

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Романчук Иван Сергеевич
Должность: Ректор
Дата подписания: 30.11.2022 11:25:15
Уникальный программный ключ:
6319edc2b582ffdacea443f01d5779368d0957ac34f5cd074d81181530452479

РОССИЙСКАЯ ФЕДЕРАЦИЯ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

Ивашко А.Г.

ПОДГОТОВКА ДИПЛОМНЫХ РАБОТ

Учебное пособие для студентов по направлению
«Прикладная информатика»

А.Г. Ивашко ПОДГОТОВКА ДИПЛОМНЫХ РАБОТ. Учебное пособие для студентов по направлению «Прикладная информатика (в экономике)». Тюмень: Издательство Тюменского государственного университета, 2020. с.

Данное пособие посвящено дипломным работам, в которых ставится задача - разработка информационной системы. В пособии даны советы и рекомендации как по методам работы над поставленной задачей, так и по правилам оформлений результатов. Разработка информационной системы рассматривается с точки зрения объектно-ориентированного и структурного подхода. Основой для изложения структурного подхода являются методы и средства методологии SSADM (Structured Systems Analysis and Design Method), а для изложения объектно-ориентированного - RUP (Rational Unified Process). В пособии представлен обширный перечень методов моделирования, используемых при проектировании информационной системы, таких как: IDEF0, ERP, DFD, IDEF1X, структурные карты, блок схемы, UML диаграммы и т.д. На реальных студенческих проектах показаны все этапы разработки информационных систем, выполняемых в различных методологиях. Рассмотрены основные ошибки и трудности реализации проекта.

В пособии представлен богатый справочный материал, используемый при моделировании, анализе, проектировании и построении информационных систем. Как показывает многолетний опыт кафедры информационных систем ТюмГУ, выполнение дипломных работ не редко ставит в затруднительное положение не только студента, но и научного руководителя. Научные руководители могут познакомиться с примерами реализации полного жизненного цикла разработки ПО на основе современных методологий и фреймворков.

Ответственный за А.Г. Ивашко, д-р техн. наук, профессор

выпуск

СОДЕРЖАНИЕ

Введение	6
1. Методические рекомендации выполнения дипломной работы.....	8
Выпускная квалификационная работа. Основные понятия ее жизненного цикла.	9
Выбор тема дипломной работы и задание на ее выполнение	19
Оформление дипломной работы	25
Структура дипломной работы	26
Правила оформления дипломной работы	32
2. Жизненный цикл дипломной работы	41
Основные виды жизненного цикла разработки ПО	41
Каскадная модель.....	41
Инкрементная модель	43
Эволюционная модель	45
Выбор модели жизненного цикла	46
3. Подходы, используемые при разработке ИС.....	72
Планирование развития информационных технологий на предприятии.....	72
Подход SWOT	73
Подход BPR.....	74
Подход VCF (Value Chain Framework).....	76
Подход построение EA (Enterprise Architecture – архитектуры предприятия)	80
Подход, основанный на знании предметной области	88
Подходы к разработке программного обеспечения	89
Структурный подход.....	89
Объектно-ориентированный подход.....	91
Выбор распределенной архитектуры.	93

Трехзвенная архитектура.....	94
Реализация Web- приложений /39/.....	99
4. Методологии структурного моделирования.....	103
Моделирование потоков данных /53/.....	103
Методика построения DFD диаграммы.....	106
Моделирование данных	111
Концептуальное проектирование базы данных.....	111
Логическое проектирование.....	122
Физическое проектирование базы данных	136
5. Применение методологии ООП при разработке ИС	138
Бизнес моделирование	139
Определение требований к системе	142
Анализ и проектирование.....	146
Проектирование базы данных	153
6. Применение проектных методологий при выполнении дипломных работ	157
ТЕМА 1 «Информационная система распределения нагрузки по кафедре»	157
Описание задачи.....	158
Анализ требований к системе.....	161
Построение функциональной модели (ТО-ВЕ) выявление требований к обеспечивающим частям ИС.....	165
Схема данных.....	171
Определение логической структуры базы данных.....	173
Руководство пользователя.....	176
ТЕМА 2 «Информационная система поддержки дистанционного проведения практических и индивидуальных занятий»	181
Описание задачи.....	181

Описание организационной структуры	182
Постановка задачи	185
Требования к системе	187
Порядок сдачи/приемки.....	195
Проектирование информационного обеспечения	195
Схема данных.....	209
Проектирование программного обеспечения	211
Список используемых источников.....	221
Приложение А. План-график выполнения дипломной работы	227
Приложение Б. Шаблон задания на выполнение дипломной работы	229
Приложение В. Основные понятия жизненного цикла разработки по RUP ...	232
Приложение Г. Применение нотации UML	261
Приложение Д. Входные и выходные документы темы «Информационная система распределения нагрузки по кафедре»	274
Приложение Е. Входные и выходные документы темы «Информационная система поддержки дистанционного проведения практических и индивидуальных занятий»	282

Введение

В данное учебное пособие делается попытка рассмотреть вопросы создания информационных систем с практической точки зрения. Рассматриваются два подхода, используемые в этой области: объектно-ориентированный и структурный. Каждый из этих подходов включает большое многообразие методологий, достаточно хорошо освоенных в учебной литературе, как отечественных авторов, так и переводной. Предполагается, что читатель знаком с синтаксисом IDEF0, DFD, ERD, IDEF1X и языком UML, хотя в приложении дается краткое описание синтаксиса и семантики этих языков. Рассмотрено несколько примеров, которые дают возможность проиллюстрировать основные этапы создания системы.

Учебное пособие адресовано, в первую очередь, дипломникам специальностей «Прикладная информатика (по областям)» и «Информационные системы и технологии». Исходя из специфики этих двух специальностей, основное внимание уделяется построению архитектуры и структуры системы, и в меньшей степени уделяется внимание программной реализации этих архитектурных и структурных решений. Многие вопросы создания информационных систем, как, например, тестирование, оптимизация программного кода, обеспечение нефункциональных требований и т.д. описаны в учебном пособии только концептуально. По мнению автора, материал, представленный в книге, дает возможность, как планировать, так и выполнить дипломную работу проектного характера по этим специальностям. В тоже время, описанные методологии могут быть полезны при выполнении дипломных работ исследовательского характера.

В учебном пособии делается попытка формализовать процесс подготовки дипломной работы в соответствии с методологиями разработки программного обеспечения. В тоже время освещаются на многие вопросы, интересующие выпускника, такие как: характеристика дипломная работа или

дипломный проект; порядок защиты квалификационной работы; требованию к оформлению работа и т.д.

1. Методические рекомендации выполнения дипломной работы

Каждый студент понимает термины «дипломная работа» и «курсовая работа» где-то на интуитивном уровне. Курсовая работа – то, что выполняется на данном курсе, а дипломная работа выполняется по окончании обучения. Учитывая специфику в подготовки студентов по специальности «Прикладная информатика по областям» и «Информационные системы и технологии» мы будем связывать задачи дипломной работы с задачами, выполняемыми на стадиях жизненного цикла разработки информационной системы¹ (Software life cycle model - SLCM). Задачи, решаемые в дипломной работе, могут охватывать, как часть фаз SLCM, так и полный его полный цикл. В тоже время задачи, степень проработанности задач, количество и широта их охвата от темы к теме может существенно меняться. Этапы SLCM могут включаться в этапы жизненного цикла (ЖЦ) выполнения дипломной работы. Этапы этих двух жизненных циклов во многом переплетаются, их разделение может выглядеть надуманным, но в тоже время это дает возможность отделить методику выполнения учебной работы от практики разработки ИС, изложенных в последующих главах.

Данная глава посвящена вопросам ЖЦ выполнения дипломных работ, т.е. методики выполнения дипломной работы. ЖЦ работ во многом схож: начинаются с выбора руководителя и темы, а заканчиваются защитой данной работы перед комиссией. Пример графика выполнения дипломных работ приведен в Приложении А.

Дипломная работа может выполняться на протяжении всего обучения в университете. Ее выполнение может быть начато в виде научной работы уже

¹ Существует большое многообразие методологий разработки информационных систем, реализующее основные модели жизненного цикла, которые разработчик выбирает на начальном этапе создания системы. В пособие используются, в основном, примеры реализованные в методологиях Rational Unified Process и ГОСТ 34.

на первом курсе и продолжено в виде курсовых работ. Поэтому рекомендуется студентам младших курсов при выполнении курсовых работ ознакомиться с данным учебным пособием.

Выпускная квалификационная работа. Основные понятия ее жизненного цикла.

В соответствии с Положения об итоговой государственной аттестации выпускников высших учебных заведений Российской Федерации /1/ к видам итоговой государственной аттестации выпускников высших учебных заведений относятся: защита выпускной квалификационной работы и государственный экзамен. Форма выпускной квалификационной работы конкретизируется государственным стандартом по соответствующей специальности.

Для информатика (с квалификацией в области)² выпускная квалификационная работа представляет собой законченную разработку (дипломный проект³) в профессиональной области /2/. Квалификационная работа выпускника по специальности «Информационные системы и технологии» /3/ (далее ИСТ) представляет собой законченную разработку (дипломная работа⁴ или дипломный проект), в которой решается актуальная задача в соответствии с направлением подготовки.

Можно обратить внимание на различие в понятиях «дипломная работа» и «дипломный проект». Часто дипломную работу ассоциируют с работой научно-исследовательского характера, а дипломный проект с инженерной разработкой. Такое упрощенное понимание может привести к некоторой пу-

² Специальность прикладная информатика (по областям) – далее ПИЭ

³ Дипломный проект состоит из теоретических или экспериментальных исследований, расчётов, чертежей и объяснительной записки с обоснованием технико-экономической целесообразности и расчётно-конструкторскими данными /4/.

⁴ Дипломная работа, как правило, представляет собой самостоятельное исследование какого-либо актуального вопроса в области избранной студентом специальности и имеет целью систематизацию, обобщение и проверку специальных теоретических знаний и практических навыков выпускников. Дипломная работа предполагает достаточную теоретическую разработку темы с анализом экспериментов, наблюдений, литературных и др. источников по исследуемому вопросу /4/.

