понимания этого языка моделирования (язык должен быть одновременно точным и доступным для понимания, без лишнего формализма);
- стимулировать рост рынка объектно-ориентированных инструментальных средств;
- интегрировать лучший практический опыт.

UML находится в процессе стандартизации, проводимом OMG

(Object Management Group) - организацией по стандартизации в области объектно-ориентированных методов и технологий, в настоящее время принят в качестве стандартного языка моделирования и получил широкую поддержку в индустрии ПО. UML принят на вооружение практически всеми крупнейшими компаниями - производителями ПО (Microsoft, Oracle, IBM, Hewlett-Packard, Sybase и др.). Кроме того, практически все мировые производители CASE-средств, помимо IBM Rational Software, поддерживают UML в своих продуктах (Oracle Designer, Together Control Center (Borland), AllFusion Component Modeler (Computer Associates), Microsoft Visual Modeler и др.). Полное описание UML можно найти на сайтах <http://www.omg.org> и <http://www.rational.com>.

Стандарт UML содержит следующий набор диаграмм:

- Структурные (structural) модели:
 - диаграммы классов (class diagrams) - для моделирования статической структуры классов системы и связей между ними;
 - диаграммы компонентов (component diagrams) - для моделирования иерархии компонентов (подсистем) системы;
 - диаграммы размещения (deployment diagrams) - для моделирования физической архитектуры системы.
- Модели поведения (behavioral):
 - диаграммы вариантов использования (use case diagrams) - для моделирования функциональных требований к системе (в виде сценариев взаимодействия пользователей с системой);
 - диаграммы взаимодействия (interaction diagrams):
 - диаграммы последовательности (sequence diagrams) и кооперативные диаграммы (collaboration diagrams) - для моделирования процесса обмена сообщениями между объектами;
 - диаграммы состояний (statechart diagrams) - для моделирования поведения объектов системы при переходе из одного состояния в другое;
 - диаграммы деятельности (activity diagrams) - для моделирования поведения системы в рамках различных вариантов использования, или потоков управления.

Диаграммы вариантов использования показывают взаимодействия между вариантами использования и действующими лицами, отражая функциональные требования к системе с точки зрения пользователя. Цель построения диаграмм вариантов использования - это документирование функциональных требований в самом общем виде, поэтому они должны быть предельно простыми.

Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициа-

руемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между пользователем и системой и отражает представление о поведении системы с точки зрения пользователя. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать, или целей, которые он преследует по отношению к разрабатываемой системе.

Диаграмма вариантов использования является самым общим представлением функциональных требований к системе. Для последующего проектирования системы требуются более конкретные детали, которые описываются в документе, называемом "сценарием варианта использования" или "поток событий" (flow of events). Основной поток событий описывает нормальный ход событий (при отсутствии ошибок). Альтернативные потоки описывают отклонения от нормального хода событий (ошибочные ситуации) и их обработку.

Достоинства модели вариантов использования заключаются в том, что она:

- определяет пользователей и границы системы;
- определяет системный интерфейс;
- удобна для общения пользователей с разработчиками;
- используется для написания тестов;
- является основой для написания пользовательской документации;
- хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).

Диаграммы взаимодействия описывают поведение взаимодействующих групп объектов (в рамках варианта использования или некоторой операции класса). Как правило, диаграмма взаимодействия охватывает поведение объектов в рамках только одного потока событий варианта использования. На такой диаграмме отображается ряд объектов и те сообщения, которыми они обмениваются между собой. Существует два вида диаграмм взаимодействия: диаграммы последовательности и кооперативные диаграммы.

Диаграммы последовательности отражают временную последовательность событий, происходящих в рамках варианта использования, а кооперативные диаграммы концентрируют внимание на связях между объектами.

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня

абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации).

Диаграммы состояний определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий. Диаграммы состояний не надо создавать для каждого класса, они применяются только в сложных случаях. Если объект класса может существовать в нескольких состояниях и в каждом из них ведет себя по-разному, для него может потребоваться такая диаграмма.

Диаграммы деятельности, в отличие от большинства других средств UML, заимствуют идеи из нескольких различных методов, в частности, метода моделирования состояний SDL и сетей Петри. Эти диаграммы особенно полезны в описании поведения, включающего большое количество параллельных процессов. Диаграммы деятельности являются также полезными при параллельном программировании, поскольку можно графически изобразить все ветви и определить, когда их необходимо синхронизировать.

Диаграммы деятельности можно применять для описания потоков событий в вариантах использования. С помощью текстового описания можно достаточно подробно рассказать о потоке событий, но в сложных и запутанных потоках с множеством альтернативных ветвей будет трудно понять логику событий. Диаграммы деятельности предоставляют ту же информацию, что и текстовое описание потока событий, но в наглядной графической форме.

Диаграммы компонентов моделируют физический уровень системы. На них изображаются компоненты ПО и связи между ними. На такой диаграмме обычно выделяют два типа компонентов: исполняемые компоненты и библиотеки кода.

Каждый класс модели (или подсистема) преобразуется в компонент исходного кода. Между отдельными компонентами изображают зависимости, соответствующие зависимостям на этапе компиляции или выполнения программы.

Диаграммы компонентов применяются теми участниками проекта, кто отвечает за компиляцию и сборку системы. Они нужны там, где начинается генерация кода.

Диаграмма размещения отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать размещение объектов и компонентов в распределенной системе.

Диаграмма размещения показывает физическое расположение сети и

местонахождение в ней различных компонентов. Ее основными элементами являются узел (вычислительный ресурс) и соединение - канал взаимодействия узлов (сеть).

Диаграмма размещения используется менеджером проекта, пользователями, архитектором системы и эксплуатационным персоналом, чтобы понять физическое размещение системы и расположение ее отдельных подсистем.

UML обладает механизмами расширения, предназначенными для того, чтобы разработчики могли адаптировать язык моделирования к своим конкретным нуждам, не меняя при этом его метамодель. Наличие механизмов расширения принципиально отличает UML от таких средств моделирования, как IDEF0, IDEF1X, IDEF3, DFD и ERM. Перечисленные языки моделирования можно определить как сильно типизированные (по аналогии с языками программирования), поскольку они не допускают произвольной интерпретации семантики элементов моделей. UML, допуская такую интерпретацию (в основном за счет стереотипов), является слабо типизированным языком. К его механизмам расширения относятся:

- стереотипы;
- тегированные (именованные) значения;
- ограничения.

Стереотип - это новый тип элемента модели, который определяется на основе уже существующего элемента. Стереотипы расширяют нотацию модели и могут применяться к любым элементам модели. Стереотипы классов - это механизм, позволяющий разделять классы на категории. Разработчики ПО могут создавать свои собственные наборы стереотипов, формируя тем самым специализированные подмножества UML (например, для описания бизнес-процессов, Web-приложений, баз данных и т.д.). Такие подмножества (наборы стереотипов) в стандарте языка UML носят название профилей языка.

Именованное значение - это пара строк "тег = значение", или "имя = содержимое", в которых хранится дополнительная информация о каком-либо элементе системы, например, время создания, статус разработки или тестирования, время окончания работы над ним и т.п.

Ограничение - это семантическое ограничение, имеющее вид текстового выражения на естественном или формальном языке (OCL - Object Constraint Language), которое невозможно выразить с помощью нотации UML.

Гради Буч сформулировал главное достоинство объектно-ориентированного подхода (ООП) следующим образом: объектно-ориентированные системы более открыты и легче поддаются внесению изменений, поскольку их конструкция базируется на устойчивых формах.

Это дает возможность системе развиваться постепенно и не приводит к полной ее переработке даже в случае существенных изменений исходных требований.

Буч отметил также ряд следующих преимуществ ООП:

- объектная декомпозиция дает возможность создавать программные системы меньшего размера путем использования общих механизмов, обеспечивающих необходимую экономию выразительных средств. Использование ООП существенно повышает уровень унификации разработки и пригодность для повторного использования не только ПО, но и проектов, что в конце концов ведет к сборочному созданию ПО. Системы зачастую получаются более компактными, чем их не объектно-ориентированные эквиваленты, что означает не только уменьшение объема программного кода, но и удешевление проекта за счет использования предыдущих разработок;
- объектная декомпозиция уменьшает риск создания сложных систем ПО, так как она предполагает эволюционный путь развития системы на базе относительно небольших подсистем. Процесс интеграции системы растягивается на все время разработки, а не превращается в единовременное событие;
- объектная модель вполне естественна, поскольку в первую очередь ориентирована на человеческое восприятие мира, а не на компьютерную реализацию;
- объектная модель позволяет в полной мере использовать выразительные возможности объектных и объектно-ориентированных языков программирования.

К недостаткам ООП относятся некоторое снижение производительности функционирования ПО (которое, однако, по мере роста производительности компьютеров становится все менее заметным) и высокие начальные затраты. Объектная декомпозиция существенно отличается от функциональной, поэтому переход на новую технологию связан как с преодолением психологических трудностей, так и дополнительными финансовыми затратами. При переходе от структурного подхода к объектному, как при всякой смене технологии, необходимо вкладывать деньги в приобретение новых инструментальных средств. Здесь следует учесть расходы на обучение методу, инструментальным средствам и языку программирования. Для некоторых организаций эти обстоятельства могут стать серьезными препятствиями.

Объектно-ориентированный подход не дает немедленной отдачи. Эффект от его применения начинает сказываться после разработки двух-трех проектов и накопления повторно используемых компонентов, отражающих типовые проектные решения в данной области. Переход органи-

зации на объектно-ориентированную технологию - это смена мировоззрения, а не просто изучение новых CASE-средств и языков программирования.

Таким образом, структурный подход по-прежнему сохраняет свою значимость и достаточно широко используется на практике. На примере языка UML хорошо видно, что его авторы заимствовали то рациональное, что можно было взять из структурного подхода: элементы функциональной декомпозиции в диаграммах вариантов использования, диаграммы состояний, диаграммы деятельности и др. Очевидно, что в конкретном проекте сложной системы невозможно обойтись только одним способом декомпозиции. Можно начать декомпозицию каким-либо одним способом, а затем, используя полученные результаты, попытаться рассмотреть систему с другой точки зрения.

Основой взаимосвязи между структурным и объектно-ориентированным подходами является общность ряда категорий и понятий обоих подходов (процесс и вариант использования, сущность и класс и др.). Эта взаимосвязь может проявляться в различных формах. Так, одним из возможных вариантов является использование структурного анализа как основы для объектно-ориентированного проектирования. При этом структурный анализ следует прекращать, как только структурные модели начнут отражать не только деятельность организации (бизнес-процессы), а и систему ПО. После выполнения структурного анализа можно различными способами приступить к определению классов и объектов. Так, если взять какую-либо отдельную диаграмму потоков данных, то кандидатами в классы могут быть элементы структур данных.

Другой формой проявления взаимосвязи можно считать интеграцию объектной и реляционной технологий. Реляционные СУБД являются на сегодняшний день основным средством реализации крупномасштабных баз данных и хранилищ данных. Причины этого достаточно очевидны: реляционная технология используется достаточно долго, освоена огромным количеством пользователей и разработчиков, стала промышленным стандартом, в нее вложены значительные средства и создано множество корпоративных БД в самых различных отраслях, реляционная модель проста и имеет строгое математическое основание; существует большое разнообразие промышленных средств проектирования, реализации и эксплуатации реляционных БД. Вследствие этого реляционные БД в основном используются для хранения и поиска объектов в так называемых объектно-реляционных системах.

Взаимосвязь между структурным и объектно-ориентированным подходами достаточно четко просматривается в различных ТС ПО.

Методы моделирования бизнес-процессов и спецификации требований

Моделирование бизнес-процессов является важной составной частью проектов по созданию крупномасштабных систем ПО. Отсутствие таких моделей является одной из главных причин неудач многих проектов.

Назначением будущего ПО является, в первую очередь, решение проблем бизнеса. Требования к ПО формируются на основе бизнес-модели, а критерии проектирования систем прежде всего основываются на наиболее полном их удовлетворении.

Модели бизнес-процессов являются не просто промежуточным результатом, используемым консультантом для выработки каких-либо рекомендаций и заключений. Они представляют собой самостоятельный результат, имеющий большое практическое значение.

На сегодняшний день в моделировании бизнес-процессов преобладает процессный подход. Его основной принцип заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой. Модель, основанная на организационно-штатной структуре, может продемонстрировать лишь хаос, царящий в организации (о котором в принципе руководству и так известно, иначе оно бы не инициировало соответствующие работы), на ее основе можно только внести предложения об изменении этой структуры. С другой стороны, модель, основанная на бизнес-процессах, содержит в себе и организационно-штатную структуру предприятия.

Процессный подход может использовать любые из перечисленных выше средств моделирования. Однако, в настоящее время наблюдается тенденция интеграции разнообразных методов моделирования и анализа систем, проявляющаяся в форме создания интегрированных средств моделирования. Одним из таких средств является продукт, носящий название ARIS - Architecture of Integrated Information System, разработанный германской фирмой IDS Scheer.

Система ARIS представляет собой комплекс средств анализа и моделирования деятельности предприятия. Ее методическую основу составляет совокупность различных методов моделирования, отражающих разные взгляды на исследуемую систему. Одна и та же модель может разрабатываться с использованием нескольких методов, что позволяет использовать ARIS специалистам с различными теоретическими знаниями и настраивать его на работу с системами, имеющими свою специфику.

ARIS поддерживает четыре типа моделей, отражающих различные аспекты исследуемой системы:

- организационные модели, представляющие структуру системы - иерархию организационных подразделений, должностей и конкретных

лиц, связи между ними, а также территориальную привязку структурных подразделений;

- функциональные модели, содержащие иерархию целей, стоящих перед аппаратом управления, с совокупностью деревьев функций, необходимых для достижения поставленных целей;
- информационные модели, отражающие структуру информации, необходимой для реализации всей совокупности функций системы;
- модели управления, представляющие комплексный взгляд на реализацию бизнес-процессов в рамках системы.

Для построения перечисленных типов моделей используются как собственные методы моделирования ARIS, так и различные известные методы и языки моделирования - ERM, UML, OMT и др.

В процессе моделирования каждый аспект деятельности предприятия сначала рассматривается отдельно, а после детальной проработки всех аспектов строится интегрированная модель, отражающая все связи между различными аспектами.

Модели в ARIS представляют собой диаграммы, элементами которых являются разнообразные объекты - "функция", "событие", "структурное подразделение", "документ" и т.п. Между объектами устанавливаются разнообразные связи. Каждому объекту соответствует определенный набор атрибутов, которые позволяют ввести дополнительную информацию о конкретном объекте. Значения атрибутов могут использоваться при имитационном моделировании или для проведения стоимостного анализа.

Основная бизнес-модель ARIS - eEPC (extended Event Driven Process Chain - расширенная модель цепочки процессов, управляемых событиями). По существу, она расширяет возможности IDEF0, IDEF3 и DFD, обладая всеми их достоинствами и недостатками. Применение большого числа различных объектов, связанных различными типами связей, может значительно увеличить размер модели и сделать ее плохо читаемой.