танице. Во-первых, и дипломный проект и дипломная работа предполагает теоретические или экспериментальные исследования. Во-вторых, научно-исследовательская работа может носить прикладной характер⁵. В-третьих, инженерная разработка автоматизированной системы, в соответствии с ГОСТ 34.601 /5/, включает научно-исследовательскую работу. Поэтому разграничение квалификационной работы на дипломный проект и дипломную работу весьма условно.

Порядок подготовки квалификационной работы не регламентируется нормативными документами, поэтому допускаются различные сценарии. В Государственном образовательном стандарте для каждой специальности и направлению отводится время для подготовки квалификационной работы. Так, например, по специальности ПИЭ это время составляет 12 недель. В соответствии с учебным планом, работа выполняется с апреля по середину июня на пятом курсе.

Подготовка квалификационной работы начинается гораздо раньше!!! Официально она начинается с утверждения тем дипломных работ (для специальности ПИЭ в октябре) перед преддипломной практикой. Далее работа продолжается на преддипломной практике. Для специальности ПИЭ она продолжается 14 недель. В это время должна быть выполнена основная часть дипломной работы (примерно 70-80%). Только после этого начинается подготовка квалификационной работы, обозначенная в учебном плане.

С другой стороны, на четвертом курсе студенты выполняют курсовую работу, которая также называется преддипломная курсовая работа по специальности. При этом рекомендуется тему курсовой преддипломной работы согласовывать с темой будущей дипломной работы. В курсовой работе, как правило, выполняется либо один из этапов разработки информационной си-

⁵ Прикладные научные исследования - исследования, направленные преимущественно на применение новых знаний для достижения практических целей и решения конкретных задач (ст. 2 Федерального закона от 23.08.1996 № 127-ФЗ "О науке и государственной научно - технической политике")

стемы дипломной работы (например, эскизный или технический проект по ГОСТ 34.601) либо выполняется полная проработка одного из ее модулей.

Рекомендуется следующий сценарий подготовки квалификационной работы (жизненный цикл квалификационной работы).

На третьем курсе вместе с научным руководителем курсовой работы по специальности обсудить возможные темы будущей квалификационной работы и в соответствии с планом ее подготовки выбрать тему курсовой работы. Так, например, тема курсовой работы третьего курса может быть связана: с освоением новой технологии, предполагаемой использовать в будущем; с исследованием предметной области; с выполнением «легкого» прототипа будущей информационной системы.

На четвертом курсе провести оценку результата достигнутого в курсовой работе третьего курса и, исходя из этого, определить тематику квалификационной работы. Именно на этом этапе необходимо сформулировать вместе с научным руководителем тему дипломной работы, при этом тема курсовой преддипломной работы по специальности должна быть согласована с этой темой. В рамках курсовой преддипломной работы можно выполнить эскизный проект информационной системы (разработка архитектуры информационной системы), а также разработать один из ее модулей (целесообразно тот, который связан с наибольшими рисками ее реализации).

В начале пятого курса необходимо оценить результаты, достигнутые в преддипломной курсовой работе по специальности, и в соответствии с ними скорректировать тему квалификационной дипломной работы и план ее выполнения (на преддипломной практике и на дипломе). Тема дипломной работы утверждается на заседании кафедры. При этом необходимо помнить, что тема может быть скорректирована в зависимости от результатов, полученных при выполнении преддипломной практике.

Вне зависимости является ли тема дипломной работы продолжением предыдущих изысканий или нет, она, с точки зрения Project Management /23,

24/ является новым проектом⁶. Модель жизненного цикла этого проекта и методология, используемая для его выполнения, определяется на начальном этапе его выполнения. Так, например, стандарт ИСО/МЭК 12207 /25/ определяет их первой задачей процесса разработки. В данном пособии рассматриваются две методологии разработки информационной системы: структурная и объектно-ориентированная. Применение каждой из этих методологий не ограничивает выбор модели жизненного цикла.

В тоже время, Унифицированный процесс разработки программного обеспечения использует объектно-ориентированный подход и инкрементную модель жизненного цикла, основанную на итерациях. Каждый из структурных методов, применяемых в данном пособии, использует итерационный характер разработки, при этом модель жизненного цикла не ограничивается.

Исходя из норм часов отведенных ГОСом по специальности ПИЭ на преддипломную практику и диплом, ориентировочно можно считать, что на преддипломной практике должно быть выполнено 70-80 % объема квалификационной работы. Именно этими цифрами руководствуется комиссия при оценке выполнения преддипломной практики⁷. В силу того, что задачи, решаемые дипломной работой, могут охватывать не все стадии жизненного цикла разработки информационной системы, очень сложно сформулировать требования к тем артефактам⁸, которые должны быть выполнены на преддипломной практике.

В том случае если в дипломной работе стоит задача разработка и выпуска первой версии информационной системы, то на преддипломной практике рекомендуется: закончить модель предметной области, проработать ос-

⁶ Проект - это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов.

⁷ По окончании преддипломной практики производится ее защита. Для проведения защиты создаются комиссии, состоящие из преподавателей кафедры (не менее трех преподавателей в комиссии).

⁸ Артефакт – спецификация физического элемента информации, используемого или порождаемого в процессе разработки программного обеспечения (например, им может быть внешний документ или рабочий продукт). Артефакт может быть модель, описание или программный продукт /19/.

новные сценарии работы системы, описание архитектуры системы, реализована базовая функциональность системы (сборка и окончательное тестирование системы могут быть не выполнены), выпущено предварительное руководство пользования.

Во время, отведенное для подготовки выпускной квалификационной дипломной работы, завершаются теоретические и экспериментальные исследования (например, завершается работа над информационной системой), производится оформление полученных результатов в форме дипломной работы (проекта), а также осуществляется подготовка доклада для ее защиты (тест доклада и демонстрационный материал).

Материал, изложенный в учебном пособии, в первую очередь посвящен процессу подготовки дипломной работы, хотя с некоторыми поправками он может быть использован и в практике разработки информационных систем (ИС) и программного обеспечения (ПО). Что отличает дипломную работу от профессиональной разработки информационной системы или ПО? Можно отметить несколько важных отличий.

1. Задачи, решаемые на дипломной работе, могут охватывать только часть ЖЦ разработки информационной системы или программного обеспечения. В частности, проект может быть учебный, т.е. не иметь практической реализации. В этом случае фаза внедрения (развертывание ИС) не выполняется, что в свою очередь упрощает тестирование системы на всех этапах ее разработки.
2. В качестве участника проекта (stakeholders) возникает еще одно действующее лицо – научный руководитель проекта, который очень часто одновременно выполняет несколько ролей, как со стороны заказчика, так и со стороны исполнителя проекта.
3. Государственная аттестационная комиссия оценивает не качество реализованного продукта, а определяет соответствие подготовки выпускника требованиям государственного образовательного стандарта высшего профессионального. В тоже время качество реализо-

ванной информационной системы характеризует и профессиональную подготовку студента.

4. Члены комиссии, очень часто, не имеет доступ к реализованной информационной системе, а производит оценку ее по представляемому материалу: доклад, ответы на вопросы, демонстрационный материал, дипломная работа (проект). Следовательно, разработанная информационная система (даже успешно внедренная), это только один из факторов, по которому оценивается квалификационная подготовка выпускника. В тоже время, опыт показывает, что успешная защита квалификационной работы возможна только при успешном решении практических задач, таких как, например, реализация информационной системы или программного обеспечения.

Указанные особенности можно отнести в равной степени и к курсовым работам, с учетом того, что объем курсовой работы меньше.

После защиты преддипломной практике, студент переходит к выполнению вида деятельности «Подготовка выпускной квалификационной работы». В этот период выпускник: исправляет ошибки, выявленные на этапе выполнения преддипломной практики; выполняет нерешенные задачи дипломной работы (например: реализует оставшиеся программные модули, выпускает руководство пользователем, дорабатывает модели стадии проектирования) и демонстрирует результат их выполнения руководителю; подготавливает и согласовывает с руководителем первый вариант текста дипломной работы, доклада и демонстрационного материала. Необходимо вновь подчеркнуть тот факт, что уровень квалификационной подготовки выпускника комиссия оценивает по представляемому материалу (текст диплома, доклад, презентация, ответы на вопросы), поэтому подготовки их нужно уделить достаточно времени и сил. Требования к структуре и оформлению текста дипломной работы и доклада представлены ниже.

Руководитель оценивает уровень выполненной студентом работы и его квалификационную подготовку. В том случае если оцениваемые параметры

соответствуют требованиям ГОСа, то руководитель допускает выпускника к защите дипломной работы.

На защите выпускник представляет: текст дипломной работы, доклад (продолжительность доклада 5-10 минут) и демонстрационный материал. Защита проводится перед комиссией, созданной из преподавателей выпускающей кафедры. В случае, если дипломная работа рекомендуется к защите, то назначается рецензент⁹ дипломной работы. Рецензирование дипломных работ осуществляется преподавателями данного или другого вузов и ведущими специалистами, сфера деятельности которых соответствует тематике дипломной работы.

После защиты выпускник исправляет выявленные ошибки и готовит окончательный вариант дипломной работы, подписывает его у руководителя и консультанта. Руководитель составляет письменный отзыв на дипломную работу. Дипломная работа передается рецензенту. В рецензии должны быть отмечены:

- актуальность темы дипломного проекта (работы),
- степень соответствия дипломного проекта (работы) заданию,
- оценка основных результатов работы (полученные модели, разработанные и примененные оригинальные методы исследования, разработанные информационные системы и программное обеспечение, полученные зависимости и т.д.);
- практическая значимость и возможность внедрения результатов работы, ожидаемый эффект;
- логичность построения текста дипломной работы,
- анализ обоснованности выводов и предложений;
- имеющиеся недостатки работы по содержанию, изложению и оформлению материала;

⁹ Выпускные квалификационные работы, выполненные по завершении основных образовательных программ подготовки специалистов и магистров, подлежат рецензированию /1/.

- оценка дипломного проекта (работы): "отлично", "хорошо", "удовлетворительно", "неудовлетворительно".

В том случае если рецензент не является работником данного вуза, его подпись должна быть заверена печатью той организации, в которой работает рецензент.

Дипломная работа (проект) вместе с отзывом руководителя и рецензией сдаются заблаговременно, перед защитой, для утверждения заведующим выпускающей кафедрой.