Бизнес-процесс в нотации eEPC представляет собой поток последовательно выполняемых работ (процедур, функций), расположенных в порядке их выполнения. Реальная длительность выполнения процедур в eEPC визуально не отражается. Это приводит к тому, что при создании моделей возможны ситуации, когда на одного исполнителя будет возложено выполнение двух задач одновременно. Используемые при построении модели символы логики позволяют отразить ветвление и слияние бизнес-процесса. Для получения информации о реальной длительности процессов необходимо использовать другие инструменты описания, например графики Ганта в системе MS Project.

Ряд современных методов моделирования бизнес-процессов основан на использовании языка UML. Хотя UML изначально предназначался для

моделирования систем ПО, его использование в другой области стало возможным благодаря наличию в UML механизмов расширения (стереотипов).

Среди таких методов наиболее известными являются метод Ericsson-Penker и метод, реализованный в технологии Rational Unified Process (RUP).

Метод Ericsson-Penker представляет интерес прежде всего в связи с попыткой применения UML в рамках процессного подхода к моделированию бизнес-процессов. Авторы метода создали свой профиль UML для моделирования бизнес-процессов, введя набор стереотипов, описывающих процессы, ресурсы, правила и цели деятельности организации. Метод использует четыре основные категории бизнес-модели:

- Ресурсы - различные объекты, используемые или участвующие в бизнес-процессах (люди, материалы, информация или продукты).
- Процессы - виды деятельности, изменяющие состояние ресурсов в соответствии с бизнес-правилами.
- Цели - назначение бизнес-процессов Цели могут быть разбиты на подцели и соотнесены с отдельными процессами.
- Бизнес-правила - условия или ограничения выполнения процессов (функциональные, поведенческие или структурные). Правила могут быть определены с использованием языка OCL.

Основной диаграммой UML, используемой в данном методе, является диаграмма деятельности. Метод Eriksson-Penker представляет процесс в виде деятельности со стереотипом "process" (основой данного представления является расширение метода IDEF0). Полная бизнес-модель включает множество представлений, подобных представлениям архитектуры ПО. Каждое представление выражено в одной или более диаграммах UML. Диаграммы могут иметь различные типы и изображать процессы, правила, цели и ресурсы во взаимодействиях друг с другом. Метод использует четыре различных представления бизнес-модели:

- концептуальное представление - структура целей и проблем;
- представление процессов - взаимодействие между процессами и ресурсами (в виде набора диаграмм деятельности);
- структурное представление - структура организации и ресурсов (в виде диаграмм классов);
- представление поведения - поведение отдельных ресурсов и детализация процессов (в виде диаграмм деятельности, состояний и взаимодействия).
- Методика моделирования RUP предусматривает построение двух моделей:
- модели бизнес-процессов (Business Use Case Model);

- модели бизнес- анализа (Business Analysis Model).

Модель бизнес-процессов представляет собой расширение модели вариантов использования UML за счет введения набора стереотипов - Business Actor (стереотип действующего лица) и Business Use Case (стереотип варианта использования). Business Actor (действующее лицо бизнес-процессов) - это некоторая роль, внешняя по отношению к бизнес-процессам организации. Business Use Case (вариант использования с точки зрения бизнес-процессов) определяется как описание последовательности действий (потока событий) в рамках некоторого бизнес-процесса, приносящей ощутимый результат конкретному действующему лицу. Это определение подобно общему определению бизнес-процесса, но имеет более точный смысл. В терминах объектной модели Business Use Case представляет собой класс, объектами которого являются конкретные потоки событий в рамках описываемого бизнес-процесса.

Описание Business Use Case может сопровождаться целью процесса, которая, так же, как и в методе Eriksson-Penker, моделируется с помощью класса со стереотипом "goal", а дерево целей изображается в виде диаграммы классов.

Для каждого Business Use Case строится модель бизнес-анализа - объектная модель, описывающая реализацию бизнес-процесса в терминах взаимодействующих объектов (бизнес-объектов - Business Object), принадлежащих к двум классам - Business Worker и Business Entity. Business Worker (исполнитель) - класс, представляющий собой абстракцию исполнителя, выполняющего некоторые действия в рамках бизнес-процесса. Исполнители взаимодействуют между собой и манипулируют различными сущностями, участвуя в реализациях сценариев Business Use Case. Business Entity (сущность) является объектом различных действий со стороны исполнителей.

Кроме диаграммы данных классов, модель бизнес-анализа может включать:

- Диаграммы последовательности (и кооперативные диаграммы), описывающие сценарии Business Use Case в виде последовательности обмена сообщениями между объектами-действующими лицами и объектами-исполнителями. Такие диаграммы помогают явно определить в модели обязанности каждого исполнителя в виде набора его операций.
- Диаграммы деятельности, описывающие взаимосвязи между сценариями одного или различных Business Use Case.
- Диаграммы состояний, описывающие поведение отдельных бизнес-объектов.

Методика моделирования Rational Unified Process обладает следую-

щими достоинствами:

- модель бизнес-процессов строится вокруг участников процессов (заинтересованных лиц) и их целей, помогая выявить все потребности клиентов организации. Такой подход в наибольшей степени применим для организаций, работающих в сфере оказания услуг (торговые организации, банки, страховые компании и т.д.);
- моделирование на основе вариантов использования способствует хорошему пониманию бизнес-модели со стороны заказчиков.

Однако следует отметить, что при моделировании деятельности крупной организации, занимающейся как производством продукции, так и оказанием услуг, необходимо применять различные методики моделирования, поскольку для моделирования производственных процессов более предпочтительным является процессный подход (например, метод Eriksson-Penker).

Спецификация требований к ПО является составной частью процесса управления требованиями [15]. Выявленные в результате применения перечисленных методов требования к ПО оформляются в виде ряда документов и моделей. Так, в технологии RUP функциональные требования к системе моделируются и документируются с помощью вариантов использования. Стиль их написания зависит от масштаба, количества участников и критичности проекта.

Спецификация требований в RUP не требует обязательного моделирования бизнес-процессов организации, для которых создается ПО, однако, наличие бизнес-моделей существенно упрощает построение системной модели вариантов использования.

Методы анализа и проектирования ПО

Целью анализа требований является трансформация функциональных требований к ПО в предварительный системный проект и создание стабильной основы архитектуры системы. В процессе проектирования системный проект "погружается" в среду реализации с учетом всех нефункциональных требований.

Все современные ТС ПО реализуют ту или иную методику анализа и проектирования ПО. Одна из типичных методик ООАП реализована в технологии RUP. Согласно этой методике, объектно-ориентированный анализ включает два вида деятельности: архитектурный анализ и анализ вариантов использования. Архитектурный анализ выполняется архитектором системы и включает в себя:

- утверждение общих стандартов (соглашений) моделирования и документирования системы;
- предварительное выявление архитектурных механизмов (надежности, безопасности и т.д.);

- формирование набора основных абстракций предметной области (классов анализа);
- формирование начального представления архитектурных уровней. Анализ вариантов использования выполняется проектировщиками и

включает в себя:

- идентификацию классов, участвующих в реализации потоков событий варианта использования;
- распределение поведения, реализуемого вариантом использования, между классами (определение обязанностей классов);
- определение атрибутов и ассоциаций классов.

В потоках событий варианта использования выявляются классы трех типов:

- Граничные классы (Boundary) - служат посредниками при взаимодействии внешних объектов с системой. Типы граничных классов: пользовательский интерфейс (обмен информацией с пользователем, без деталей интерфейса - кнопок, списков, окон), системный интерфейс и аппаратный интерфейс (используемые протоколы, без деталей их реализации).
- Классы-сущности (Entity) - представляют собой основные абстракции (понятия) разрабатываемой системы, рассматриваемые в рамках конкретного варианта использования.
- Управляющие классы (Control) - обеспечивают координацию поведения объектов в системе. Примеры управляющих классов: менеджер транзакций, координатор ресурсов, обработчик ошибок.

Классы анализа отражают функциональные требования к системе и моделируют объекты предметной области. Совокупность классов анализа представляет собой начальную концептуальную модель системы

Наиболее важной частью объектно-ориентированного анализа является распределение обязанностей между классами (в виде операций классов). Оно выполняется с помощью диаграмм взаимодействия. При построении диаграмм взаимодействия возникают проблемы правильного распределения обязанностей между классами. Для их решения существует ряд образцов [14].

Атрибуты классов анализа определяются, исходя из знаний о предметной области и требований к системе. Связи между классами (ассоциации) определяются на основе анализа кооперативных диаграмм, затем анализируются и уточняются.

Целью объектно-ориентированного проектирования является адаптация предварительного системного проекта (набора классов "анализа"), составляющего стабильную основу архитектуры системы, к среде реализации с учетом всех нефункциональных требований.

Объектно-ориентированное проектирование включает два вида деятельности:

- проектирование архитектуры системы;
- проектирование элементов системы.

Проектирование архитектуры системы выполняется архитектором системы и включает в себя:

- идентификацию архитектурных решений и механизмов, необходимых для проектирования системы;
- анализ взаимодействий между классами анализа, выявление подсистем и интерфейсов;
- формирование архитектурных уровней;
- проектирование конфигурации системы.

Проектирование элементов системы включает в себя:

- проектирование классов (детализация классов, уточнение операций и атрибутов, моделирование состояний, уточнение связей между классами);
- проектирование баз данных (в зависимости от типа используемой для хранения данных СУБД - объектной или реляционной).

3.4. Проектирование задач обработки данных

Задачи обработки данных в АИС могут сводиться к выполнению типовых и специализированных функций.

К типовым задачам обработки данных относятся: определение схемы БД, ввод, коррекция, контроль целостности БД, просмотр, упорядочение, поиск, отбор, группировка, расчеты, формирование и вывод отчетов. Рассмотрим постановку этих задач применительно к реляционной базе данных.

Определение схемы является первой задачей даталогического проектирования базы данных и состоит в описании на специальном языке полей БД (их имен, типов, значений по умолчанию и т.д.), ключевых полей, связей между таблицами, хранимых процедур и триггеров. Это описание используется СУБД для создания на внешних носителях файлов БД, для обработки запросов и контроля целостности БД.

При определении схемы БД обычно выделяют более мелкие операции создания, копирования, переименования и удаления объектов БД (файла, таблицы, индекса и др.).

Многие современные СУБД обеспечивают ведение *словаря данных*, в котором содержится описание схемы БД. При этом СУБД автоматически отслеживает соответствие словаря и базы данных, что значительно облегчает работу администратора БД по модернизации БД.

Задача *ввода данных* состоит в переносе данных с первичных носителей в БД и контроле их достоверности. Операция контроля всегда сопут-

ствуется вводу, так как неверные данные нет смысла обрабатывать.

Задача коррекции данных включает в себя операции замены, удаления и добавления данных в БД; они могут применяться к отдельным полям, записям, файлу или БД в целом. Так же, как и при вводе, контроль необходим и при коррекции.

Контроль целостности БД состоит в проверке ограничений целостности при любой попытке изменения данных в БД. При нарушении правильности и согласованности данных текущая операция, приводящая к изменению состояния БД, отменяется.

Просмотр данных в БД заключается в выводе на экран монитора фрагмента БД, который интересует пользователя. При просмотре данных должна обеспечиваться возможность навигации по БД. Форма представления информации должна быть удобна для пользователя. Как правило, просмотр данных совмещается с их коррекцией, но иногда коррекция при просмотре бывает запрещена.

Упорядочение данных состоит в изменении логического или физического порядка записей в БД. Изменение логического порядка называют также индексацией, так как оно связано с созданием дополнительных по отношению к таблицам объектов БД - индексов. Многие СУБД автоматически упорядочивают записи по значению ключевого поля таблицы, в других - создание индексов целиком ложится на пользователя (СУБД семейства dBase).

Как правило, индексация более эффективна, так как занимает меньше времени, позволяет многократно использовать и поддерживать однажды созданный логический порядок записей, обеспечивает более быстрый поиск данных и правильную установку связей между таблицами.

Поиск данных состоит в определении конкретной записи, которая удовлетворяет заданному условию. Эта задача решается путем перебора записей в индексе или таблице по тому или иному алгоритму. Поиск прекращается, как только очередная запись удовлетворяет условию поиска или когда выполняется заданный критерий прекращения поиска, например, достигнут конец файла, нажата определенная клавиша и т.п.

Отбор (выборка, селекция) данных состоит в выделении из БД всех записей, которые удовлетворяют заданному условию. Отличие задачи отбора от задачи поиска состоит в том, что последняя имеет целью найти одну единственную запись, а отбор выделяет группу записей (которая, впрочем, может состоять и из одной записи и даже быть пустой). Все прочие записи как бы скрываются, становятся не видны пользователю.

Группировка данных (разрез, сводка) - это объединение данных, сходных по заданному критерию, в группы.

Задача расчета состоит в вычислении некоторых показателей. Наи-

более типичными из групповых расчетных операций являются вычисление суммы и среднего значения числовых полей, определение количества записей в группе.

Задача формирования результатов состоит в оформлении результатов запросов к БД в форме документов, соответствующих принятым стандартам и удобным для восприятия.

Задача вывода отчетов представляет собой перенос сформированных отчетов из памяти ЭВМ на печать, экран, в файл или канал связи.

Примеры запросов *поиска*:

- определить, сколько заданного товара осталось на складе;
- найти продавца, который продал конкретный товар в заданный день.

Примеры формулировки *запросов выборки*:

- вывести перечень товаров ценой более 100 руб.;
- выдать список наименований, которых на складе осталось меньше 5 штук.

Примеры *вычислительных запросов*:

- какова сумма продаж за январь текущего года с группировкой по типам товаров;
- вывести сравнительные суммы продаж за год по каждому продавцу, группировка по отделам;
- каков график продажи товара за прошедшие полгода.

Примеры запросов *на коррекцию данных*:

- удалить сведения о реализации товаров за прошлый год;
- изменить цену товаров в связи с изменением курса валюты.

Примеры задач формирования и выдачи отчетов:

- напечатать рекламные листовки по заданной группе товаров (рекламная информация содержится в БД);
- вывести в файл и на печать товарный чек о покупке заданного товара;
- оформить и выдать на печать и экран диаграмму и таблицу продаж заданного товара за заданный период.

К специализированным задачам относятся задачи, реализация которых невозможна типовыми задачами обработки данных. Как правило, такие задачи предполагают использование программных модулей использующих запросы к БД и на их основе выполняющих расчеты по методикам используемым в предметной области. Примером таких задач могут являться, оптимизационные расчеты показателей, прогнозирование показателей и т.д.

Глава 4. Реализация АИС

Данный раздел посвящен описанию основных этапов реализации проектных решений изложенных в предыдущей главе.

4.1. Реализация БД.

4.2. Реализация задач обработки данных.

4.3. Реализация пользовательского и программного интерфейса.