В соответствии с «Положения об итоговой государственной аттестации выпускников высших учебных заведений Российской Федерации» к защите выпускной квалификационной работы допускается лицо, успешно завершившее в полном объеме освоение основной образовательной программы по направлениям подготовки (специальности) высшего профессионального образования, разработанной высшим учебным заведением в соответствии с требованиями государственного образовательного стандарта высшего профессионального образования и успешно прошедшее все другие виды итоговых аттестационных испытаний /1/.

Утвержденная дипломная работа принимается к защите при представлении Государственной аттестационной комиссии следующих документов:

- зачетная книжка студента с отметкой деканата о выполнении студентом учебного плана;
- отзыва руководителя;
- рецензии на дипломную работу;
- дипломная работа, утвержденная заведующим кафедрой.

По желанию студента в ГАК могут быть представлены и другие материалы, характеризующие научную и практическую ценность дипломной работы (например, макеты, справки о внедрении, авторские свидетельства и т.д.).

Защита выпускной квалификационной работы (за исключением работ по закрытой тематике) проводится на открытом заседании государственной аттестационной комиссии. Дипломник представляет свою работу в виде доклада, с показом демонстрационного материала.

Регламент работы аттестационной комиссией утверждается и доводится до сведения студентов заранее. Типичный регламент работы аттестационной комиссии приведен ниже:

- доклад выпускника (до 15 минут);
- ответы на вопросы по работе;
- выступление научного руководителя, в котором он, как правило, характеризует личные качества выпускника, а не качество его квалификационной работы (в случае если научный руководитель отсутствует, то зачитывается отзыв руководителя);
- выступление рецензента квалификационной работы (в случае его отсутствия рецензия зачитывается);
- ответ выпускника на замечания, отмеченные рецензентом;
- выступление любого лица, пожелавшего высказаться по теме данной работе;
- заключительное слово выпускника.

Доклад должен быть четким конкретным и иллюстрирован слайдами презентации. Наиболее важные положения работы и иллюстративный материал, который трудно представить на слайдах могут быть выполнены в печатном виде, сброшюрованы и розданы перед докладом каждому члену ГАК.

Можно предложить следующую структуру построения доклада (Таблица 1.1).

Решения государственной аттестационной комиссий принимаются на закрытых заседаниях простым большинством голосов членов комиссий, участвующих в заседании, при обязательном присутствии председателя комиссии или его заместителя /1/.

Таблица 1.1 - Пример схемы построения доклада

Название и пример описания	Кол. слайдов	Кол. минут
Представление студента, темы работы и научного руководителя (соответствует первому слайду презентации)	1	0,5
Обоснование актуальности ¹⁰ темы, ее цели и задачи (описание существующих проблем; обзор известных способов их решения и их недостатков; обоснование цели исследования/разработки на основании перечисленных недостатков; постановка задач исследования/разработки на основании выбранной цели);	2-4	3-5
Используемые методы и инструменты, и их обоснование;	1-2	
Анализ выполненной работы (результаты работы сгруппированным согласно перечисленным выше задачам);	более 8	8-10
Практическая значимость результатов работы (внедрение результатов исследования/разработки и обоснование ожидаемого эффекта);	1-2	2-3
Заключение по работе (выводы и перспективы ее продолжения).	1-2	1-2
Апробация работы (название статей и докладов, сделанных по работе)	0-1	

Выпускник, защита квалификационной работы которого была оценена аттестационной комиссией на «неудовлетворительно», при восстановлении в вузе назначаются повторные итоговые аттестационные испытания в порядке, определяемом высшим учебным заведением. Повторные итоговые аттестационные испытания не могут назначаться высшим учебным заведением более двух раз /1/.

¹⁰ Актуальность- важность, значительность чего-либо в настоящее время, современность, злободневность (например, актуальный вопрос) /4/.

Выбор тема дипломной работы и задание на ее выполнение

Тема¹¹ курсовой (дипломной) работы определяются научным руководителем и должна соответствовать профилю специальности, быть направленной на решение актуальной задачи и соответствовать современному уровню техники и технологий. При выборе темы может приниматься во внимание: тематика будущей или настоящей профессиональной работы студента, предпочтение в используемой технологии, доступность автоматизируемого бизнес процесса и т.д. Так, например, наиболее существенным доводом при выборе бизнес процесса, в котором предполагается использование разрабатываемая информационная система, является его доступность для изучения.

Студенту может предоставляться право выбора темы, вплоть до предложения своей тематики с необходимым обоснованием целесообразности ее разработки.

Профиль специальности прописан в Государственном образовательном стандарте (ГОС). В качестве примера в таблице 1.2 приведены основные положения, определяющие профиль специальностей «Прикладная информатика (по областям)» /2/ и «Информационные системы и технологии» (в рамках направления подготовки «Информационные системы» /3/).

При описании темы должны быть, по крайней мере, определены: объект исследования¹² или разработки; цель работы; задачи, решаемые для достижения цели; метод или методологию проведения работы;

Исходя из описания объекта профессиональной деятельности этих специальностей, основным объектом исследования или разработки в курсовых и

¹¹ Тема (от греч. *théma*, буквально — то, что положено (в основу)) - предмет описания, изображения, исследования, выступления, дискуссии /4/.

¹² Объект – 1. В философии – всякое явление, существующее независимо от человеческого сознания. 2. В широком смысле – предмет, явление, которые человек стремится познать и на которые направлена его деятельность /16/

дипломных работах могут быть: информационные системы и различные виды ее обеспечения; информационные процессы, ресурсы, потоки и т. д.; функциональные процесс (технологические процессы и бизнес процессы¹³) и базы данных в предметной области.

Необходимо обратить внимание, что курсовые работы, даже на младших курсах, не могут иметь реферативный¹⁴ характер. Изучение новых информационных технологий, математического, информационного и программного обеспечения информационных систем, способов и методов проектирования, отладки, программной реализации и т.д. обязательно связано с практическим опробованием изучаемого материала. Поэтому одна из задач курсовой работы реализация и/или практическое опробование: методов, языков, алгоритмов и т.д.

Еще одной распространенной ошибкой при выборе темы - обобщенное ее формулирование. Так, например: «Информационная система бухгалтерского учета», «АРМ¹⁵ бухгалтера», или совсем абсурдные: «Информационная система строительного предприятия», «Информационные системы ЖКХ».

Рассмотрим некорректность формулировки каждой темы.

«Информационная система бухгалтерского учета». Из данного названия следует то, что объектом исследования (разработки) является информа-

¹³ Процесс: совокупность взаимосвязанных и взаимодействующих видов деятельности, преобразующая входы в выходы. /17/

¹⁴ Реферат (от лат. *refero* — докладываю, сообщаю), краткое изложение в письменном виде или в форме публичного выступления содержания книги, научной работы, результатов изучения научной проблемы; доклад на определённую тему, включающий обзор соответствующих литературных и других источников /4/.

¹⁵ Автоматизированное рабочее место - АРМ

ционная система, которая призвана использоваться в предметной области «бухгалтерский учет¹⁶» (призвана автоматизировать ее функции).

Таблица 1.2- Характеристики профиля специальности 080801 и направления подготовки 230200

Характеристики	Прикладная информатика (по областям)- 080801	Информационные системы и технологии- 230201
Области профессиональной деятельности	<p>Профессионально-ориентированная информационная система, – это совокупность:</p> <ol style="list-style-type: none"> 1) функциональных процессов и связанных с ними информационных процессов, специфичных в конкретной предметной области; 2) средств, способов и методов, направленных на создание и применение технологий сбора, хранения, анализа, обработки и передачи информации, существенно зависящих от специфики области применения; 3) единого управления процессами решения функциональных задач, а также информационными, материальными и денежными потоками в предметной области. 	<p>Информационные системы - область науки и техники, которая включает совокупность средств, способов и методов человеческой деятельности, направленных на создание и применение систем сбора, передачи, обработки, хранения и накопления информации.</p>
Объекты профессиональной деятельности	<p>Информационные процессы, которые определяются спецификой предметной области.</p> <p>События, функциональные процессы и базы данных в предметной области, действия по выработке управленческого решения или по разработке экспертного заключения, информационные потоки, ресурсы.</p> <p>Профессионально-ориентированные информационные системы</p>	<p>Информационные системы и сети, их математическое, информационное и программное обеспечение, способы и методы проектирования, отладки, производства и эксплуатации программных средств информационных систем в различных областях.</p>

¹⁶ Бухгалтерский учет — упорядоченная система сбора, регистрации и обобщения информации (в денежном выражении) об имуществе, обязательствах организаций и их движении путем сплошного, непрерывного и документального учета всех хозяйственных операций /18/

Виды профессиональной деятельности	<ul style="list-style-type: none"> - организационно-управленческая, - проектно-технологическая, - маркетинговая, - экспериментально- исследовательская, - консалтинговая, - аналитическая, - эксплуатационная деятельность. 	<ul style="list-style-type: none"> - проектно-конструкторская; - технологическая; - организационно-управленческая; - научно-исследовательская; - эксплуатационная.
------------------------------------	--	---

Замечание 1. Трудно себе представить, что в рамках дипломной работы может быть реализована информационная система, охватывающая такую обширную область деятельности.

Замечание 2. Можно утверждать с большой долей уверенности, что не существует деятельности или бизнес процесса, который реализуется на всех предприятиях одинаково. Так, например, есть бухгалтерский учет малых предприятий, государственных предприятий и т.д.

«АРМ бухгалтера»¹⁷. Смотрите замечание 2.

Замечание 3. Разработка АРМа предполагает изучение действий выполняемых одной ролью (в данном случае бухгалтер). В тоже время, современные методы анализа и оптимизации бизнес процессов (функциональное моделирование, реинжиниринг бизнес процессов, методы TQM и т.д.) рекомендуют рассматривать действия, выполняемые в бизнес-процессе, вне привязки к структурной организации предприятия, в которой обязанности закрепляются за конкретной должностью. Функции информационной системы (АРМ является частым случаем информационной системы) не могут определяться организационной структурой. В нашем примере у бухгалтера может быть очень много разнородных функций (например, снятие бухгалтерии с сигнализации). Информационная система может использоваться в выполнении бухгалтером одних бизнес-функций и не использоваться в других. Поэтому такая формулировка темы не раскрывает объект исследования.

¹⁷ Рассматривается тот случай, АРМ бухгалтера не является именем собственным.