Глава 5. Обоснование экономической эффективности проекта.

5.1. Выбор и обоснование источников и методики оценки экономической эффективности проекта

В данном разделе необходимо обосновать источники экономического эффекта, а также выбрать методику оценки показателей экономической эффективности. При этом необходимо представлять, что экономический эффект может проявляться в следующих случаях.

1. Экономический эффект возникает от решения экономической задачи.

2. Экономический эффект возникает при использовании автоматизированных средств обработки экономической информации и задач управления экономическим объектом.

Оптимальным является выявление и оценка экономического эффекта в обоих случаях, однако при большом объеме расчетов допускается выбор одного из вариантов.

После обоснования источников эффекта необходимо выбрать и обосновать методику расчета показателей эффективности.

Оценка эффективности используемой модели ЭВМ связана с получением некоторого полезного результата - эффекта. Однако, этот эффект достигается ценой затрат определенных ресурсов. Поэтому эффективность автоматизации рассматривается в виде соотношения между выигрышем и затратами. Это соотношение определяет конкретные количественные характеристики ее эффективности. Они должны выбираться исходя из назначения автоматизации.

Показатели эффективности используемой информационной системы зависят от множества самых различных факторов. Их можно объединить в несколько групп.

К **первой группе** можно отнести факторы, связанные с параметрами входных информационных потоков, поступающих на обработку в ЭВМ или в вычислительную систему /ВС/. К ним относятся :

- объем информации в единицу времени и его изменение во времени (в течение суток, месяца, года);
- тип носителя входной информации;
- характер входной информации (соотношение между алфавитной и цифровой информацией и др.).

Во **вторую группу** можно включить факторы, зависящие от характе-

ра задач, которые должны решаться на ЭВМ или ВС, и алгоритмов их решения. Такие факторы включают:

- срочность задач;
- допустимость задержки в выдаче результатов, а также величина допустимой задержки;
- возможность разделения задач на подзадачи, которые можно решать в разное время или на различных средствах (например, разных ЭВМ);
- количество и качество стандартных программ и условно-постоянной информации, используемых при решении задач;
- наличие или отсутствие специального программного обеспечения (например, пакетов прикладных программ), ориентированных на характер решаемых задач и т.п.

К **третьей группе** целесообразно отнести факторы, определяемые техническими характеристиками ЭВМ и ВС. Укажем лишь некоторые из них:

- производительность процессора;
- емкость оперативной памяти;
- емкость и быстродействие внешней памяти;
- система счисления, используемая для ввода и обработки данных;
- форма представления данных при вводе, выводе и обработки данных;
- степень развитости системы команд с точки зрения обработки конкретных задач;
- режимы работы (пакетные, разделения времени и др.);
- возможности объединения в многопроцессорные и многомашинные комплексы;
- возможности подключения достаточно широкого набора разнообразных устройств ввода-вывода;
- степень полноты автоматического контроля выполнения операций.

В **четвертую группу** можно включить эксплуатационные характеристики ЭВМ и ВС:

- надежность ЭВМ и ВС и их отдельных устройств, а также связанные с надежностью характеристики (средняя наработка на отказ, полезное суточное время работы и др.);
- общая потребляемая мощность;
- требуемые условия эксплуатации;
- необходимый штат обслуживающего персонала и его квалификация.

В пятую группу факторов целесообразно выделить стоимостные показатели, к которым принято относить следующие:

- капитальные вложения, т.е. затраты на приобретение и установку ЭВМ и ВС;
- затраты на содержание обслуживающего персонала;
- затраты на электроэнергию;
- затраты на проведение и организацию профилактических и ремонтных работ;
- затраты на вспомогательные материалы (включая расходы на бумагу для печати, чернила, тонер, ленту для печатающих устройств и др.) и оборудование.

На основе анализа задач, алгоритмов их решения, входных потоков информации можно определить требования к набору основных технических характеристик ЭВМ и ВС. Каждая реальная ЭВМ или ВС обладает конкретными значениями основных технических характеристик.

Современные ЭВМ и ВС характеризуются большим числом различных технических, эксплуатационных и экономических параметров и показателей. Практически учесть все характеристики ЭВМ и ВС **невозможно**.

Многие из них, например, степень развития системного программного обеспечения, полнота функционального контроля и диагностика неисправностей, форма представления чисел и т.п., в основном носят качественный характер и трудно поддаются количественной оценке. Целесообразно определить минимальный набор основных технических характеристик, допускающих количественную трактовку, чтобы была возможность оценить значение каждой характеристики. Для обоснования выбора ЭВМ и ВС необходимо сопоставить набор характеристик с требуемыми для решения конкретной задачи параметрами.

В разделе **выбор и обоснование методики расчета экономической эффективности проекта** в зависимости от выбранного направления расчета должна быть изложена методика расчета экономической эффективности проекта.

5.2. Расчет показателей и обоснование экономической эффективности проекта

В данном разделе производится расчет и анализ показателей эффективности проектных предложений с учетом положений изложенных в предыдущем разделе.

Результаты **расчета показателей экономической эффективности проекта** необходимо представить в форме таблиц, графиков, рекомендуемых методическими материалами. Здесь следует определить улучшение качественных характеристик процесса управления соответствующим объектом и оценить влияние автоматизированного комплекса задач на эффек-

тивность деятельности органов управления и конечные результаты.

Ниже приводится одна из возможных методик оценки экономической эффективности проекта.

Шаг 1. "Планирование объемов производства продукции и услуг". Здесь определяется количество программных продуктов, отчетов, различной документации, расчетов и справок, которые выдаются о качестве производимой продукции, выполняемых предприятием услугах.

Шаг 2. "Планирование затрат денежных средств на функционирование предприятия". Планирование затрат денежных средств на функционирование предприятия осуществляется на основании калькулирования затрат в зависимости от численности работников, наличия ОС, затрат на их содержание и содержание инфраструктуры.

Шаг 3. "Планирование организации работ". В данном случае определяется численность работников, формы и методы оплаты труда, устанавливаются оценочные критерии деятельности работников и коллектива в целом и разрабатывается система оплаты труда.

Шаг 4. "Экономическая оценка разрабатываемых мероприятий по совершенствованию организации работы предприятия".

Выводы

Методические указания по экономической оценке разрабатываемых мероприятий

Исходные данные. Для сравнительной экономической оценки различных вариантов организации работы предприятия или подразделения необходимо иметь следующие исходные данные:

1. Площадь помещения и его балансовая стоимость, его расположение
2. Общая площадь и балансовая стоимость здания;
3. Перечень, количество, балансовая стоимость и потребляемая мощность токоприемников основного и вспомогательного оборудования;
4. Перечень, количество и балансовая стоимость материальных ценностей и инвентаря, стоимость которых выше 5 тыс. рублей;
5. Объем продукции работ, услуг, выполняемых предприятием или подразделением;
6. Нормы амортизационных отчислений на содержание, ремонт зданий, оборудования, инструментов;
7. Нормы расхода материалов и запасных частей на обслуживание и ремонт компьютерной и другой оргтехники и на производство продукции, работ, услуг – нормы расхода вспомогательных материалов;
8. Часовые тарифные ставки основных производственных рабочих, ИТР и служащих;
9. Процент дополнительной заработной платы и начисление на социальную защиту;

10. Затраты труда на выполнение работ и услуг до и после внедрения мероприятий;

11. Стоимость единицы различных производственно-технических ресурсов (запасных частей, материалов, воды, газа, тепла).

Рассмотрим содержание некоторых шагов проекта.

Шаг 1. Планирование объемов производства продукции и услуг

Объемы работ и услуг по информационному обеспечению предприятия или подразделения могут планироваться следующим образом:

- по среднегодовому значению показателя за последние 3-5 лет;
- если структура разрабатывается впервые, то используется имеющаяся нормативная база аналогичного предприятия;
- путем нормирования затрат труда на основании фактических данных и наблюдений.

При планировании трудозатрат работы и услуги по информационно-технологическому обеспечению или на обслуживание и ремонт компьютерной техники и оргтехники исходят из периодичности ее использования с учетом интенсивности загрузки. Затраты на текущий ремонт планируются на основании нормативов затрат аналогичных фирм и количества отказов.

Если виды работ, которые выполняются структурным подразделением имеют большую номенклатуру, то выбираемый вид продукции, работ, услуг является основным для него, а остальные приводятся к нему по формуле:

$$A_{np} = \frac{\sum A_i \times t_i}{t_{np}}$$

где A_{np} – объем продукции, работ, услуг в натуральных единицах;

t_i – трудоемкость i -ого вида продукции, работ, услуг (нормо-часов, человеко-часов);

t_{np} – трудоемкость того вида продукции, работ, услуг, к которому приводится весь объем работ по предприятию.

Определение стоимости основных производственных фондов осуществляется по формуле:

$$C_o = C_{зд} + C_{об} + C_u ;$$

где C_o – стоимость основных производственных фондов;

$C_{зд}$ – стоимость здания (помещения); $C_{об}$ – стоимость оборудования;

C_u – стоимость инвентаря, срок службы которого более 1 года или цена свыше 5 т. руб.

Стоимость здания (помещения) определяется по формуле:

$$C_{зд} = C'_{уд} \times S,$$

где $C_{уд}$ - удельная стоимость здания (помещения), руб./м²;

S – площадь здания (помещения), м².

Удельная стоимость 1 м² здания (помещения) определяется по формуле:

$$C'_{уд} = \frac{B_{ст}^{зд}}{S_{\Sigma}},$$

где $B_{ст}^{зд}$ - балансовая стоимость здания (помещения), руб.;

S - суммарная производственная площадь здания (помещения), м².

Балансовая стоимость определяется по данным предприятия или, если планируется новое строительство здания (помещения) для офиса, то определяется по удельным расценкам стоимости 1 м² строительной организации.

Стоимость основных производственных фондов ($C_{об}$) определяется на основании численности работающего персонала, объемов производимой продукции, работ и услуг и нормативов трудозатрат на их производство. $C_{об}$ определяется на основании расчетной табл.3.4.

Таблица 3.4

Стоимость основных средств отдела (подразделения)

Наименование	Стоимость единицы, руб.	Количество, шт.		Цена, руб.		Установленная мощность токоприемников, кВт
		исходное	проектируемое	исходная	проектируемая	
1	2	3	4	5	6	7
Инвентарь						
Стол	1600	5	5	8000	8000	-
Стул	680	8	8	5440	5440	-
Шкаф	5400	4	4	21600	21600	-
Тумбочка	650	5	5	3250	3250	-
ИТОГО	-	-	22	38290	38290	-
Оргтехника до внедрения проекта						
Факс	2900	1	1	2900	2900	0,72
Телефон	900	2	2	1800	1800	0,348
Ксерокс	5800	1	1	5800	5800	0,69
Калькулятор	420	5	5	2100	2100	-
ИТОГО	-	-	9	12600	12600	-

Оргтехника после внедрения проекта						
Компьютер	32000	-	3	-	92000	0,64
Принтер	6200	-	1	-	6200	0,48
Сканер	2900	-	1	-	2900	0,66
Модем	1500	-	1	-	1500	0,45
ИТОГО	-	-	6	-	102600	-
Программное обеспечение						
Программа «1С:Предприятие»	15000	-	1	-	15000	-

Балансовая стоимость единицы оборудования, используемого на предприятии, берется по фактическим учетным данным предприятия, а при строительстве зданий или приобретении нового оборудования – определяется по прейскурантам, прайс-листам, каталогам. Указанные цены корректируются на коэффициент, учитывающий затраты на доставку, монтаж, установку, подготовку специалистов по эксплуатации и обслуживанию. При этом на пассивные Ос (шкафы, столы и т.д.) применяют коэффициент 1,1, для компьютеров и оргтехники – 1,2.

Шаг 2. Планирование затрат денежных средств на функционирование отдела (подразделения)

Расчет себестоимости единицы продукции, работ или услуг отдела (подразделения) производится по одному виду продукции. Расчет себестоимости осуществляется по калькуляционным статьям и учитываются только те затраты, которые изменятся в связи с предлагаемыми к внедрению технологиями, организацией и программными продуктами.

При этом общепроизводственные и общехозяйственные расходы должны быть скорректированы прямым счетом по изменяющимся статьям (расчет пропорционально заработной плате работников не допускается).

а) затраты на расходные материалы определяется по формуле:

$$C_{pm} = H_p \times C_n,$$

где H_p - норма расходов, руб./чел.;

C_n – цена единицы материалов, руб.

Расчет затрат на расходные материалы можно представить в табл.3.5 в следующей форме:

Таблица 3.5

Затраты на расходные материалы

Наименование материалов	Норма расхода на единицу работ, кг (шт., компл.)		Цена, руб./кг	Сумма единицу работ, руб.		Возвратные отходы, руб.	
	Исход.	Проект.		Исход.	Проект.	Исход.	Проект.
1. Тонер	0,64	0,56	45	28,8	25,2	–	–
2. Бумага писчая	3	2,7	7,6	22,8	20,5	–	–
3. Канцтовары	12	12	6,4	76,8	76,8	45	45
Итого	–	–	–	128,4	122,5	45	45

б) затраты на запасные части для ремонта компьютерной и оргтехники (табл. 3.6) рассчитываются аналогично затратам на расходные материалы. Расчет затрат на расходные материалы и запасные части производится не по полной их номенклатуре, а только по тем видам, по которым нормы расхода изменяются в связи с предлагаемыми к внедрению информационными технологиями или организацией работ.

Таблица 3.6

Затраты на ремонтные материалы и запасные части

Наименование	Нормы расходов, штук/ 10 комп.		Цена, руб./ед.	Сумму, руб.	
	исходная	проектируемая		исходная	проектируемая
Картридж	2	1	1780	3560	1780
Оперативная память	3	2	3000	9000	6000
Вентилятор	4	2	100	400	200
Итого	–	–	–	12560	8980

в) транспортно-заготовительные расходы определяются в процентах к основной стоимости расходных материалов, которого берется по фактическим данным предприятия, на базе которого осуществляется проектирование или рассчитывается на основании действующих тарифов на транспортное обслуживание.

г) заработанная плата работников структурного подразделения (Z_p) определяется по формуле:

$$Z_p = T_p * P_{cp},$$

где T_p – трудоемкость работ и услуг, чел.-ч.;

P_{cp} – часовая тарифная ставка среднего разряда, руб.