Набор тем таких как: «ИС строительной организации», «Информационная система ЖКХ», «Информационная система ОАО «Рога и копыта»» и т.д. Все приведенные выше темы не раскрывают объект исследования (смотрите замечание 2 и 3). На ранних этапах работы научным руководителем может определять только направление работы, для того, что бы иметь большую свободу при реализации квалификационной работы. Такая ситуация может возникнуть, например, в том случае, когда рассматривается несколько бизнес процессов, выполняемых одной организационной структурой предприятия. Для того что бы выявить объект автоматизации необходимо изучить эти бизнес-процессы. В этом случае и появляются темы, в которых указывается не объект исследования/разработки, а объект предварительного изучения.

С другой стороны, тема квалификационной работы не может раскрыть цели и задачи, решаемые в ней. Поэтому рекомендуется студентам не ограничиваться при обсуждении с научным руководителем только названием темы. Необходимо на этом этапе (предпроектном) выявить требования к работе, позволяющие сформировать его план-график. С этой целью в рамках выполнения дипломной работе разрабатывается задание для ее выполнения. В задание, кроме объекта исследования, определяются цели, задачи. Задачи исследования (разработки) можно сформулировать только в том случае, если определена его концепция работы, т.е. существует понимание реализации проекта в общем виде. Построение концепции исследования или разрабатываемой системы может изменить или скорректировать формулировку темы. Можно предложить следующую последовательность работ по составлению задания и плана графика:

- 1) формулировка темы и цели работы, обоснование ее актуальности;
- 2) определение задач, которые необходимо решить для достижения цели;
- 3) нахождение методов решения поставленных задач;

- 4) составление плана-графика выполнения работ, после чего возможна корректировка темы курсовой работы и задач (возврат на второй шаг).

Работа над заданием это совместная работа научного руководителя со студентом. Студент оформляет задание на дипломную работу (Приложение Б), в котором отражены основные моменты технического задания и план-графика. Неформальный вид этого документа дает возможность иметь большую свободу при реализации жизненного цикла выполнения дипломной работы и упрощает процесс документирования. Утвержденное задание¹⁸ на выполнение дипломной работы является основанием для допуска студента к прохождению преддипломной практики (приложение А).

Рекомендуется подготовку задания проводить параллельно с документальным описанием границ (scope) своей выпускной квалификационной работы¹⁹. Границы проекта²⁰ могут быть описаны в произвольном виде. Вид документа и его назначение фактически определяется моделью жизненного цикла и методологией выполнения проекта. Можно использовать шаблон «Vision» (Видение) унифицированного процесса /7, 8/(например, Rational Unified Process) или же техническое задание²¹, оформленного согласно ГОСТ 34.602-89 /9/. Применение стандартизированных документов дает возможность воспользоваться опытом, накопленным в данной области ни одним поколением программистов.

Границы проекта определяют то, что мы хотели бы получить в результате выполнения проекта, но не определяют, как мы этого можем достигнуть.

¹⁸ Темы дипломных работ утверждаются на заседании выпускающей кафедры

¹⁹ Выпускная квалификационная работа с точки зрения управления проектами (PMBok, PRINCE II, ISO/TR 10006: 1997 и т.д.) является проектом (проект - это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов /3/), поэтому в учебном пособии используется терминология данной предметной области.

²⁰ Границы проекта или содержание проекта (Project Scope) - работы, которые необходимо выполнить, чтобы получить продукт, услуги или результат с указанными характеристиками и функциями /3/.

²¹ Часть технического задания, которая определяет цели и задачи создаваемой системы

Требуется определить перечень работ, требуемых для реализации проекта. Он определяется методологией разработки. Порядок выполнения работ определяются выбранной моделью жизненного цикла. Глава 2 посвящена описанию модели жизненного цикла и технологическому процессу создания информационных систем.

Оформление дипломной работы

Правила оформления дипломной и курсовой работы не регламентируются нормативными документами. Стандарты ГОСТ 7.32-2001/26/ и ГОСТ 2.105-95 /27/ определяют правила оформления текстовых документов. Учитывая то, что ГОСТ 7.32-2001 распространяется на отчеты о фундаментальных, поисковых, прикладных научно-исследовательских работ, в дальнейшем используются требования этого стандарта.

В стандартах регламентируются правила оформления, но они не содержат рекомендаций по составлению этого текста. Порядок решения задач, поставленных в работе, лучше выстраивать таким образом, чтобы вначале выполнялись работы с наибольшим риском. Исследование и проектная работа могут включать действия, которые дают отрицательный результат, что потребует либо корректировку задач, либо поиск других путей решения для поставленных задач. Более того, полученные результаты иногда требуют переосмыслить поставленные цели и задачи. Отчет по работе, напротив, должен быть выдержан в строгом стиле, последовательного изложения:

- постановка задач, которые вытекают из обзора существующих знаний в данной предметной области,
- результаты, полученные при решении поставленных задач.
- выводы по работе, являющиеся концентрированным выражением основных достижений работы.

В результате выполнения работы могут создаваться довольно большое количество документов. Так, например, при разработке информационной системы по методологии RUP могут создаваться модели на стадиях определения требований, анализа и проектирования, реализации и тестирования. Эти модели могут составлять довольно увесистый том. В отчет по дипломной работе требуется включать только те из этих моделей, который иллюстрируют результаты, вытекающие из поставленных задач. Так для представления архитектуры системы требуется представлять не все прецеденты, а только архитектурно-значимые.

Структура дипломной работы

Рекомендуется следующая структура дипломной и курсовой работы:

- **титульный лист;**
- **реферат;**
- **содержание;**
- определения;
- обозначения и сокращения;
- **введение;**
- **основная часть;**
- **заключение;**
- **список использованных источников;**
- приложения.

Обязательные структурные элементы выделены полужирным шрифтом. Остальные структурные элементы включаются по необходимости.

Введение представляет собой наиболее ответственную часть работы, поскольку содержит в сжатой форме все ее положения. Введение это как раз та часть работы, которую чаще всего читают члены комиссии. Рекомендуется приступать к написанию введения на заключительном этапе подготовки диплома. Можно предложить следующую структуру введения:

- актуальность темы: примерно на одной странице текста нужно показать, почему необходимо заниматься этой темой, какое современное представление о предметной области, в чем практическая и/или научная необходимость проведения исследований/разработки;
- цель работы и задачи: в работе должна быть, как правило, одна цель, часто цель выражает достижение экономического или социального эффекта (получение желаемого бизнес-результата); задачи (обычно 3-4 задачи) должны быть направлены на достижения цели, содержание задач должно отражать результат разработки в области информационных технологий, выраженный примерно на одном уровне абстракции (создание интерфейса и создание информационной системы отражает действия на разных уровнях абстракции);
- объекта исследования/разработки: описание объекта реального мира, в пределах которого проводилось исследование/разработка;
- методологические и теоретические основы исследования: перечень основных трудов, охватывающих отрасль знания, к которой относится данная тема, а также перечень признанных авторитетов (лидирующих фирм) в данной области;
- методы исследования/разработки: методы, применяемые в данной работе, например: системный анализ, DFD, ERD, математические и статистические методы;
- основные положения, выносимые на защиту: краткие выводы по работе;
- научная новизна: признаки которые позволяют утверждать о научной новизне /28/:
 - постановка новой научной проблемы,
 - введение новых научных категорий и понятий, развивающих представление о данной отрасли знаний,

- раскрытие новых закономерностей протекания естественных и общественных процессов,
- применение новых методов, инструментов, аппарата исследования,
- разработка и научное обоснование предложений об обновлении объектов, процессов и технологий,
- развитие научных представлений об окружающем мире, природе, обществе;
- практическая значимость работы: описание того, как и где результаты работы могут использоваться;
- апробация работы: перечисление журналов, сборников и других изданий, в которых были опубликованы материалы работы, а также перечисление конференций, на которых докладывались результаты работы.
- структура и объем работы: включает перечень количества: глав (например: введение, 4 глав, списка используемой литературы и 4 приложений), страниц, таблиц, рисунков, литературных источников.

Основная часть должна содержать:

- а) выбор направления исследований/разработки, включающий обоснование этого направления, методы решения задач и их сравнительную оценку, описание выбранной общей методики проведения работы;
- б) процесс теоретических и (или) экспериментальных исследований, включая определение характера и содержания теоретических исследований, методы исследований, методы расчета, обоснование необходимости проведения экспериментальных работ, принципы действия разработанных объектов, их характеристики;
- в) обобщение и оценку результатов исследований, включающих оценку полноты решения поставленной задачи и предложения по дальнейшим направлениям работ, оценку достоверности полученных результатов и их сравнение с аналогичными результатами отечественных и зарубежных работ, обоснование необходимости проведения дополнительных

исследований, отрицательные результаты, приводящие к необходимости прекращения дальнейших исследований и/или разработок

г) каждая глава работы должна содержать выводы.

Структура данной части отчета сильно зависит от выбранной темы и от характера выполняемой работы.

Для работы посвященной разработке информационной системы можно предложить следующую структуру.

1-глава – аналитический обзор. В данной главе проводится анализ имеющихся информационных систем, которые могут быть использованы для решения поставленных целей, выявляются их достоинства и недостатки. На основании проведенного анализа определяется направление разработки о возможные проблемы. В остальных параграфах данной главы проводится анализ известных из литературы путей решения выявленных проблем и формализуются задачи, которые требуется решить при проектировании и внедрении информационной системы. Задачи разработки являются выводами данной главы.

2-глава – методологии и средства разработки. В данной главе требуется обосновать выбор модели жизненного цикла, методологии и инструментальных средств, требуемых для разработки системы.

3-глава - исследование предметной области и определение функциональных требований к системе. Обосновываются требования к функциональной и обеспечивающим частям информационной системы на основании описания моделей предметной области.

4-глава - архитектура системы и описание ее основных компонентов. Приводится описание логического, реализационное и процедурное представление системы, а также описаны обеспечивающие части системы. В этой главе может быть описано представление развертывание и проведен анализ развертывание системы в производственных условиях.

5-глава- экономическая эффективность использования системы.

Ниже приводится одна из возможных структур отчета научно-исследовательской работы экспериментального характера.

1 глава - аналитический обзор. Приводится анализ публикаций, технической документации и других материалов, с целью определения способа достижения поставленной цели. Параграфы данной главы должны быть посвящены аналитическому обзору решения какой-то одной проблеме или группе проблем по данной тематике. На основании выявленных недостатков известных способов решений формулируются задачи требуемых решить в данном исследовании. Выводом по данной главе является постановка задач исследования.

2-глава – методы и средства исследования. В данной главе проводится описание методик и инструментальных средств, применяемых в работе.

3-глава – экспериментальные материалы исследования. В данной главе приводятся результаты проводимых испытаний и полученных экспериментальным путем зависимостей, а также проводится обсуждение и анализ полученных результатов.

4-глава- экономическая эффективность использования полученных результатов.

Данные примеры дают описание структуры дипломной работы, которая охватывает полный жизненный цикл разработки или исследования. Выше уже говорилось, что как дипломная работа, так и курсовые работы могут включать только некоторые этапы разработки информационной системы. В зависимости от объекта исследования/ разработки будет меняться и структура отчета.

Текст глав должен содержать иллюстративный материал в виде графиков, таблиц и рисунков. Но в тоже время он не должен быть ими перегружен. Если необходимо включить большое количество иллюстративного материала или отдельных документов, то они помещаются в приложение.

Заключение должно содержать:

- краткие выводы по результатам выполнений исследовательской (проектной разработки) или отдельных ее этапов;
- оценку полноты решений поставленных задач;
- разработку рекомендаций и исходных данных по конкретному использованию результатов полученных результатов;
- оценку технико-экономической эффективности внедрения;
- оценку научно-технического уровня выполненной исследований/разработки в сравнении с лучшими достижениями в данной области.

Реферат должен содержать: сведения об объеме отчета, количестве иллюстраций, таблиц, приложений, количестве частей отчета, количестве использованных источников; перечень ключевых слов; текст реферата. Текст реферата должен отражать:

- объект исследования или разработки;
- цель работы;
- метод или методологию проведения работы;
- результаты работы;
- технико-эксплуатационные характеристики;
- итоги внедрения;
- область применения;
- экономическую эффективность или значимость работы.

Если отчет не содержит сведений по какой-либо из перечисленных структурных частей реферата, то в тексте реферата она опускается, при этом последовательность изложения сохраняется.

Содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование), заключение, список использованных источников и наименование приложений с указанием номеров страниц, с которых начинаются эти элементы отчета.

Перечень определений начинают со слов: «В настоящем дипломной работе применяют следующие термины с соответствующими определениями».

Структурный элемент «**Обозначения и сокращения**» содержит перечень обозначений и сокращений, применяемых в данном документе. Запись обозначений и сокращений проводят в порядке приведения их в тексте работы с необходимой расшифровкой и пояснениями.

Список использованных источников должен содержать сведения об источниках, использованных при составлении отчета. Сведения об источниках приводятся в соответствии с требованиями ГОСТ 7.1.

В **приложения** рекомендуется включать материалы, связанные с выполненной работы, которые по каким-либо причинам не могут быть включены в основную часть. В приложения могут быть включены:

- промежуточные математические доказательства, формулы и расчеты;
- таблицы вспомогательных цифровых данных;
- схемы и модели, полученные при разработке системы
- инструкции, методики, разработанные в процессе выполнения работы;
- иллюстрации вспомогательного характера;
- копии технического задания, программы работ, договора или других исходных документов (шаблоны документов, используемые в бизнес процессе, примеры печатных форм, подготавливаемых с помощью системы);
- акты внедрения результатов работы.

Правила оформления дипломной работы

Для оформления дипломной (курсовой) работы необходимо ознакомиться со следующими ГОСТами:

ГОСТ 8.417-2002 ГСИ. Единицы физических величин;

ГОСТ 7.54–88 Система стандартов по информации, библиотечному и издательскому делу. Представление численных данных о свойствах веществ и материалов в научно-технических документах. Общие требования;

ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

ГОСТ 7.1-2003 Система стандартов по информации, библиотечному и издательскому делу. Библиографическое описание документа. Общие требования и правила оформления;

ГОСТ 7.12-93 Система стандартов по информации, библиотечному и издательскому делу. Сокращение слов на русском языке. Общие требования и правила;

ГОСТ 7.32-2001 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления;

ГОСТ 2.105-95 Общие требования к текстовым документам.

Дипломная (курсовая) работа должна быть выполнена печатным способом на одной стороне листа белой бумаги формата А4 через полтора интервала. Цвет шрифта должен быть черным, высота букв, цифр и других знаков — не менее 1,8 мм (кегель не менее 12).

Текст отчета следует печатать, соблюдая следующие размеры полей: правое — 10 мм, верхнее — 20 мм, левое и нижнее — 20 мм.

Сокращение русских слов и словосочетаний в отчете — по ГОСТ 7.12.

3.2 Построение отчета

Основную часть диплома (отчета по курсовой работе) следует делить на разделы, подразделы и пункты. При делении текста на пункты необходимо, чтобы каждый пункт содержал законченную информацию.

Разделы, подразделы, пункты и подпункты следует нумеровать арабскими цифрами и записывать с абзацного отступа. Разделы должны иметь порядковую нумерацию в пределах всего текста, за исключением приложений.

Номер подраздела или пункта включает номер раздела и порядковый номер подраздела или пункта, разделенные точкой. После номера раздела, подраздела, и пункта в тексте точку не ставят. Если текст отчета подразделяют только на пункты, их следует нумеровать, за исключением приложений,

порядковыми номерами в пределах всего отчета. Если раздел или подраздел имеет только один пункт, то нумеровать его не следует. На рисунке 1.1 приведен пример нумерации разделов, подразделов и пунктов.

Разделы, подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны четко и кратко отражать содержание разделов, подразделов. Заголовки разделов, подразделов и пунктов следует печатать с абзацного отступа с прописной буквы без точки в конце, не подчеркивая. Если заголовок состоит из двух предложений, их разделяют точкой.

Страницы отчета следует **нумеровать** арабскими цифрами, соблюдая сквозную нумерацию по всему тексту отчета. Номер страницы проставляют в центре нижней части листа без точки. Титульный лист включают в общую нумерацию страниц отчета. Номер страницы на титульном листе не проставляют. Нумерация страниц отчета и приложений, входящих в состав отчета, должна быть сквозная.

1 Типы и основные размеры

- 1.1 }
1.2 } ***Нумерация пунктов первого раздела документа***
1.3

2 Технические требования

- 2.1 }
2.2 } ***Нумерация пунктов второго раздела документа***
2.3

a)

a) _____

b) _____

1) _____

2) _____

в) _____

б)

Рисунок 1.1- Пример нумерации разделов, подразделов (а) и пунктов и подпунктов (б).

Иллюстрации (чертежи, графики, схемы, компьютерные распечатки, диаграммы, фотоснимки) следует располагать в отчете непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

Иллюстрации, за исключением иллюстрации приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается “Рисунок 1”. Слово “рисунок” и его наименование располагают посередине строки. Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например, Рисунок 1.1. Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Слово “Рисунок” и наименование помещают после пояснительных данных.

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например, Рисунок А.3.

При ссылках на иллюстрации следует писать «... в соответствии с рисунком 2». В тексте должна быть ссылка на каждую иллюстрацию. Иллюстрация предназначена для пояснения текста. Не допускается использовать иллюстрацию самостоятельно.

На все **таблицы** должны быть ссылки в отчете. При ссылке следует писать слово “таблица” с указанием ее номера. Пример оформления таблицы приведен на рисунке 1.2.

Таблица 3 Пример названия таблицы

Шапка					} Заголовки граф
					} Подзаголовки граф
					Строки
					(горизонтальные ряды)

Боковик (графа) Графа (колонки)

для заголовков)

Рисунок 1.2 – Пример оформления таблиц.

Примечания приводят в документах, если необходимы пояснения или справочные данные к содержанию текста, таблиц или графического материала. Примечания не должны содержать требований.

Если примечание одно, то после слова “Примечание” ставится тире и примечание печатается с прописной буквы. Одно примечание не нумеруют. Несколько примечаний нумеруют по порядку арабскими цифрами без проставления точки. Примечание к таблице помещают в конце таблицы над линией, обозначающей окончание таблицы. Несколько примечаний нумеруются по порядку арабскими цифрами.

Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Если уравнение не уместится в одну строку, то оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:), или других математических знаков, причем знак в начале следующей строки повторяют. При переносе формулы на знаке, символизирующем операцию умножения, применяют знак “X”.

Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Формулы в отчете следует нумеровать порядковой нумерацией в пределах всего отчета арабскими цифрами в круглых скобках в крайнем правом положении на строке (рисунок 1.3).

Пример

$$A=a:v, \tag{1}$$

$$B=c:e. \tag{2}$$

Рисунок 1.3- Пример нумерации формулы

Ссылки в тексте на порядковые номера формул дают в круглых скобках. Допускается нумерация формул в пределах раздела. В этом случае номер формулы состоит из номера раздела и порядкового номера формулы, разделенных точкой (например - 3.1).

Сведения об источниках следует располагать в порядке появления ссылок на источники в тексте отчета и нумеровать арабскими цифрами без точки и печатать с абзацного отступа. Оформление библиографическое описание осуществляется согласно ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. Ниже приводится краткое описание правил библиографического описания.

Ниже приводятся примеры библиографического описания.

Однотомные издания

Перроун, П. Д. Создание корпоративных систем на базе Java 2 Enterprise Edition [Текст] : рук. разработчика : [пер. с англ.] / Поль Дж. Перроун, Венката С. Р. «Кришна», Р. Чаганти. - М. [и др.] : Вильямс, 2001. - 1179 с.

Агафонова, Н. Н. Гражданское право [Текст] : учеб. пособие для вузов / Н. Н. Агафонова, Т. В. Богачева, Л. И. Глушкова ; под. общ. ред. А. Г. Калпина ; авт. вступ. ст. Н. Н. Поливаев ; М-во общ. и проф. образования РФ, Моск. гос. юрид. акад. - Изд. 2-е, перераб. и доп. - М. : Юристъ, 2002. - 542 с.

«Воспитательный процесс в высшей школе России», межвузовская науч.-практическая конф. (2001 ; Новосибирск). Межвузовская научно-практическая конференция «Воспитательный процесс в высшей школе России», 26-27 апр. 2001 г. [Текст] : [посвящ. 50-летию НГАВТ : материалы] / редкол.: А. Б. Борисов [и др.]. - Новосибирск : НГАВТ, 2001. - 157 с.