Часовая тарифная ставка по среднему разряду ($P_{ср}$) определяется по формуле:

$$P_{ср} = \frac{P_1 M_1 + P_2 M_2 + \dots + P_6 M_6}{M_1 + M_2 + \dots + M_6},$$

где P_1, P_6 – часовые тарифные ставки работников соответствующих разрядов по установленной тарифной сетке, руб/ч.;

M_1, \dots, M_6 – число работников соответствующих разрядов, участвующих в производстве работ и услуг или трудоемкость работ соответствующего разряда, чел. (чел.-ч.)

$$P_{ср0} = \frac{30 \times 1 + 32,5 \times 1 + 35 \times 1}{3} = 32,5 \text{ руб./ч}$$

$$P_{ср1} = \frac{30 \times 1 + 32,5 \times 1 + 35 \times 2}{4} = 33,5 \text{ руб./ч}$$

$$Зр_0 = 89 \times 32,5 = 2893 \text{ руб.}$$

$$Зр_1 = 98 \times 33,5 = 3283 \text{ руб.}$$

д) дополнительная зарплата производственным рабочим определяется в процентах к основной зарплате. Величина процента берется по фактическим данным предприятия (8,9 %)

$$Зд_0 = 2893 \times 0,089 = 257 \text{ руб.}$$

$$Зд_1 = 3283 \times 0,089 = 292 \text{ руб.}$$

е) социальный налог начисляется в установленном размере в 25,8 % от суммы основной и дополнительной зарплат производственных рабочих

$$Зн_0 = (2893 + 257) \times 0,258 = 813 \text{ руб.}$$

$$Зн_1 = (3283 + 292) \times 0,258 = 922 \text{ руб.}$$

ж) затраты на силовую электроэнергию (Эл) рассчитываются укрупнено на основании установленной на оборудовании подразделения мощности токоприемников по формуле:

$$Эл = \frac{(F_c * K_c * K_{об} * \Phi_{до}) * Ц_э}{A},$$

где F_c – установленная мощность токоприемников, кВт;

$K_c, K_{об}$ – средние коэффициенты соответственно спроса и использования оборудования;

$Ц_э$ – стоимость одного кВт ч электроэнергии, руб.;

$\Phi_{до}$ – действительный фонд времени работы оборудования, ч.;

A – производственная программа подразделения, шт. (пр. ед., усл. ед.):

$$Эл = \frac{(62,7 * 0,6 * 0,87 * 2040) * 1,4}{47} = 1989 \text{ руб. ,}$$

$$Эл = \frac{(107,2 * 0,6 * 0,87 * 2040) * 1,4}{64} = 2496 \text{ руб. ,}$$

з) амортизационные отчисления (АО) и отчисления на (КТр) основных производственных фондов определяются по следующим формулам:

$$АО = \frac{Сзд Qзд + Соб Qоб + Спу Qпу}{100 * A},$$

$$Кр = \frac{Сзд Рзд + Сзд Роб + Спу Рпу}{100 * A},$$

где *Сзд*, *Соб*, *Спу* – балансовая стоимость здания, оборудования, ценного инструмента, приспособлений и инвентаря, руб.;

Qзд, *Qоб*, *Qпу* – средняя норма амортизационных отчислений соответственно здания, оборудования, программных продуктов и инвентаря, %;

Рзд, *Роб*, *Рпу* – средняя норма отчислений на ремонт соответственно здания, оборудования, программных продуктов и инвентаря, %.

$$АО_0 = \frac{126000 * 2,5 + 265000 * 6 + 5400 * 12,5}{100 * 47} = 420 \text{ руб. ,}$$

$$АО_1 = \frac{182000 * 2,5 + 850000 * 6 + 15000 * 12,5}{100 * 64} = 897 \text{ руб. ,}$$

$$Кр_0 = \frac{126000 * 6 + 265000 * 12,5 + 265000 * 20}{100 * 47} = 889 \text{ руб. ,}$$

$$Кр_1 = \frac{182000 * 4,5 + 850000 * 6,5 + 15000 * 12,5}{100 * 64} = 1021 \text{ руб.}$$

Себестоимость единицы продукции, работ и услуг по каждой калькуляционной статье сводится в табл. 3.7.

Таблица 3.7

Себестоимость ремонта единицы продукции по изменяющимся статьям

Наименование статей	Себестоимость ремонта по соответствующему варианту (<i>Су₀</i>), руб.	Себестоимость ремонта по проекту (<i>Су₁</i>), руб.
1. Расходные материалы	83,4	78,9

2. Запасные части	1345	192
3. Транспортно-заготовительные расходы	254	48
4. Основная зарплата персонала	2893	3283
5. Дополнительная зарплата персонала	257	292
6. Отчисление соцстраху от зарплаты персонала	813	922
7. Электроэнергия	1989	2496
8. Амортизационные отчисления	420	897
9. Ремонт	889	1021
10. Накладные расходы	3000	2000
Итого:	11943	11230

При планировании себестоимости продукции, работ, услуг необходимо учитывать общепроизводственные и общехозяйственные расходы, которые планируются 2 способами:

– процентное соотношение этих расходов к сумме основной заработной платы основного персонала или в процентах в общей структуре полной коммерческой себестоимости (общехозяйственные расходы приблизительно составляют 50% от общепроизводственных расходов).

– планируются по калькуляционным статьям расходов (табл. 3.8)

Таблица 3.8

Планирование накладных расходов по подразделению

Наименование статьи	Формула исчисления	Значения	
		исходные	проектируемые
1. Основная заработная плата административно управленческого персонала по категориям	$ЗПО = M_O \cdot K_P \cdot 12$		
- директор			
- специалисты и т.д.			
Итого			
младшего обслуживающего персонала по категориям			
-...			
Итого			
2. Дополнительная заработная плата по категориям	$ЗПД = K_D \cdot ЗПО$ K_D – коэффициент		

	учитывающий доп. заработную плату $K_D = \%ДЗП/100$		
3. Премирование работников по категориям	$ЗП_{ПР} = K_D \cdot ЗП_О$ K_D – коэффициент премирования		
4. Социальный налог	$C_H = K_{CH} \cdot (ЗП_О + ЗП_Д + ЗП_{ПР})$ K_{CH} – коэффициент социального налога		
5. Сумма заработной платы с начислениями	$ЗП = ЗП_О + ЗП_Д + ЗП_{ПР} + C_H$		
6. Коммунальные услуги			
- вода холодная	$C_B = Расц. \cdot Q$ $(Ч_{общ} \cdot Расц.)$		
- вода горячая	$C_B = Расц. \cdot Q$ $(Ч_{общ} \cdot Расц.)$		
- освещение	$C_O = P_{ЭЛ} \cdot \text{Фонд времени} \cdot Расц.$		
- силовая энергия	$C_C = P_{ЭЛ} \cdot \text{Фонд времени} \cdot Расц.$		
- отопление	$C_{OT} = S \cdot Расц.$		
7. Охрана труда и техника безопасности	$З_{ТБ} = Ч_n \cdot Oч$ $Oч$ – отчисления на 1 работника		
8. Рационализация и техника безопасности	$З_{ТБ} = Ч_n \cdot Oч$		
9. Содержание зданий	$З_{зд} = 0,5\% \text{ от } Б_{зд}$		
10. Все остальные общепроизводственные расходы			
11. Прочие общепроизводственные расходы	$Пр = 4-6\% \text{ от } \sum ОПР$		

Себестоимость продукции, работ, услуг содержит прямые и косвенные расходы, которые зависят или не зависят от объема произведенной продукции, работ, услуг. Поэтому при анализе себестоимости необходимо это учитывать и исходить из возможного исчисления себестоимости для различного объема работ, услуг.

Шаг 3. Расчет экономической эффективности проекта

Расчет экономической эффективности применения информационных технологий производится по следующим формулам:

Годовая экономия от снижения себестоимости продукции, работ и услуг:

$$\mathcal{E}_z = (C_{y_0} - C_{y_1}) \times A_1,$$

где A_1 – объем продукции, работ, услуг, выполненных подразделением.

$$\mathcal{E}_z = (11943 - 11230) \times 64 = 45632 \text{ руб.}$$

1. Годовая эффективность от снижения приведенных затрат.

$$\mathcal{E} = \left[(C_{y_0} + E_n \times K_{y_1}) + (C_{y_1} + E_n \times K_{y_1}) \right] \times A_1,$$

где K_{y_0} , K_{y_1} – капиталовложения, отнесенные на единицу объема работ по подразделению;

E_n – нормативный коэффициент капиталовложений. Обычно данный коэффициент принимается равным процентной ставке рефинансирования Центробанка плюс процент за обслуживание кредита.

$$K_{y_0} = \frac{K_0(K_1)}{A},$$

$$\mathcal{E} = (11943 + 0,24 \times 39640/47) - (11230 + 0,24 \times 104700/64) = 55565 \text{ руб.}$$

2. **Срок окупаемости дополнительных капиталовложений** при условии, что капиталовложения по исходному варианту меньше, чем по проектируемому ($K_1 > K_0$), определяется по формуле:

$$T_{ок} = \frac{K_1 - K_0}{\mathcal{E}} = \frac{\Delta K}{(C_{y_0} - C_{y_1}) \times A_1}$$

$$T_{ок} = \frac{104700 - 39640}{45632} = 1,4 \text{ года,}$$

Расчетный коэффициент эффективности капиталовложений определяется по формуле:

$$E_p = \frac{\mathcal{E}_z}{K_1 - K_0} = \frac{(C_{y_0} - C_{y_1}) \times A_1}{\Delta K}$$

$$E_p = \frac{45632}{104700 - 39640} = 0,7$$

Наиболее целесообразным для применения считается проектируемый вариант, если соблюдается следующее условие:

$$E_p \geq E_n$$

В результате оценки экономической эффективности от внедрения проекта информационно-технологического обеспечения предприятия, технического оснащения и совершенствования организации или технологии обработки и предоставления информации получена годовая экономия в размере около 182,0 тыс. руб. за счет снижения себестоимости работ, что позволяет окупить единовременные капиталовложения в течение 1,4 года.

Основные показатели технико-экономической оценки предприятия, подразделения или технологии заносятся в следующие табл.3.9, 3.10.

Таблица 3.9

Наименование показателей	Условные обозначения	Значения	
		исходные	проектируемые
1. Производительная программа или объем продукции, работ и услуг	A		
2. Численность работников, в т.ч.	Ч _п		
- численность основных рабочих	Ч _о		
- численность вспомогательных рабочих	Ч _в		
- численность административно-управленческого персонала	Ч _{ауп}		
3. Стоимость основных производственных фондов	Ф		
4. Полные капиталовложения в основные фонды	К ₀ , К ₁		
5. Установленная мощность токоприемников	F _с		
6. Дополнительные капиталовложения	К _д =К ₁ -К ₀		
7. Себестоимость	С _у		

Таблица 3.10

Наименование показателя	Формула исчисления	Значения	
		исходные	проектируемые
1. Вооруженность труда	$ВТ=K_T/Ч_n$		
2. Электровооруженность	$ВЭ=F_C/Ч_n$		
3. Производительность труда	$П_T=A/Ч_0$		
4. Повышение или снижение производительности труда	$\frac{П_{T0}-П_{T1}}{П_{T0}} \times 100$		
5. Срок окупаемости	$T_{OK}=1/E_p$		
6. Коэффициент эффективности дополнительных капиталовложений	$E_p = \frac{\Delta T}{K_1 - K_0}$		

Глава 6. Специальный вопрос (эргономика, информационная безопасность, экология, охрана труда)

Выводы и предложения

Рекомендуется сделать выводы по проекту, определить пути его внедрения и направления дальнейшего совершенствования АИС.

Список литературы

Порядок построения списка определяется самим дипломником. Наиболее распространенными способами расположения литературы в списке литературы является алфавитный, систематический и в порядке упоминания в тексте.

Список должен содержать описание только тех источников, которые использованы при написании дипломного проекта.

В дипломном проекте на приведенные литературные источники обязательно должны быть ссылки в форме номера, заключенного в прямоугольные кавычки.

Приложения

В приложении обязательно должна быть распечатка на исходном языке программирования отлаженных основных расчетных модулей или адаптированных программных средств, использованных в работе.

3.3. Содержание разделов структуры проекта по проектированию и созданию программных средств обработки экономической информации

Данное направление дипломного проектирования нацелено на разработку и создание программных средств обработки экономической информации. Особенностью направления является его ориентация на поддержку процессов управления экономическим объектом.

Задачи решаемые в рамках направления:

- Оптимизация показателей деятельности экономического объекта на основе экономико-математических моделей, имитационного моделирования и т.д.;
- Планирование показателей деятельности экономического объекта, на основе статистических прогнозных моделей;
- Анализ показателей деятельности экономического объекта на основании существующих методик.

Структура и содержание первой, второй, пятой и шестой глав в дипломных проектах данного направления аналогичны соответствующим главам дипломных проектов направления «Проектирование и создание АИС» (см. предыдущий раздел).

Структура и содержание проектной части изложены ниже.

Глава 3. Проектирование ПО

3.1. Обоснование и выбор подхода к проектированию ПО

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.

Графические (визуальные) модели представляют собой средства для визуализации, описания, проектирования и документирования архитектуры системы. Разработка модели системы ПО в такой же мере необходима, как и наличие проекта при строительстве большого здания. Это утверждение справедливо как в случае разработки новой системы, так и при адаптации типовых продуктов. Хорошие модели являются основой взаимодействия участников проекта и гарантируют корректность архитектуры.

Состав моделей, используемых в каждом конкретном проекте, и степень их детальности в общем случае зависят от следующих факторов:

- сложности проектируемой системы;
- необходимой полноты ее описания;
- знаний и навыков участников проекта;

- времени, отведенного на проектирование.

Визуальное моделирование оказало большое влияние на развитие ТС ПО вообще и CASE-средств в частности. 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.

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.

3.2. Методы структурного анализа и проектирования ПО

Нисходящее проектирование

Метод нисходящего проектирования предполагает последовательное разложение общей функции обработки данных на простые функциональные элементы ("сверху-вниз").

В результате строится иерархическая схема, отражающая состав и взаимоподчиненность отдельных функций, которая носит название функциональная структура алгоритма (ФСА) приложения.

Последовательность действий по разработке функциональной структуры алгоритма приложения:

- определяются цели автоматизации предметной области и их иерархия (*цель-подцель*);
- устанавливается состав приложений (задач обработки), обеспечивающих реализацию поставленных целей;
- уточняется характер взаимосвязи приложений и их основные характеристики (информация для решения задач, время и периодичность решения, условия выполнения и др.);
- определяются необходимые для решения задач функции обработки данных;
- выполняется декомпозиция функций обработки до необходимой структурной сложности, реализуемой предполагаемым инструментарием.

Подобная структура приложения (рис. 3.10) отражает наиболее важное - состав и взаимосвязь функций обработки информации для реализа-

ции приложений, хотя и не раскрывает логику выполнения каждой отдельной функции, условия или периодичность их вызовов.

Разложение должно носить строго функциональный характер, т.е. отдельный элемент ФСА описывает законченную содержательную функцию обработки информации, которая предполагает определенный способ реализации на программном уровне.

Функции ввода-вывода информации рекомендуется отделять от функций вычислительной или логической обработки данных.

По частоте использования функции делятся на:

- однократно выполняемые;
- повторяющиеся.

Степень детализации функций может быть различной, но иерархическая схема должна давать представление о составе и структуре взаимосвязанных функций и общем алгоритме обработки данных. Широко используемые функции приобретают ранг стандартных (встроенных) функций при проектировании внутренней структуры программного продукта.