История России [Текст] : учеб. пособие для студентов всех специальностей / В. Н. Быков [и др.] ; отв. ред В. Н. Сухов ; М-во образования Рос. Федерации, С.-Петербург. гос. лесотехн. акад. - 2-е изд., перераб. и доп. / при участии Т. А. Суховой. - СПб. : СПбЛТА, 2001. - 231 с. - ISBN 5-230-10656-5.

Отдельный том

Казьмин, В. Д. Справочник домашнего врача [Текст]. В 3 ч. Ч. 2. Детские болезни / Владимир Казьмин. - М. : АСТ : Астрель, 2002. - 503 с.

Законодательные материалы

Российская Федерация. Конституция (1993). Конституция Российской Федерации [Текст] : офиц. текст. - М. : Маркетинг, 2001. - 39 с.; - ISBN 5-94462-025-0.

Правила

Правила безопасности при обслуживании гидротехнических сооружений и гидромеханического оборудования энергоснабжающих организаций [Текст] : РД 153-34.0-03.205-2001: утв. М-вом энергетики Рос. Федерации 13.04.01 : введ. в действие с 01.11.01. - М. : ЭНАС, 2001. -158 с. ; 22 см. - В надзаг.: ...РАО «ЕЭС России». - 5000 экз. - ISBN 5-93196-091-0.

Стандарты

ГОСТ Р 517721-2001. Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования [Текст]. - Введ. 2002-01-01. - М. : Изд-во стандартов, 2001. -IV, 27 с.

Сборник стандартов

Система стандартов безопасности труда : [сборник]. - М. : Изд-во стандартов, 2002. - 102 с.

Патентные документы

Пат. 2187888 Российская Федерация, МПК7 Н 04 В 1/38, Н 04 J 13/00. Приемопередающее устройство [Текст] /Чугаева В. И. ; заявитель и патентообладатель Воронеж. науч.-ислед. ин-т связи. - № 2000131736/09 ; заявл. 18.12.00 ; опубл. 20.08.02, Бюл. № 23 (II ч.). - 3 с.

А. с. 1007970 СССР, МКИЗ В 25 J 15/00. Устройство для захвата неориентированных деталей типа валов [Текст] / В. С. Ваулин, В. Г. Ке-майкин (СССР). - № 3360585/25-08 ; заявл. 23.11.81 ; опубл. 30.03.83, Бюл. № 12. -2с.

Промышленные каталоги

Оборудование классных комнат общеобразовательных школ [Текст] : каталог / М-во образования РФ, Моск. гос. пед. ун-т. - М. : МГПУ, 2002. - 235 с.

Отдельный том

Казьмин, В. Д. Справочник домашнего врача [Текст]. В 3 ч. Ч. 2. Детские болезни / Владимир Казьмин. - М. : АСТ : Астрель, 2002. - 503, [1] с.

Многотомные издания

Казьмин, В. Д. Справочник домашнего врача [Текст]. В 3 ч. Ч. 2. Детские болезни / Владимир Казьмин. - М. : АСТ : Астрель, 2002. - 503, [1] с.

Депонированные научные работы

Разумовский, В. А. Управление маркетинговыми исследованиями в регионе [Текст] / В. А. Разумовский, Д. А. Андреев ; Ин-т экономики города. - М., 2002. - 210 с.: схемы. - Библиогр.: с. 208-209. - Деп. в ИНИ-ОН Рос. акад. наук 15.02.02, № 139876.

Отчеты о научно-исследовательской работе

Формирование генетической структуры стада [Текст] : отчет о НИР (промежуточ.) : 42-44 / Всерос. науч.-исслед. ин-т животноводства ; рук. Попов В. А. ; исполн.: Алешин Г. П. [и др.]. - М., 2001. - 75 с. - Библиогр.: с. 72-74. - № ГР 01840051145. - Инв. № 04534333943.

Диссертации

Вишняков, И. В. Модели и методы оценки коммерческих банков в условиях неопределенности [Текст] : дис канд. экон. наук : 08.00.13 : защищена 12.02.02 : утв. 24.06.02 / Вишняков Илья Владимирович. - М., 2002. - 234 с. - Библиогр.: с. 220-230. - 04200204433.

Журнал

Актуальные проблемы современной науки [Текст] : информ.-аналит. журн. / учредитель ООО «Компания «Спутник +». - 2001, июнь - . - М. : Спутник +, 2001- . - Двухмес. - ISSN 1680-2721

Электронные ресурсы

Художественная энциклопедия зарубежного классического искусства [Электронный ресурс]. - Электрон. текстовые, граф., зв. дан. и прикладная прогр. (546 Мб). - М. : Большая Рос. энцикл. [и др.], 1996.

Статья

Двинянинова, Г. С. Комплимент : Коммуникативный статус или стратегия в дискурсе [Текст] / Г. С. Двинянинова // Социальная власть языка : сб. науч. тр. / Воронеж. межрегион. ин-т обществ. наук, Воронеж, гос. ун-т, Фак. романо-герман. истории. - Воронеж, 2001. - С. 101-106.

Боголюбов, А. Н. О вещественных резонансах в волноводе с неоднородным заполнением [Текст] / А. Н. Боголюбов, А. Л. Делицын, М. Д. Малых // Вестн. Моск. ун-та. Сер. 3, Физика. Астрономия. - 2001. - № 5. - С. 23-25.

Приложение оформляют как продолжение данного документа на последующих его листах или выпускают в виде самостоятельного документа.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ. После слова “Приложение” следует буква, обозначающая его последовательность.

2. Жизненный цикл дипломной работы

Основные виды жизненного цикла разработки ПО

В настоящее время существует множество моделей жизненного цикла, но из них можно выделить три фундаментальные [32, 33]:

- каскадная;
- инкрементная;
- эволюционная.

Каждая из указанных моделей может быть использована самостоятельно или скомбинирована с другими для создания гибридной модели жизненного цикла.

Каскадная модель

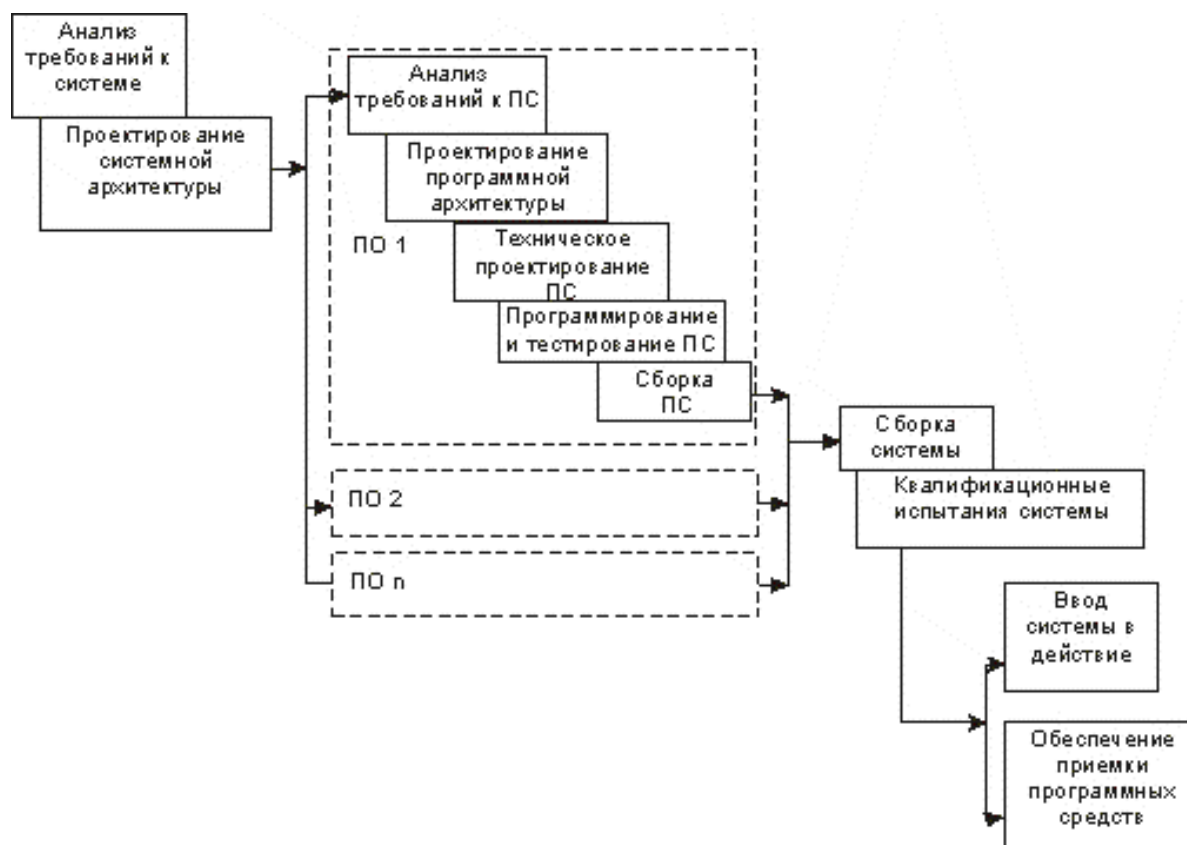
Каскадная модель жизненного цикла по существу реализует принцип однократного выполнения каждого из следующих видов деятельности в их естественных границах. Стандарт 34 выделяет следующие стадии:

- формирование требований к АС,
- разработка концепции АС,
- техническое задание,
- эскизный проект,
- технический проект,
- рабочая документация,
- ввод в действие,
- сопровождение АС.

Количество стадий может меняться весьма в широких пределах в зависимости от задач проекта. Обычно ее изображают в виде водопада.

При этом предполагается, что процесс, который стоит левее и выше доминирует над процессом правее и ниже. При применении такого принципа разработки каждого программного объекта соответствующие работы и зада-

чи процесса разработки обычно выполняют последовательно. Однако они могут быть частично выполнены параллельно в случаях перекрытия последовательных работ. В результате завершения определенной стадии каскадной модели создается тот или иной промежуточный продукт (выход поставки) и формируется базовый план²², являющейся исходной точкой для последующей стадии. Когда несколько программных объектов разрабатывают одновременно, для всех этих объектов работы и задачи процесса разработки могут быть выполнены параллельно (рисунок 2.1). Процессы сопровождения и эксплуатации обычно реализуют после процесса разработки. Процессы заказа и поставки, а также вспомогательные и организационные процессы обычно выполняют параллельно с процессом разработки.