Например. Некоторые функции, например Φ_2 , далее неразложимы на составляющие: они предполагают непосредственную программную реализацию.

Другие функции, например Φ_1 , Φ_m , могут быть представлены в виде структурно объединения более простых функций, например Φ_{11} , Φ_{12} и т.д. Для всех функций-компонентов осуществляется самостоятельная программная реализация; составные функции (типа Φ_1 , Φ_m) реализуются как программные модули, управляющие функциями-компонентами. например, в виде программ-меню.

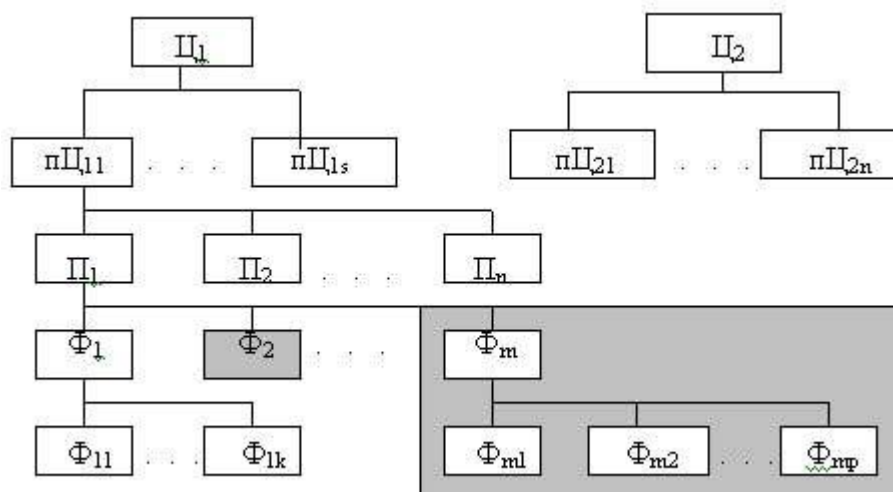


Рис.3.10. Функциональная структура приложения: Ц - цель; пЦ - подцель; П - приложение; Ф - функция

Модульное программирование

Модульное программирование основано на понятии модуля - логически взаимосвязанной совокупности функциональных элементов, оформленных в виде отдельных программных модулей.

Модуль характеризуют:

- один вход и один выход - на входе программный модуль получает определенный набор исходных данных, выполняет содержательную обработку и возвращает один набор результатных данных, т.е. реализуется стандартный принцип IPO (Input - Process - Output) - вход-процесс-выход;
- функциональная завершенность - модуль выполняет перечень регламентированных операций для реализации каждой отдельной функции в полном составе, достаточных для завершения начатой обработки;
- логическая независимость - результат работы программного модуля зависит только от исходных данных, но не зависит от работы других модулей;
- слабые информационные связи с другими программными модулями - обмен информацией между модулями должен быть по возможности минимизирован;
- обозримый по размеру и сложности программный элемент.

Таким образом, модули содержат определение доступных для обработки данных, операции обработки данных, схемы взаимосвязи с другими модулями.

Каждый модуль состоит из спецификации и тела. Спецификации определяют правила использования модуля, а тело - способ реализации процесса обработки.

Принципы модульного программирования программных продуктов во многом сходны с принципами нисходящего проектирования. Сначала определяются состав и подчиненность функций, а затем - набор программных модулей, реализующих эти функции.

Однотипные функции реализуются одними и теми же модулями. Функция верхнего уровня обеспечивается главным модулем; он управляет выполнением нижестоящих функций, которым соответствуют подчиненные модули.

При определении набора модулей, реализующих функции конкретного алгоритма, необходимо учитывать следующее:

- каждый модуль вызывается на выполнение вышестоящим модулем и, закончив работу, возвращает управление вызвавшему его модулю;
- принятие основных решений в алгоритме выносится на макси-

мально "высокий" по иерархии уровень;

- для использования одной и той же функции в разных местах алгоритма создается один модуль, который вызывается на выполнение по мере необходимости. В результате дальнейшей детализации алгоритма создается функционально-модульная схема (ФМС) алгоритма приложения, которая является основой для программирования (рис. 3.11).

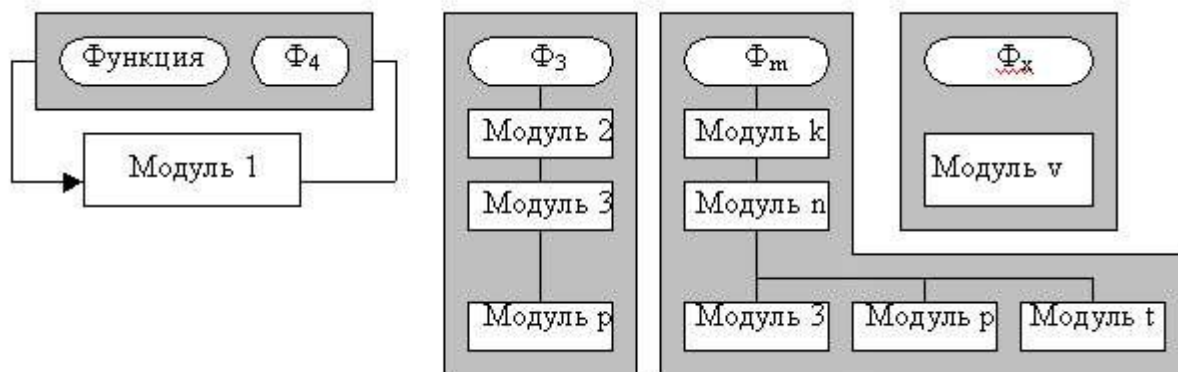


Рис.3.11.Функционально-модульная структура приложения

Например. Некоторые функции могут выполняться с помощью одного и того же программного модуля (например, функции Ф1 и Ф2).

- Функция Ф3 реализуется в виде последовательности выполнения программных модулей.
- Функция Фm реализуется с помощью иерархии связанных модулей.
- Модуль n управляет выбором на выполнение подчиненных модулей.
- Функция Фx реализуется одним программным модулем.

Состав и вид программных модулей, их назначение и характер использования в программе в значительной степени определяются инструментальными средствами. Например, применительно к средствам СУБД отдельными модулями могут быть:

- экранные формы ввода и/или редактирования информации базы данных;
- отчеты генератора отчетов;
- макросы;
- стандартные процедуры обработки информации;
- меню, обеспечивающее выбор функции обработки и др.

Алгоритмы большой сложности обычно представляются с помощью схем двух видов:

- обобщенной схемы алгоритма - раскрывает общий принцип функционирования алгоритма и основные логические связи меж-

- ду отдельными модулями на уровне обработки информации (ввод и редактирование данных, вычисления, печать результатов и т.п.);
- детальной схемы алгоритма представляет содержание каждого элемента обобщенной схемы с использованием управляющих структур в блок-схемах алгоритма, псевдокода либо алгоритмических языков высокого уровня.

Структурное программирование

Структурное программирование основано на модульной структуре программного продукта и типовых управляющих структурах алгоритмов обработки данных различных программных модулей.

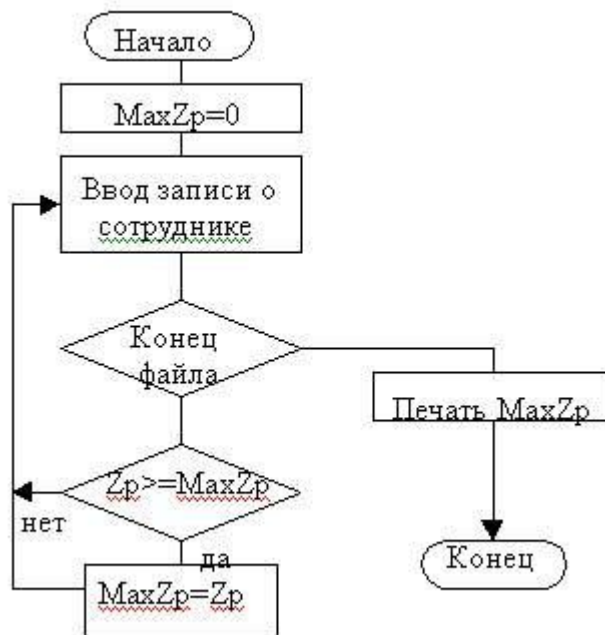


Рис 3.12. Блок-схема алгоритма поиска в базе данных.

В любой типовой структуре блок, кроме условного, имеет только один вход и выход, безусловный переход на блок с нарушением иерархии запрещен (оператор типа GoTo в структурном программировании не используется). Виды основных управляющих структур алгоритма приведены в табл. 3.11.

Например. Алгоритм поиска в базе данных сведений о максимальном окладе сотрудников, рис 3.12.

Таблица 3.11

Управляющие структуры алгоритмов

Типы управляющей структуры	Применение управляющей структуры
Последовательность Блок 1 Блок 2	Последовательность включает фиксированный перечень блоков (операторов). Каждый очередной блок обрабатывается после завершения пре-

Конец	дыдущего без дополнительных условий. Для изменения порядка обработки блоков редактируется последовательность выполняемых операций.
Альтернатива (условие выбора) Начало Да Условие Нет Альтернатива1 Альтернатива2 Конец	В блоке Условие содержится условие выбора альтернативы обработки. Каждая альтернатива выполняется 1 раз; выполнение одной из двух альтернатив - обязательно. Развитие данного типа структуры является множественная альтернатива, когда последовательно проверяются условия выполнения определенных альтернатив. Если очередное условие истинно, обрабатывается соответствующая ему альтернатива, после чего происходит выход. В противном случае - переход к проверке условия следующей альтернативы. Если ни одно из условий не выполнилось, происходит выход.
Цикл ("пока") Начало Условие Нет Да Тело цикла Конец	В блоке Условие задается условие тела цикла - определенной обработки. Если условие не выполняется, цикл прерывается и осуществляется выход. Условие может содержать счетчик повторений тела цикла либо логическое условие. Тело цикла - произвольная последовательность операций.

3.3. Методы объектно-ориентированного анализа и проектирования ПО

Концептуальной основой объектно-ориентированного анализа и проектирования ПО (ООАП) является объектная модель. Ее основные принципы (абстрагирование, инкапсуляция, модульность и иерархия) и понятия (объект, класс, атрибут, операция, интерфейс и др.) наиболее четко сформулированы Гради Бучем.

Большинство современных методов ООАП основаны на использовании языка UML. Унифицированный язык моделирования UML (Unified Modeling Language, представляет собой язык для определения, представления, проектирования и документирования программных систем, организационно-экономических систем, технических систем и других систем различной природы. UML содержит стандартный набор диаграмм и нотаций самых разнообразных видов.

Главными в разработке UML были следующие цели:

- предоставить пользователям готовый к использованию выразительный язык визуального моделирования, позволяющий им разрабатывать осмысленные модели и обмениваться ими;

- предусмотреть механизмы расширяемости и специализации для расширения базовых концепций;
- обеспечить независимость от конкретных языков программирования и процессов разработки.
- обеспечить формальную основу для понимания этого языка моделирования (язык должен быть одновременно точным и доступным для понимания, без лишнего формализма);
- стимулировать рост рынка объектно-ориентированных инструментальных средств;
- интегрировать лучший практический опыт.

Стандарт UML может содержать следующий набор диаграмм:

Структурные (structural) модели:

- диаграммы классов (class diagrams) – для моделирования статической структуры классов системы и связей между ними;
- диаграммы компонентов (component diagrams) - для моделирования иерархии компонентов (подсистем) системы;
- диаграммы размещения (deployment diagrams) - для моделирования физической архитектуры системы.

Модели поведения (behavioral):

- диаграммы вариантов использования (use case diagrams) - для моделирования функциональных требований к системе (в виде сценариев взаимодействия пользователей с системой);
- диаграммы взаимодействия (interaction diagrams):
- диаграммы последовательности (sequence diagrams) и кооперативные диаграммы (collaboration diagrams) - для моделирования процесса обмена сообщениями между объектами;
- диаграммы состояний (statechart diagrams) - для моделирования поведения объектов системы при переходе из одного состояния в другое;
- диаграммы деятельности (activity diagrams) - для моделирования поведения системы в рамках различных вариантов использования, или потоков управления.

Диаграммы вариантов использования показывают взаимодействия между вариантами использования и действующими лицами, отражая функциональные требования к системе с точки зрения пользователя. Цель построения диаграмм вариантов использования - это документирование функциональных требований в самом общем виде, поэтому они должны быть предельно простыми.

Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант ис-

пользования описывает типичное взаимодействие между пользователем и системой и отражает представление о поведении системы с точки зрения пользователя. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать, или целей, которые он преследует по отношению к разрабатываемой системе.

Диаграмма вариантов использования является самым общим представлением функциональных требований к системе. Для последующего проектирования системы требуются более конкретные детали, которые описываются в документе, называемом "сценарием варианта использования" или "поток событий" (flow of events). Основной поток событий описывает нормальный ход событий (при отсутствии ошибок). Альтернативные потоки описывают отклонения от нормального хода событий (ошибочные ситуации) и их обработку.

Достоинства модели вариантов использования заключаются в том, что она:

- определяет пользователей и границы системы;
- определяет системный интерфейс;
- удобна для общения пользователей с разработчиками;
- используется для написания тестов;
- является основой для написания пользовательской документации;
- хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).

Диаграммы взаимодействия описывают поведение взаимодействующих групп объектов (в рамках варианта использования или некоторой операции класса). Как правило, диаграмма взаимодействия охватывает поведение объектов в рамках только одного потока событий варианта использования. На такой диаграмме отображается ряд объектов и те сообщения, которыми они обмениваются между собой. Существует два вида диаграмм взаимодействия: диаграммы последовательности и кооперативные диаграммы.

Диаграммы последовательности отражают временную последовательность событий, происходящих в рамках варианта использования, а кооперативные диаграммы концентрируют внимание на связях между объектами.

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в

процессе анализа) или элементы программной системы (в процессах проектирования и реализации).

Диаграммы состояний определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий. Диаграммы состояний не надо создавать для каждого класса, они применяются только в сложных случаях. Если объект класса может существовать в нескольких состояниях и в каждом из них ведет себя по-разному, для него может потребоваться такая диаграмма.

Диаграммы деятельности, в отличие от большинства других средств UML, заимствуют идеи из нескольких различных методов, в частности, метода моделирования состояний SDL и сетей Петри. Эти диаграммы особенно полезны в описании поведения, включающего большое количество параллельных процессов. Диаграммы деятельности являются также полезными при параллельном программировании, поскольку можно графически изобразить все ветви и определить, когда их необходимо синхронизировать.

Диаграммы деятельности можно применять для описания потоков событий в вариантах использования. С помощью текстового описания можно достаточно подробно рассказать о потоке событий, но в сложных и запутанных потоках с множеством альтернативных ветвей будет трудно понять логику событий. Диаграммы деятельности предоставляют ту же информацию, что и текстовое описание потока событий, но в наглядной графической форме.