ПО – программный объект

²² Базовый план (Baseline) Утвержденный план с указанными временными фазами {проекта, элементов иерархической структуры работ, пакета работ или плановой операции); возможно включение содержания проекта, стоимости, расписания и технических изменений /34/.

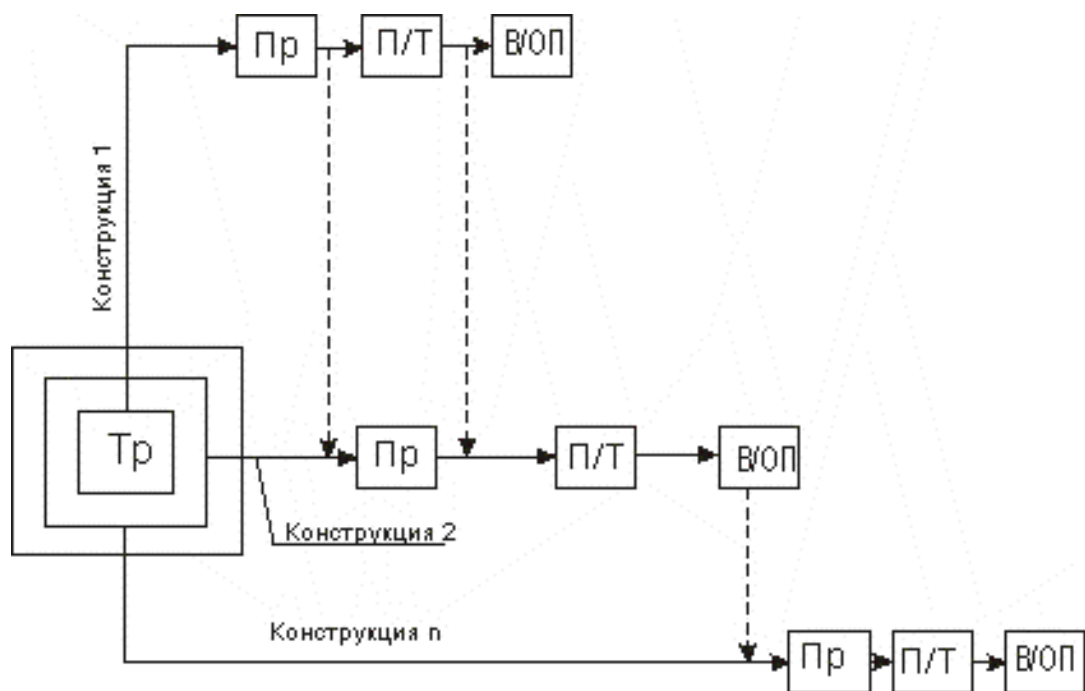
Рисунок 2.1 Представление каскадной модели в руководстве ГОСТ Р ИСО/МЭК ТО 15271-2002 /32/

Обычно считается, что классическая каскадная модель не предполагает возвращения на предыдущие стадии. В тоже время, можно модифицировать модель обратной связью, которая дает возможность вернуться на предыдущий этап. Существование обратных связей не усиливают модели, а наоборот ее разрушают, так как они реализуют циклы разработки, для которых в данной модели не существует методик управления.

Инкрементная модель

Инкрементная модель жизненного цикла, называемая также запланированным усовершенствованием продукта, начинается с выдачи набора требований и реализует разработку последовательности конструкций (рисунок 2.2). Первая конструкция содержит часть требований, в последующую конструкцию добавляют дополнительные требования и так далее до тех пор, пока не будет закончено создание системы. Для каждой конструкции выполняют необходимые процессы, работы и задачи, например анализ требований и создание архитектуры могут быть выполнены сразу, в то время как разработку технического проекта программного средства, его программирование и тестирование, сборку программных средств и их квалификационные испытания выполняют при создании каждой из последующих конструкций.

В данной модели при разработке каждой конструкции работы и задачи процесса разработки выполняют, также как и в каскадной модели, последовательно или частично параллельно с перекрытием. При частично одновременной разработке последовательных конструкций работы и задачи процесса разработки могут быть выполнены параллельно для ряда конструкций.



-----> информационные потоки

Тр- требования;

Пр- проектирование;

П/Т- программирование и тестирование;

В/ПП- ввод в действие и обеспечение приемки

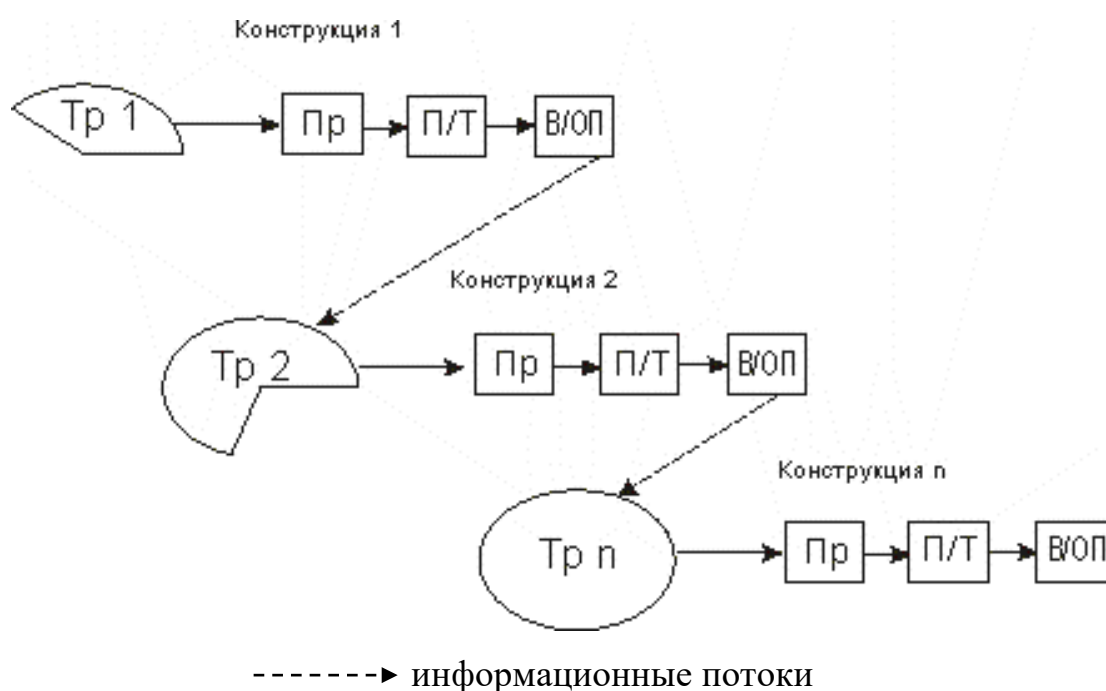
Рисунок 2.2. - Пример инкрементной модели /32/

Для каждой конструкции обычно выполняются одни и те же работы и задачи процесса разработки и в той же последовательности. Процессы сопровождения и эксплуатации могут быть реализованы параллельно с процессом разработки. Процессы заказа и поставки, а также вспомогательные и организационные процессы обычно выполняют параллельно с процессом разработки.

В методологии RUP модель называют инкрементная и итерационная. Этим подчеркивается то, что в каждой итерации (последовательность: бизнес-моделирование, управление требованиями, анализ и проектирование, реализация, тестирование и развертывание) реализуется артефакты, которые являются инкрементами системы.

Эволюционная модель

Модель предполагает разработка продукта (прототипа), который на последующих стадиях разработки будет усовершенствован. В эволюционной модели жизненного цикла систему также разрабатывают в виде отдельных конструкций, но в отличие от инкрементной модели требования изначально не могут быть полностью осознаны и установлены. В данной модели требования устанавливаются частично и уточняются в каждой последующей конструкции (рисунок 2.3).



Тр- требования;

Пр- проектирование;

П/Т-программирование и тестирование;

В/ПП- ввод в действие и обеспечение приемки

Рисунок 2.3 - Пример эволюционной модели /32/

При эволюционном методе для каждой конструкции работы и задачи процесса разработки выполняют последовательно или параллельно с частичным перекрытием.

Работы и задачи процесса разработки обычно выполняют многократно в той же последовательности для всех конструкций. Процессы сопровожде-

ния и эксплуатации могут быть реализованы параллельно с процессом разработки. Процессы заказа и поставки, а также вспомогательные и организационные процессы обычно выполняют параллельно с процессом разработки.

Эволюционная модель представлена в литературе моделью эволюционного прототипирования и спиральной моделью.

В инкрементной модели и в модели эволюционной используется, так называемый, итерационный подход, который проявляется в многократном повторении практически одной последовательности процессов, работ и задач. Последовательное уточнений требований к системе на каждой последующей фазе и есть тот краеугольный камень, который отличает две итерационные модели. В эволюционной модели разрабатывается промежуточный прототип, для того, что бы определить или уточнить требования к системе, в модели инкрементной разрабатывается часть будущей системы. И в том и в другом случае в конце каждой итерации анализируется ее результат и планируется новая.

Выбор модели жизненного цикла

На выбор модели жизненного цикла дипломной работы могут влиять различные факторы:

- рекомендации научного руководителя,
- опыт работы с определенной моделью жизненного цикла,
- требования заказчиков информационной системы,
- наличие опыта разработки в данной предметной области,
- наличие задела в ранее выполненных курсовых работах.

Наибольшее внимание заслуживают две модели: каскадная и инкрементная. Использование эволюционной модели затруднительно по следующим причинам:

- для использования этой модели требуется опыт, как у дипломника, так и у научного руководителя;

- дипломная работа может быть одной из итераций эволюционной модели, в этом случае жизненный цикл ее можно свести к каскадной модели;
- для выполнения дипломной работы отводится не так много времени (на пятом курсе присутствуют такие значимые задачи, как: сдача Государственного экзамена, выполнение заданий на преддипломной практике и т.д.), поэтому осуществить две (или более) итераций эволюционного прототипирования трудно.

Выбранная модель жизненного цикла должна быть вписана в график выполнения учебных задач, приведенный в приложении А. Модель жизненного цикла является только скелетом, который должен быть заполнен планируемыми процессами входными и выходными артефактами проекта и сроками их выполнения. Жизненный цикл является базовым планом выполнения дипломной работы. По мере выполнения каждой стадии или итерации необходимо проводить анализ полученных результатов, что бы скорректировать и детализировать базовый план.