Диаграммы компонентов моделируют физический уровень системы. На них изображаются компоненты ПО и связи между ними. На такой диаграмме обычно выделяют два типа компонентов: исполняемые компоненты и библиотеки кода.

Каждый класс модели (или подсистема) преобразуется в компонент исходного кода. Между отдельными компонентами изображают зависимости, соответствующие зависимостям на этапе компиляции или выполнения программы.

Диаграммы компонентов применяются теми участниками проекта, кто отвечает за компиляцию и сборку системы. Они нужны там, где начинается генерация кода.

Диаграмма размещения отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать размещение объектов и компонентов в распределенной системе.

Диаграмма размещения показывает физическое расположение сети и местонахождение в ней различных компонентов. Ее основными элемента-

ми являются узел (вычислительный ресурс) и соединение - канал взаимодействия узлов (сеть).

Диаграмма размещения используется менеджером проекта, пользователями, архитектором системы и эксплуатационным персоналом, чтобы понять физическое размещение системы и расположение ее отдельных подсистем.

UML обладает механизмами расширения, предназначенными для того, чтобы разработчики могли адаптировать язык моделирования к своим конкретным нуждам, не меняя при этом его метамодель. Наличие механизмов расширения принципиально отличает UML от таких средств моделирования, как IDEF0, IDEF1X, IDEF3, DFD и ERM. Перечисленные языки моделирования можно определить как сильно типизированные (по аналогии с языками программирования), поскольку они не допускают произвольной интерпретации семантики элементов моделей. UML, допуская такую интерпретацию (в основном за счет стереотипов), является слабо типизированным языком. К его механизмам расширения относятся:

- стереотипы;
- тегированные (именованные) значения;
- ограничения.

Стереотип - это новый тип элемента модели, который определяется на основе уже существующего элемента. Стереотипы расширяют нотацию модели и могут применяться к любым элементам модели. Стереотипы классов - это механизм, позволяющий разделять классы на категории. Разработчики ПО могут создавать свои собственные наборы стереотипов, формируя тем самым специализированные подмножества UML (например, для описания бизнес-процессов, Web-приложений, баз данных и т.д.). Такие подмножества (наборы стереотипов) в стандарте языка UML носят название профилей языка.

Именованное значение - это пара строк "тег = значение", или "имя = содержимое", в которых хранится дополнительная информация о каком-либо элементе системы, например, время создания, статус разработки или тестирования, время окончания работы над ним и т.п.

Ограничение - это семантическое ограничение, имеющее вид текстового выражения на естественном или формальном языке (OCL - Object Constraint Language), которое невозможно выразить с помощью нотации UML.

Сопоставление и взаимосвязь структурного и объектно-ориентированного подходов

Г.Буч сформулировал главное достоинство объектно-ориентированного подхода (ООП) следующим образом: объектно-ориентированные системы более открыты и легче поддаются внесению

изменений, поскольку их конструкция базируется на устойчивых формах. Это дает возможность системе развиваться постепенно и не приводит к полной ее переработке даже в случае существенных изменений исходных требований.

Г.Буч отметил также ряд следующих преимуществ ООП:

- объектная декомпозиция дает возможность создавать программные системы меньшего размера путем использования общих механизмов, обеспечивающих необходимую экономию выразительных средств. Использование ООП существенно повышает уровень унификации разработки и пригодность для повторного использования не только ПО, но и проектов, что в конце концов ведет к сборочному созданию ПО. Системы зачастую получаются более компактными, чем их не объектно-ориентированные эквиваленты, что означает не только уменьшение объема программного кода, но и удешевление проекта за счет использования предыдущих разработок;
- объектная декомпозиция уменьшает риск создания сложных систем ПО, так как она предполагает эволюционный путь развития системы на базе относительно небольших подсистем. Процесс интеграции системы растягивается на все время разработки, а не превращается в единовременное событие;
- объектная модель вполне естественна, поскольку в первую очередь ориентирована на человеческое восприятие мира, а не на компьютерную реализацию;
- объектная модель позволяет в полной мере использовать выразительные возможности объектных и объектно-ориентированных языков программирования.

К недостаткам ООП относятся некоторое снижение производительности функционирования ПО (которое, однако, по мере роста производительности компьютеров становится все менее заметным) и высокие начальные затраты. Объектная декомпозиция существенно отличается от функциональной, поэтому переход на новую технологию связан как с преодолением психологических трудностей, так и дополнительными финансовыми затратами. При переходе от структурного подхода к объектному, как при всякой смене технологии, необходимо вкладывать деньги в приобретение новых инструментальных средств. Здесь следует учесть расходы на обучение методу, инструментальным средствам и языку программирования. Для некоторых организаций эти обстоятельства могут стать серьезными препятствиями.

Объектно-ориентированный подход не дает немедленной отдачи. Эффект от его применения начинает сказываться после разработки двух-трех проектов и накопления повторно используемых компонентов, отра-

жающих типовые проектные решения в данной области. Переход организации на объектно-ориентированную технологию - это смена мировоззрения, а не просто изучение новых CASE-средств и языков программирования.

Таким образом, структурный подход по-прежнему сохраняет свою значимость и достаточно широко используется на практике. На примере языка UML хорошо видно, что его авторы заимствовали то рациональное, что можно было взять из структурного подхода: элементы функциональной декомпозиции в диаграммах вариантов использования, диаграммы состояний, диаграммы деятельности и др. Очевидно, что в конкретном проекте сложной системы невозможно обойтись только одним способом декомпозиции. Можно начать декомпозицию каким-либо одним способом, а затем, используя полученные результаты, попытаться рассмотреть систему с другой точки зрения.

Основой взаимосвязи между структурным и объектно-ориентированным подходами является общность ряда категорий и понятий обоих подходов (процесс и вариант использования, сущность и класс и др.). Эта взаимосвязь может проявляться в различных формах. Так, одним из возможных вариантов является использование структурного анализа как основы для объектно-ориентированного проектирования. При этом структурный анализ следует прекращать, как только структурные модели начнут отражать не только деятельность организации (бизнес-процессы), а и систему ПО. После выполнения структурного анализа можно различными способами приступить к определению классов и объектов. Так, если взять какую-либо отдельную диаграмму потоков данных, то кандидатами в классы могут быть элементы структур данных.

Другой формой проявления взаимосвязи можно считать интеграцию объектной и реляционной технологий. Реляционные СУБД являются на сегодняшний день основным средством реализации крупномасштабных баз данных и хранилищ данных. Причины этого достаточно очевидны: реляционная технология используется достаточно долго, освоена огромным количеством пользователей и разработчиков, стала промышленным стандартом, в нее вложены значительные средства и создано множество корпоративных БД в самых различных отраслях, реляционная модель проста и имеет строгое математическое основание; существует большое разнообразие промышленных средств проектирования, реализации и эксплуатации реляционных БД. Вследствие этого реляционные БД в основном используются для хранения и поиска объектов в так называемых объектно-реляционных системах.

Взаимосвязь между структурным и объектно-ориентированным подходами достаточно четко просматривается в различных ТС ПО.

3.4. Проектирование бизнес процессов, спецификаций и требований

Моделирование бизнес-процессов является важной составной частью проектов по созданию ПО. Отсутствие таких моделей является одной из главных причин неудач многих проектов.

Назначением будущего ПО является, в первую очередь, решение проблем бизнеса. Требования к ПО формируются на основе бизнес-модели, а критерии проектирования систем прежде всего основываются на наиболее полном их удовлетворении.

Модели бизнес-процессов являются не просто промежуточным результатом, используемым консультантом для выработки каких-либо рекомендаций и заключений. Они представляют собой самостоятельный результат, имеющий большое практическое значение.

На сегодняшний день в моделировании бизнес-процессов преобладает процессный подход. Его основной принцип заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой. Модель, основанная на организационно-штатной структуре, может продемонстрировать лишь хаос, царящий в организации (о котором в принципе руководству и так известно, иначе оно бы не инициировало соответствующие работы), на ее основе можно только внести предложения об изменении этой структуры. С другой стороны, модель, основанная на бизнес-процессах, содержит в себе и организационно-штатную структуру предприятия.

Процессный подход может использовать любые из перечисленных выше средств моделирования. Однако, в настоящее время наблюдается тенденция интеграции разнообразных методов моделирования и анализа систем, проявляющаяся в форме создания интегрированных средств моделирования. Одним из таких средств является продукт, носящий название ARIS - Architecture of Integrated Information System, разработанный германской фирмой IDS Scheer.

Система ARIS представляет собой комплекс средств анализа и моделирования деятельности предприятия. Ее методическую основу составляет совокупность различных методов моделирования, отражающих разные взгляды на исследуемую систему. Одна и та же модель может разрабатываться с использованием нескольких методов, что позволяет использовать ARIS специалистам с различными теоретическими знаниями и настраивать его на работу с системами, имеющими свою специфику.

ARIS поддерживает четыре типа моделей, отражающих различные аспекты исследуемой системы:

- организационные модели, представляющие структуру системы - иерархию организационных подразделений, должностей и конкретных лиц, связи между ними, а также территориальную привязку струк-

турных подразделений;

- функциональные модели, содержащие иерархию целей, стоящих перед аппаратом управления, с совокупностью деревьев функций, необходимых для достижения поставленных целей;
- информационные модели, отражающие структуру информации, необходимой для реализации всей совокупности функций системы;
- модели управления, представляющие комплексный взгляд на реализацию бизнес-процессов в рамках системы.

Для построения перечисленных типов моделей используются как собственные методы моделирования ARIS, так и различные известные методы и языки моделирования - ERM, UML, OMT и др.

В процессе моделирования каждый аспект деятельности предприятия сначала рассматривается отдельно, а после детальной проработки всех аспектов строится интегрированная модель, отражающая все связи между различными аспектами.

Ряд современных методов моделирования бизнес-процессов основан на использовании языка UML. Хотя UML изначально предназначался для моделирования систем ПО, его использование в другой области стало возможным благодаря наличию в UML механизмов расширения (стереотипов).

Среди таких методов наиболее известными являются метод Ericsson-Penker и метод, реализованный в технологии Rational Unified Process (RUP).

Метод Ericsson- Penker представляет интерес прежде всего в связи с попыткой применения UML в рамках процессного подхода к моделированию бизнес-процессов. Авторы метода создали свой профиль UML для моделирования бизнес-процессов, введя набор стереотипов, описывающих процессы, ресурсы, правила и цели деятельности организации. Метод использует четыре основные категории бизнес-модели:

- Ресурсы - различные объекты, используемые или участвующие в бизнес-процессах (люди, материалы, информация или продукты);
- Процессы - виды деятельности, изменяющие состояние ресурсов в соответствии с бизнес-правилами;
- Цели - назначение бизнес-процессов Цели могут быть разбиты на подцели и соотнесены с отдельными процессами;
- Бизнес-правила - условия или ограничения выполнения процессов (функциональные, поведенческие или структурные).

Основной диаграммой UML, используемой в данном методе, является диаграмма деятельности. Метод Eriksson-Penker представляет процесс в виде деятельности со стереотипом "process" (основой данного представления является расширение метода IDEF0). Полная бизнес-модель включает

множество представлений, подобных представлениям архитектуры ПО. Каждое представление выражено в одной или более диаграммах UML. Диаграммы могут иметь различные типы и изображать процессы, правила, цели и ресурсы во взаимодействиях друг с другом. Метод использует четыре различных представления бизнес-модели:

- концептуальное представление - структура целей и проблем;
- представление процессов - взаимодействие между процессами и ресурсами (в виде набора диаграмм деятельности);
- структурное представление - структура организации и ресурсов (в виде диаграмм классов);
- представление поведения - поведение отдельных ресурсов и детализация процессов (в виде диаграмм деятельности, состояний и взаимодействия).

Методика моделирования RUP предусматривает построение двух моделей:

- модели бизнес-процессов (Business Use Case Model);
- модели бизнес-анализа (Business Analysis Model).

Модель бизнес-процессов представляет собой расширение модели вариантов использования UML за счет введения набора стереотипов - Business Actor (стереотип действующего лица) и Business Use Case (стереотип варианта использования). Business Actor (действующее лицо бизнес-процессов) - это некоторая роль, внешняя по отношению к бизнес-процессам организации. Business Use Case (вариант использования с точки зрения бизнес-процессов) определяется как описание последовательности действий (потока событий) в рамках некоторого бизнес-процесса, приносящей ощутимый результат конкретному действующему лицу. Это определение подобно общему определению бизнес-процесса, но имеет более точный смысл. В терминах объектной модели Business Use Case представляет собой класс, объектами которого являются конкретные потоки событий в рамках описываемого бизнес-процесса.

Описание Business Use Case может сопровождаться целью процесса, которая, так же, как и в методе Eriksson-Penker, моделируется с помощью класса со стереотипом "goal", а дерево целей изображается в виде диаграммы классов.

Для каждого Business Use Case строится модель бизнес-анализа - объектная модель, описывающая реализацию бизнес-процесса в терминах взаимодействующих объектов (бизнес-объектов - Business Object), принадлежащих к двум классам - Business Worker и Business Entity. Business Worker (исполнитель) - класс, представляющий собой абстракцию исполнителя, выполняющего некоторые действия в рамках бизнес-процесса. Исполнители взаимодействуют между собой и манипулируют различными

сущностями, участвуя в реализациях сценариев Business Use Case. Business Entity (сущность) является объектом различных действий со стороны исполнителей.

Кроме диаграммы данных классов, модель бизнес-анализа может включать:

- Диаграммы последовательности (и кооперативные диаграммы), описывающие сценарии Business Use Case в виде последовательности обмена сообщениями между объектами-действующими лицами и объектами-исполнителями. Такие диаграммы помогают явно определить в модели обязанности каждого исполнителя в виде набора его операций.
- Диаграммы деятельности, описывающие взаимосвязи между сценариями одного или различных Business Use Case.
- Диаграммы состояний, описывающие поведение отдельных бизнес-объектов.

Методика моделирования Rational Unified Process обладает следующими достоинствами:

- модель бизнес-процессов строится вокруг участников процессов (заинтересованных лиц) и их целей, помогая выявить все потребности клиентов организации. Такой подход в наибольшей степени применим для организаций, работающих в сфере оказания услуг (торговые организации, банки, страховые компании и т.д.);
- моделирование на основе вариантов использования способствует хорошему пониманию бизнес-модели со стороны заказчиков.

Однако следует отметить, что при моделировании деятельности крупной организации, занимающейся как производством продукции, так и оказанием услуг, необходимо применять различные методики моделирования, поскольку для моделирования производственных процессов более предпочтительным является процессный подход (например, метод Eriksson-Penker).

Спецификация требований к ПО является составной частью процесса управления требованиями. Выявленные в результате применения перечисленных методов требования к ПО оформляются в виде ряда документов и моделей. Так, в технологии RUP функциональные требования к системе моделируются и документируются с помощью вариантов использования. Стилль их написания зависит от масштаба, количества участников и критичности проекта.

Спецификация требований в RUP не требует обязательного моделирования бизнес-процессов организации, для которых создается ПО, однако, наличие бизнес-моделей существенно упрощает построение системной модели вариантов использования.

3.5. Построение модели ПО

Целью анализа требований является трансформация функциональных требований к ПО в предварительный системный проект и создание стабильной основы архитектуры системы. В процессе проектирования системный проект "погружается" в среду реализации с учетом всех нефункциональных требований.

Все современные ТС ПО реализуют ту или иную методику анализа и проектирования ПО. Одна из типичных методик ООАП реализована в технологии RUP. Согласно этой методике, объектно-ориентированный анализ включает два вида деятельности: архитектурный анализ и анализ вариантов использования. Архитектурный анализ выполняется архитектором системы и включает в себя:

- утверждение общих стандартов (соглашений) моделирования и документирования системы;
- предварительное выявление архитектурных механизмов (надёжности, безопасности и т.д.);
- формирование набора основных абстракций предметной области (классов анализа);
- формирование начального представления архитектурных уровней.
- анализ вариантов использования выполняется проектировщиками и включает в себя:
 - идентификацию классов, участвующих в реализации потоков событий варианта использования;
 - распределение поведения, реализуемого вариантом использования, между классами (определение обязанностей классов);
 - определение атрибутов и ассоциаций классов.

В потоках событий варианта использования выявляются классы трех типов:

- Граничные классы (Boundary) - служат посредниками при взаимодействии внешних объектов с системой. Типы граничных классов: пользовательский интерфейс (обмен информацией с пользователем, без деталей интерфейса - кнопок, списков, окон), системный интерфейс и аппаратный интерфейс (используемые протоколы, без деталей их реализации).

- Классы-сущности (Entity) - представляют собой основные абстракции (понятия) разрабатываемой системы, рассматриваемые в рамках конкретного варианта использования.

- Управляющие классы (Control) - обеспечивают координацию поведения объектов в системе. Примеры управляющих классов: менеджер транзакций, координатор ресурсов, обработчик ошибок.

Классы анализа отражают функциональные требования к системе и моделируют объекты предметной области. Совокупность классов анализа

представляет собой начальную концептуальную модель системы

Наиболее важной частью объектно-ориентированного анализа является распределение обязанностей между классами (в виде операций классов). Оно выполняется с помощью диаграмм взаимодействия. При построении диаграмм взаимодействия возникают проблемы правильного распределения обязанностей между классами.

Атрибуты классов анализа определяются, исходя из знаний о предметной области и требований к системе. Связи между классами (ассоциации) определяются на основе анализа кооперативных диаграмм, затем анализируются и уточняются.

Целью объектно-ориентированного проектирования является адаптация предварительного системного проекта (набора классов "анализа"), составляющего стабильную основу архитектуры системы, к среде реализации с учетом всех нефункциональных требований.

Объектно-ориентированное проектирование включает два вида деятельности:

- проектирование архитектуры системы;
- проектирование элементов системы.

Проектирование архитектуры системы выполняется архитектором системы и включает в себя:

- идентификацию архитектурных решений и механизмов, необходимых для проектирования системы;
- анализ взаимодействий между классами анализа, выявление подсистем и интерфейсов;
- формирование архитектурных уровней;
- проектирование конфигурации системы.

Проектирование элементов системы включает в себя:

- проектирование классов (детализация классов, уточнение операций и атрибутов, моделирование состояний, уточнение связей между классами);
- проектирование баз данных.

Глава 4. Реализация ПО

Данный раздел посвящен описанию основных этапов реализации проектных решений изложенных в предыдущей главе.

4.1. Обоснование и выбор языка и среды разработки.

4.2. Реализация программных модулей.

4.3. Реализация интерфейсов.

4.4. Реализация контрольного примера.

3.4. Содержание разделов структуры проекта по модификации, адаптации, внедрению и сопровождению тиражируемых вариантов АИС и программных средств обработки экономической информации

Глава 3. Моделирование направлений адаптации и внедрения программных средств и АИС в предметную область

Данное направление дипломного проектирования связано с адаптацией и внедрением готовых (тиражируемых) АИС в предметную область.

Дипломные проекты могут выполняться по двум направлениям:

- модификация АИС;
- внедрение АИС.

Возможна также совместная разработка по двум направлениям, в случае если масштабность по каждому из направлений недостаточна для выполнения дипломного проекта.

Реализуемые задачи, как правило, подразделяются по функциональному признаку в экономической системе, например:

- бухгалтерский учет;
- планирование;
- управленческий учет;
- управление трудовыми ресурсами;
- управление маркетингом;
- аналитическая деятельность;
- и т.д.

Экономическим объектам свойственны динамичность и развитие, что непосредственно влияет на состояние ИС. Поэтому на стадии эксплуатации ИС усиливаются факторы, доказывающие необходимость последующей модернизации. Среди них:

- изменения на объекте управления и во внешней среде (дрейф параметров предметной области);
- изменение состава рабочей нагрузки вычислительной системы, замена оборудования, рост объема файлов;
- накопление опыта работы с АИС,
- обнаружение проектных ошибок.

Задачи модификации АИС обычно рассматриваются как неперспективные и нежелательные. Эта работа считается очень простой в сравнении с проектированием АИС, она ассоциируется с исправлением проектных ошибок, сделанных другими. Вместе с тем модифицируемая система обычно плохо документирована, попытки улучшения прикладных программ иногда кончаются ничем. В итоге довольно быстро наступает момент, когда интерес к системе теряется и начинается новая разработка. Од-

нако планомерная модификация базы данных и других компонентов АИС позволяет поддерживать в требуемых границах ее технические и эксплуатационные характеристики, отсрочить момент морального старения системы.

На стадии эксплуатации системы в отсутствие специальных мероприятий по модернизации ИС ухудшаются ее эксплуатационные показатели, например, снижается пропускная способность. Происходит также ухудшение соответствия между параметрами предметной области и параметрами БД.

В процессе эксплуатации АИС производится слежение за изменением параметров АИС и предметной области. Для этого используются, например:

- информация об изменениях в системе документооборота и структуре отдельных документов,
- данные об изменениях в составе решаемых экономических задач, системе экономических показателей и методах их расчета,
- характеристики потока запросов к БД,
- оценки пользователей о качестве получаемой информации,
- информация системной мониторинговой программы или аналогичных средств, работающих в составе применяемых операционных систем и СУБД, сбор статистики о выполненных заданиях.

Должны также фиксироваться изменения количественных и качественных характеристик предметной области. В этой сфере могут происходить изменения в организационной структуре экономического объекта, составе параметров, характеризующих объект, методах их расчета. Изменения зачастую связаны с реконструкцией производства, выпуском новых изделий, освоением новых технологий, совершенствованием конструкторской документации. Может меняться состав организационных и технологических ограничений на объекте.

Сравнение результатов измерений с аналогичной информацией за прошлые периоды времени и отклонение текущих параметров функционирования АИС от нормативных могут дать основание для проведения модификации АИС. Анализ результатов наблюдений должен быть различным в зависимости от целей, которые предполагается достичь после проведения модификации. Первоначально должна быть поставлена цель модификации АИС и определено множество методов, ведущих к достижению требуемой цели. Собираемая и анализируемая информация должна лишь доказать (или опровергнуть) целесообразность применения конкретного метода модификации и позволить выработать его спецификацию

Цели модификации АИС можно разделить на шесть больших групп:

- исправление проектных ошибок,

- улучшение эксплуатационных характеристик АИС,
- адаптация к изменениям в предметной области,
- разработка нового приложения,
- обеспечение совместимости с другими ИС,
- перенос БД в новую аппаратно-программную среду.

Конкретные методы модификации АИС группируются по четырем направлениям

- реструктуризация БД,
- перепрограммирование прикладных задач,
- реорганизация БД,
- настройка вычислительной системы.

Данные табл.3.12 показывают, что автономные операции над базой данных (реструктуризация и реорганизация) не могут быть эффективно использованы для целей модификации без одновременного совершенствования вычислительной системы и программного обеспечения.

Таблица 3.12

Соответствие целей и методов модификации АИС

Цели модификации	Методы модификации АИС			
	Реструктуризация	Перепрограммирование прикладных задач	Реорганизация	Настройка вычислительной системы
Исправление проектных ошибок	+	+		+
Улучшение эксплуатационных характеристик	+	+	+	+
Адаптация к изменениям в предметной области	+	+		
Разработка нового приложения	+	+		+
Совместимость с другими АИС	+	+		+

Большинство процедур модификации АИС могут производиться без прекращения стадии эксплуатации. Однако необходим контроль всех компонентов АИС (базы данных, вычислительной системы, программных средств) после проведения каких-либо усовершенствований.

Структура и содержание первой, второй, пятой и шестой глав в дипломных проектах данного направления аналогичны соответствующим главам дипломных проектов направления «Проектирование и создание АИС» (см. предыдущий раздел).

Структура и содержание проектной части изложены ниже.

3.1. Декомпозиция комплекса задач предметной области

В данном разделе необходимо провести анализ задач по модификации и внедрению АИС. При этом необходимо наряду с анализом существующих методик и технологий решения задач обозначить направления совершенствования решений.

Детализация должна проводиться со степенью достаточной для дальнейшего определения формальных требований.

3.2. Анализ возможностей и функциональности предлагаемой системы по решению задач предметной области

Производится сопоставление требований изложенных в предыдущей главе и возможностей и функциональности тиражируемых вариантов АИС.

3.3. Модель модификации и адаптации системы

Описание методов, приемов и средств модификации и адаптации системы.

Глава 4. Модификация, адаптация, внедрение и сопровождение системы.

4.1. Реализация изменений системы.

4.2. Организация эксплуатации системы.

3.5. Содержание разделов структуры проекта по проектированию и разработке WEB-ресурсов экономической деятельности

Дипломные проекты данного направления могут выполняться по следующим вариантам:

1. Информационная поддержка существующего неэлектронного бизнеса. *Цели:* создание дополнительного рекламно-информационного канала, облегчение коммуникации с действующими и потенциальными партнерами, формирование положительного имиджа компании как предприятия, использующего современные технологии. *Концепция:* создание корпоративного сайта, содержащего информацию о компании, ее товарах, услугах, ценах, сервисе, лицензиях и т. п. В данном случае Интернет-составляющая бизнеса не получает дохода, а лишь помогает основному неэлектронному бизнесу. Однако за счет содержащейся информации можно сократить расходы на телефонные переговоры, диспетчерскую службу, распространение прайс-листов и т.д.

2. Организация продаж через Интернет товаров или услуг существующего неэлектронного бизнеса (частичный перенос неэлектронного бизнеса в Интернет) *Цели:* использование Интернета для рекламы и сбыта продукции, товаров, услуг. *Концепция:* создание корпоративного сайта, предоставляющего исчерпывающую информацию о компании, продукции;

обеспечение возможности оформления онлайн заказа; создание системы доставки. Здесь Интернет-составляющая бизнеса участвует в формировании выручки всего предприятия, обеспечивая увеличение общего объема сбыта.

3. Создание Интернет-компании реализующей товары и услуги исключительно через Интернет. *Цели:* реализация в Интернете полного бизнес-цикла ориентированного на получение прибыли от торгово-закупочной деятельности и/или от оказания услуг. *Концепция:* создание интерактивного сайта, обеспечивающего работу с клиентами (онлайн заказы, каталоги товаров и услуг, прайс-листы, и т.д.); создание системы поставок и складской системы; организация системы доставки заказов; подключение к системам приема платежей через Интернет.

В этом варианте компания должна сама покрывать все свои расходы и приносить прибыль владельцам.

4. Рекламная модель. *Цель:* сформировать на сайте проекта возможно более широкую или не столь обширную, но жестко сегментированную, аудиторию посетителей и продавать контакт с ней рекламодателям. *Концепция:* создание сайта, содержащего интересное и/или полезное целевой аудитории информационное наполнение (контент), предоставление посетителям разнообразных бесплатных сервисов, привлечение на сайт максимального количества посетителей всевозможными средствами, включая Интернет-рекламу, обычную (онлайн) рекламу, розыгрыши призов и т. п.; поиск и привлечение рекламодателей, готовых платить за контакт с аудиторией сайта.

Для успешного функционирования проекта, построенного на такой модели, доходы от рекламы должны, как минимум, покрывать все расходы компании. Это очень популярная модель сетевого предпринимательства, но она крайне неустойчива к колебаниям цен и спроса на рекламу в Интернете.

Структура и содержание первой, второй, пятой и шестой глав в дипломных проектах данного направления аналогичны соответствующим главам дипломных проектов направления «Проектирование и создание АИС» (см. предыдущий раздел).

Структура и содержание проектной части изложены ниже.

Глава 3. Проектирование Web-ресурса

3.1. Проектирование функциональности и структуры Web-ресурса

В зависимости от выбранного варианта использования Web-ресурса необходимо описать функциональность и структуру ресурса. При этом необходимо использовать методы структурного или объектного проектирования.

3.2. Проектирование БД Web-ресурса

Общие положения по проектированию БД изложены в соответствующей главе проектов направления «Проектирование и создание АИС».

3.3. Проектирование дизайна Web-ресурса.

Глава 4. Реализация Web-ресурса

Реализация Web-ресурса включает:

- 4.1. Обоснование и выбор платформы и средств реализации.
- 4.2. Реализация БД Web-ресурса.
- 4.3. Реализация дизайна Web-ресурса.
- 4.4. Размещение и сопровождение Web-ресурса.

4. ОФОРМЛЕНИЕ ДИПЛОМНОГО ПРОЕКТА

4.1. Требования и правила оформления текстового материала

Оформление дипломного проекта должно соответствовать определенным требованиям.

Материал дипломного проекта располагается в следующем порядке:

- Титульный лист (приложение 1);
- Задание на дипломный проект (приложение 2);
- Отзыв дипломного руководителя;
- Оглавление (содержание);
- Введение;
- Основная часть (Главы);
- Заключение;
- Список литературы;
- Приложение (приложения).

В содержании приводятся заголовки глав, разделов, граф, параграфов и т.д. с указанием страниц всех частей работы. При этом заголовки и их рубрикационные индексы должны быть приведены в строгом соответствии с текстом.

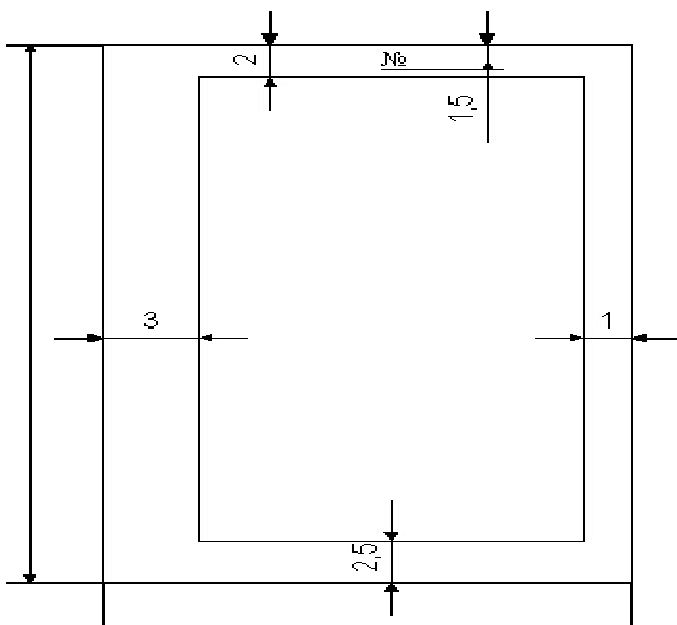


Рис 4.1 Оформление текста

4.2. Оформление дипломного проекта на компьютере

Наиболее предпочтительным является оформление дипломного проекта с использованием текстового процессора. В этом случае во время предзащиты студент, кроме дипломного проекта должен представить на кафедру дискету с дипломным проектом.

При оформлении дипломного проекта в текстовом редакторе следует соблюдать следующие параметры: выбранный шрифт должен быть четким и разборчивым (рекомендуется «Times New Roman»), размер шрифта-14, печать через 1,5 интервала. Основной объем работы 90-100 страниц.

Названия глав, параграфов, пунктов, подпунктов следует начинать с абзаца, их можно писать более крупным кеглем, чем текст. Допускается выделение интенсивностью (полужирный шрифт).

4.3. Правила оформления иллюстративного материала

Необходимым условием оформления дипломного проекта является иллюстративный материал, который может быть представлен в виде рисунков, схем, таблиц, графиков, диаграмм. Иллюстрации должны наглядно дополнять и подтверждать содержание текстового материала и отражать тему дипломного проекта. На каждую единицу иллюстративного материала должна быть хотя бы одна ссылка в тексте дипломного проекта.

В том случае, когда текст иллюстрируется таблицами, они оформля-

ются следующим образом. Таблицы следует размещать сразу после ссылки на них в тексте. Таблицы последовательно нумеруются арабскими цифрами в пределах всей работы или главы. Над правым верхним углом таблицы помещают надпись «Таблица 1». Ниже посередине страницы может быть помещен тематический заголовок.

Строки таблицы нумеруются только при переносе таблицы на другую страницу. Так же при переносе таблицы следует переносить ее шапку на каждую страницу. Тематический заголовок таблицы переносить не следует, однако над ее правым верхним углом необходимо указывать номер таблицы после слова «Продолжение».

Например: «Продолжение таблицы 1».

Столбцы таблицы нумеруются в том случае, если она не умещается по ширине на странице. Если таблица располагается на странице не вертикально, а горизонтально, то шапка таблицы должна располагаться с левого края страницы, как это показано на рисунке.

Все иллюстрации, не относящиеся к таблицам (схемы, графики, диаграммы и т.д.), именуется рисунками. Им присваивается последовательная нумерация либо сквозная для всего текста, либо в пределах главы. Все рисунки должны иметь полные наименования. Номер и наименование рисунка записываются в строчку под его изображением посередине страницы.

Например: «Рис. 3.1 Блок-схема основного модуля».

При переносе рисунка на следующую страницу его наименование указывать не следует, однако под рисунком необходимо указывать его номер после слова «Продолжение».

Например: «Продолжение Рис. 3.1».

Следует обратить внимание, что слова «Таблица» и «Рисунок» начинаются с большой буквы. Ссылки на иллюстративный материал в тексте дипломного проекта могут начинаться с маленькой буквы. Номера таблиц и рисунков указываются без каких-либо дополнительных символов.

Например: Примерный план выполнения и защиты дипломного проекта представлен в таблице 1.1.

Таблица 1.2

Модели принтеров

тип	производитель	марка	цена

Рис 4.2. Расположение горизонтальной таблицы

4.4. Правила составления списка литературы

Использованные в процессе работы специальные литературные источники указываются в конце дипломного проекта перед приложением. Список использованной литературы входит в основной объем работы. На каждый литературный источник в тексте работы обязательно должна быть хотя бы одна ссылка.

Список литературы может быть составлен либо в порядке упоминания литературных источников в дипломной работе либо в алфавитном порядке. Второй способ удобнее, т.к. в этом случае легче указывать ссылки на литературу в тексте дипломного проекта. Список адресов серверов Internet указывается после литературных источников.

При составлении списка литературы в алфавитном порядке следует придерживаться следующих правил очередности:

- законодательные акты и постановления правительства РФ;
- специальная научная литература;
- методические, справочные и нормативные материалы, статьи периодической печати.

Для многотиражной литературы при составлении списка указываются: полное название источника, фамилия и инициалы автора, издательство и год выпуска (для статьи – название издания и его номер). Полное название литературного источника приводится в начале книги на 2-3 странице.

Для законодательных актов необходимо указывать их полное название, принявший орган и дату принятия. При указании адресов серверов Internet сначала указывается название организации, которой принадлежит сервер, а затем его полный адрес.

Пример оформления литературных источников:

Информационные системы в экономике: Учебник / Под ред. проф. В.В. Дика. - М.: Финансы и Статистика, 2010. - 300 стр.: ил..

Положение по бухгалтерскому учету «Учет расходов на научно-исследовательские, опытно-конструкторские и технологические работы» (ПБУ 17/02). Утверждено Приказом Минфина РФ от 19 ноября 2002 г. №115н (с изм. и доп. от 18 сентября 2006 г.)

Фирма 1С. WWW:<http://www.1c.ru>

При ссылке на литературу в тексте приводится порядковый номер источника, заключенный в квадратные скобки. При приведении дословной цитаты из источника указывается также страница, на которой содержится данная цитата.

Например: «Программное обеспечение - это совокупность программ системы обработки данных и программных документов, необходимых для эксплуатации этих программ» [7. стр.18].

4.5. Правила оформления приложения

Приложения оформляются как продолжение дипломного проекта на последующих ее страницах, но в основной листаж не включаются. Содержание приложений определяется студентом-дипломником по согласованию с научным руководителем. При этом в основном тексте работы целесообразно оставить только тот иллюстративный материал, который позволяет непосредственно раскрыть содержание излагаемой темы. Вспомогательный же материал выносится в приложения. Объем приложений не ограничивается, поэтому основной листаж можно регулировать за счет переноса иллюстративного материала в приложения или из приложений.

Если приложения однородны по своему составу, то им предшествует отдельный лист с надписью «Приложение». В том случае, когда в работе

содержатся приложения нескольких видов, они нумеруются последовательно арабскими цифрами: «Приложение 1», «Приложение 2» и т.д., кроме того каждое приложение может иметь свое тематическое название.

Например: Приложение 5. Текст основных программных модулей. На каждое приложение в тексте работы обязательно должна быть хотя бы одна ссылка.

5. ПРИМЕРНАЯ ТЕМАТИКА ДИПЛОМНОГО ПРОЕКТИРОВАНИЯ

1. Автоматизация учета оптовых закупок товаров.
2. Автоматизация учета оптовой продажи товаров.
3. Автоматизация продажи товаров (продукции) с использованием Интернет-технологий.
4. Автоматизация учета товарных запасов.
5. Автоматизация количественно-стоимостного (складского) учета товаров.
6. Автоматизация учета складских операций.
7. Автоматизация учета товаров в магазине.
8. Автоматизация учета розничной продажи товаров.
9. Автоматизация учета денежных средств.
10. Автоматизация учета материалов.
11. Автоматизация учета реализации продукции.
12. Автоматизация расчета и учета себестоимости продукции.
13. Автоматизации-учета основных средств.
14. Автоматизация учета выполнения договоров оказания услуг.
15. Автоматизация расчета и учета заработной платы.
16. Автоматизация учета расчетов с поставщиками (заказчиками, клиентами).
17. Автоматизация учета услуг (работ).
18. Автоматизация расчета и учета налогов.
19. Автоматизация аудита заработной платы.
20. Автоматизация аудита кассово-банковских операций и расчетов.
21. Автоматизация аудита основных средств и нематериальных активов.
22. Автоматизация аудита себестоимости продукции.
23. Автоматизация аудита реализации готовой продукции.
24. Автоматизация аудита товарных (производственных) запасов.
25. Разработка АРМ «Кладовщик».
26. Разработка АРМ «Менеджер по закупкам».
27. Разработка АРМ «Менеджер по продажам».
28. Разработка АРМ «Экономист по зарплате».

29. Разработка АРМ «Бухгалтер по учету арендной платы».
30. Автоматизация учета вкладных операций.
31. Автоматизация учета кредитных операций.
32. Автоматизация учета валютных операций.
33. Автоматизация выездных налоговых проверок.
34. Система ведения документооборота в управлении образования..
35. Автоматизация принятия кредитного решения в коммерческом банке.
36. Автоматизация предварительного отбора инвестиционных проектов.
37. Автоматизация формирования пакета заявок клиентов с использованием Web-технологий.
38. Подсистема поддержки принятия инвестиционных решений на примере программного продукта Wealth-lab.
39. Подсистема прогнозирования налоговых поступлений в налоговых АИС.
40. Автоматизация разработки бюджета предприятия.
41. Разработка подсистемы оценки финансового состояния банка для банковских АИС.
42. Разработка подсистемы оценки страховых рисков для страховых АИС.
43. Разработка системы поддержки принятия решений при приемке готовой продукции на предприятии.
44. Разработка АИС управления малым торговым предприятием.
45. Разработка подсистемы оценки критических ситуаций в экономической сфере.
46. Совершенствование технологий автоматизированного планирования на основе применения программного продукта «Microsoft Project».
47. Совершенствование технологий автоматизированной оценки состояния предприятий.
48. Разработка подсистем интеллектуальной поддержки работы органов управления для АИС предприятий.
49. Разработка программного комплекса для оценки и анализа возможностей экономических систем.
50. Разработка экспертных подсистем для АИС управления предприятиями.
51. Разработка АРМ экономиста планово-экономического отдела предприятия.
52. Разработка автоматизированной системы учета торговой деятельности предприятия.
53. Разработка автоматизированной системы поддержки принятия решений отдела маркетинга.

54. Разработка автоматизированной системы учета кадров на предприятии.
55. Разработка Интернет-магазина.
56. Разработка автоматизированной системы складского учета.
57. Разработка автоматизированной экономической системы формирования отпускных цен.
58. Разработка автоматизированной системы планирования производства.
59. Разработка автоматизированной системы ведения внутрихозяйственных расчетов.
60. Разработка интеллектуальной информационной системы для экономиста хозяйства.
61. Информационная система поддержки принятия решений по выбору системы оплаты труда.
62. Информационная система анализа рядов динамики экономических показателей.

Формулировка темы включает также наименование организации, на практических материалах которой выполняется дипломный проект.

Например: «Автоматизация учета расчетов с клиентами по услугам телефонной связи в ЗАО «ЕС Энергия».

В названии и содержании проекта наряду с учетом, может быть отражен контроль, анализ.

Например: «Автоматизация учета и контроля расчетов с лизингополучателями в ООО «АПРАН».

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Боггс У., Боггс М. UML и Rational Rose 2002. – М.: Издательство «ЛОРИ», 2004. – 528 с.
2. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2002. – 352 с.
3. Грекул В. И., Денищенко Г. Н., Коровкина Н.Л. Проектирование информационных систем. Серия: Основы информационных технологий. – М.: Издательство: Бинوم. Лаборатория знаний, 2008. – 304 с.
4. Жарков С.В. Shareware: профессиональная разработка и продвижение программ. – СПб.: БХВ-Петербург, 2003. – 320 с.
5. Коберн А. Современные методы описания функциональных требований к системам: Пер. с англ. – М.: ЛОРИ, 2002. – 263 с.
6. Лешек А. Мацяшек Анализ и проектирование информационных систем с помощью UML 2.0. - 3-е издание. – М.: Издательство: Вильямс, 2008. – 816 с.
7. Маклаков С.В. BPWin и ERWin. CASE-средства разработки информационных систем. – М.: ДИАЛОГ-МИФИ, 2001. – 304 с.
8. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М.: Диалог-МИФИ, 2007. – 400 с.
9. Одинцов И.О. Профессиональное программирование. Системный подход. – СПб.: БХВ-Петербург, 2002. – 512 с.
10. Черемных С.В., Семенов И.О., Ручкин В.С. Структурный анализ систем: IDEF-технологии. - М.: Финансы и статистика, 2001. – 208 с.

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»

Кафедра _____

Допустить к защите:

Зав. кафедрой _____

(дата, подпись)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему: _____

на материалах _____

Работу выполнил(а)
студент(ка) __ курса _____
факультета _____

группы _____

(Ф.И.О. студента)

Научный руководитель:

(уч. степень, уч. звание, Ф.И.О.)

Консультант:

(уч. степень, уч. звание, Ф.И.О.)

Москва 20__

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»

Утверждаю
Зав. кафедрой
«__» _____ 20__ г.

З А Д А Н И Е
по выпускной квалификационной работе

Студенту _____
(фамилия, имя, отчество)

Факультет _____

Учебная группа _____

Специальность _____

1. Тема выпускной квалификационной работы _____

_____ утверждена приказом по университету от «__» _____ 20__ г. № _____

2. Срок сдачи студентом законченной работы на кафедре _____

3. План выпускной квалификационной работы

ВВЕДЕНИЕ

Глава 1. _____

1.1 _____

1.2 _____

1.3 _____

Глава 2. _____

2.1 _____

2.2 _____

2.3 _____

Глава 3. _____

3.1 _____

3.2 _____

3.3 _____

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

4. График выполнения квалификационной работы

№	Этапы выполнения работы и мероприятия	Сроки выполнения
1.	Подбор литературы, ее изучение и проработка	
2.	Составление библиографии по основным источникам	
3.	Составление плана выпускной квалификационной работы и согласование ее с руководителем	
4.	Разработка и представление на проверку первой главы	
5.	Накопление, систематизация и анализ практических материалов	
6.	Разработка и представление на проверку второй главы	
7.	Разработка и представление на проверку третьей главы	
8.	Согласование с руководителем выводов и предложений	
9.	Переработка (доработка) выпускной квалификационной работы в соответствии с замечаниями	
10.	Разработка тезисов доклада для защиты выпускной квалификационной работы	
11.	Ознакомление с отзывом и рецензией	
12.	Завершение подготовки к защите с учетом отзыва и рецензии, представление работы на кафедру	

Научный руководитель: _____ (Ф.И.О., подпись)

Студент: _____ (Ф.И.О., подпись)

Вдовин Виктор Михайлович, д.в.н., профессор
Мазай Владимир Николаевич, к.т.н., доцент

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ
РАБОТЫ**

Работа издается в авторской редакции

Печ. л. 8,5. Тираж 500 экз.

АНО ВПО ЦС РФ
«Российский университет кооперации»
141014, Россия, г. Мытищи, Московская обл.,
ул. В. Волошиной, д.12/30