Завершение каждой стадии или итерации формирует веху выполнения дипломной работы, результат которой должен быть измеримым. Ниже мы предложим два варианта жизненного цикла выполнения дипломной работы, в которой ставится задача разработки информационной системы.

В таблице 2.1 представлен жизненный цикл дипломной работы, выполненный в соответствии с каскадной моделью. Входом в этот процесс является сформулированная тема дипломной работы.

Таблица 2.1 – Стадии и этапы жизненного цикла выполнения дипломной работы при каскадной модели*.

Стадии жизненного цикла	Этапы работ	Краткая характеристика этапа работ	Выходные артефакты
Формирование требований к информационной системе	Характеристики предметной области (объекта автоматизации)	<p>Оценку качества функционирования объекта и осуществляемых видов деятельности, выявление проблем, решение которых возможно средствами автоматизации;</p> <p>описание существующей функциональной и информационной структуры системы;</p> <p>анализ существующей информационной структуры, выявление недостатков ее;</p> <p>обоснование необходимости совершенствования информационной системы объекта, оценку (технико-экономической, социальной и т.д.) целесообразности создания ИС</p>	<p>Обоснование необходимости совершенствования информационной системы объекта.</p> <p>Производственно-хозяйственные, научно-технические и экономические цели и критерии создания ИС</p>
	Формирование требований пользователя к ИС	<p>Определение автоматизированных функций и комплексов задач, выявление требований к характеристикам их реализации в соответствии с действующими нормативно-техническими документами. Выявлений</p>	<p>Функции и задачи создаваемой АС.</p> <p>Предложения по совершенствованию ор-</p>

Стадии жизненного цикла	Этапы работ	Краткая характеристика этапа работ	Выходные артефакты
		ограничений по созданию ИС.	ганизации и технологии процесса деятельности
Разработка концепции АС	Изучение объекта автоматизации	Построение моделей предметной области «как есть» (as is), определение входных и выходных документов используемых в бизнес-процессе, построение модели данных бизнес-объектов.	Модель предметной области «as is». Концептуальная модель данных предметной области.
	Разработка вариантов концепции ИС и выбор варианта концепции ИС, удовлетворяющего требованиям пользователя	<p>Разработка альтернативных вариантов концепции создаваемой ИС и планов их реализации; оценка необходимых ресурсов на их реализацию и обеспечение функционирования; оценку преимуществ и недостатков каждого варианта; определение порядка оценки качества и условий приёмки системы; оценку эффектов, получаемых от системы.</p> <p>Сопоставительный анализ требований пользователя и вариантов концепции ИС на предмет удовлетворения требованиям пользователя. Обоснование выбора опти-</p>	<p>Концепция информационной системы.</p> <p>Ожидаемый результат и эффективность реализации ИС.</p>

Стадии жизненного цикла	Этапы работ	Краткая характеристика этапа работ	Выходные артефакты
		мального варианта концепции и описание предлагаемой ИС, оценка ожидаемого результата и эффективность реализации выбранного варианта концепции.	
Задание на дипломную работу	Разработка и утверждение технического задания и задания на дипломную работу.	Разработка и утверждение технического задания на создание ИС, согласование технического задания с научным руководителем, разработка план графика выполнения дипломной работы, согласование задания на дипломную работу	Задание на дипломную работу. Техническое задание на создание ИС
Технический проект (этап проектирования архитектуры)	Разработка проектных решений по системе и её частям	<p>Построение модели бизнес процесса «как должно быть» («to-be»). Разрабатывается организационное обеспечение системы.</p> <p>Разработка решений по структуре системы, подсистем, средствам и способам связи для информационного обмена между компонентами системы, подсистем.</p> <p>Разработка архитектуры информационной системы.</p> <p>Определение стратегии решений (например, стратегия реализации клиент серверной архитектуры).</p>	Модель бизнес процесса «to-be». Архитектура системы. Логическая структура данных системы.

Стадии жизненного цикла	Этапы работ	Краткая характеристика этапа работ	Выходные артефакты
		<p>Разработка решений по комплексу технических средств, его размещению на объекте.</p> <p>Разработка решений по составу информации, объёму, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам. Выполняется концептуальное и логическое проектирование базы данных. Определяется состав входных и выходных документов.</p> <p>Разработка решений по составу программных средств, языкам деятельности, алгоритмам процедур и операций и методам их реализации.</p>	
	Разработка документации на ИС и её части	Создание документации на функциональную и обеспечивающие части ИС: организационная, математическая, программное обеспечение и т.д.	Входные и выходные документы. Описание обеспечивающих частей. Описание архитектуры ИС.

Стадии жизненного цикла	Этапы работ	Краткая характеристика этапа работ	Выходные артефакты
Рабочая документация (детальное проектирование и реализация)	Детализированное проектирование и разработка программной документации	Создания пользовательских интерфейсов и реализация программных модулей реализующих бизнес логику приложения. Проектирование транзакций. Физическое проектирование базы данных и ее развертывание. Выбор файловой структуры базы данных. Разработка механизмов защиты. Тестирование программных модулей. Организация мониторинга и настройка функционирования системы. Разработка программной документации в соответствии с техническим заданием.	Программный код. Программная документация. Тестовые задания.
Ввод в действие	Подготовка объекта автоматизации к вводу АС в действие	<p>Реализацию проектных решений по организационной структуре ИС; обеспечение подразделений объекта управления инструктивно-методическими материалами; внедрение классификаторов информации.</p> <p>Загрузку информации в базу данных и проверку системы её ведения</p>	
	Проведение предварительных ис-	Испытание ИС на работоспособность и соответствие техническому заданию в соответствии с программой и	Акт о приемке ИС в опытную эксплуата-

Стадии жизненного цикла	Этапы работ	Краткая характеристика этапа работ	Выходные артефакты
	пытаний	методикой предварительных испытаний; устранение неисправностей и внесение изменений в документацию на ИС; оформление акта о приёмке ИС в опытную эксплуатацию.	цию.
	Проведение опытной эксплуатации и приемочных испытаний	<p>Опытная эксплуатация; анализ результатов опытной эксплуатации; доработку (при необходимости) программного обеспечения ИС; оформление акта о завершении опытной эксплуатации.</p> <p>Испытания на соответствие техническому заданию в соответствии с программой и методикой приёмочных испытаний; анализ результатов испытания и устранение недостатков, выявленных при испытаниях; оформление акта о приёмке ИС в постоянную эксплуатацию.</p>	<p>Акт о завершении опытной эксплуатации.</p> <p>Акт о приемке ИС в постоянную эксплуатацию.</p>
Оформление дипломной работы	Оформление отчета по преддипломной практике	Представление результатов выполнения дипломной работы в отчете по практике (подготовка отчета и презентационного материала по преддипломной практике, защита практики).	Отчет по преддипломной практике, доклад и презентация.

Стадии жизненного цикла	Этапы работ	Краткая характеристика этапа работ	Выходные артефакты
	Оформление дипломной работы и ее защита.	Оформление дипломной работы. Подготовка доклада и презентации. Предзащита дипломной работы. Исправление выявленных ошибок. Представление дипломной работы рецензенту. Защита дипломной работы перед Государственной аттестационной комиссией.	Дипломная работа с отзывом научного руководителя и рецензией. Доклад и презентации.

* Стадии и этапы жизненного цикла создания информационной системы представлены в соответствии со стандартом ГОСТ 34.601-90. Представленный жизненный цикл нужно рассматривать как один из возможных вариантов. При этом нужно учитывать, что создание информационной системы в дипломной работе может охватывать только один или несколько этапов, представленных в ГОСТ 34.601-90. По согласованию с научным руководителем некоторые этапы могут быть, как удалены, так и добавлены в рассмотренную схему.

Ниже приводится пример реализации дипломной работы, с использованием инкрементной модели (таблица 2.2), построенной в соответствии с методологией RUP /31, 36, 38,39/. В приложении Ж приведен краткий обзор модели жизненного цикла RUP. Напомним, что RUP является унифицированным процессом от компании Rational (ныне IBM), который, с одной стороны, является конкретизацией модели SPeM /35/, а, с другой стороны, является шаблоном или фреймворком (на что указывает название унифицированный), требующий настройки для данного процесса разработки. В книге Кролл П.и Крачтен Ф /36/ показано, как унифицированный процесс можно настроить для малого проекта с одним разработчиком на одну неделю и для масштабного многолетнего проекта. В приложении Д рассмотрена кратко методология RUP.

Отличительной особенностью инкрементной итерационной модели используемой в унифицированном процессе является то, что все артефакты разрабатываемые в проекте могут меняться на последующих итерациях. Если в каскадной модели изменение, например, технического задания является скорее нештатной ситуацией, то изменение документа «Видения» в RUP является нормой.

Проект реализуется в четырех фазах: начало, уточнение, построение и внедрение (используется терминология RUP). Во всех фазах планируется одна итерация, кроме фазы построение, где запланировано 2 итерации. К артефактам внешним относятся: задание на выполнение дипломной работы, модели, полученные при разработке, отчет по преддипломной практике и дипломная работа. Все остальные артефакты можно считать внутренними, технологическими (в том числе и код программы).

Научный руководитель может выступать в нескольких ролях: заинтересованное лицо (лицо не участвующее в разработке, но заинтересованное в его результатах), руководитель проекта и ревьюер (reviewer - рецензент) и т.д.

Таблица 2.2 – Жизненный цикл выполнение дипломной работы. Фаза «Начало» (одна итерация), веха целей жизненного цикла.

Цели	Краткий перечень работ	Артефакты	Критерии оценки
Понять, что создавать: установить проектные возможности и границы системы.	Определение проблем и основных причин. Описание предметной области (возможно моделирование предметной области.) Создать глоссарий предметной области. Выявление заинтересованных сторон и пользователей системы. Определение границу системных решений. Выявить ограничения которые необходимо наложить на решении.	Видение, Глоссарий предметной области, бизнес-модель предметной области. Черновой вариант первой главы дипломной работы (аналитический обзор)	Согласие заинтересованных сторон по определению первоначальной оценки ресурсов.
Выяснить ключевые функции системы: определить прецеденты и первоначальные сценарии,	Выявление акторов и прецедентов. Краткое описание прецедентов. Разработка прототипов пользовательского интерфейса. Выявление ключевых функций системы. Выявление нефункциональных требований к архитектуре системы.	Диаграмма прецедентов, описание прецедентов, прототипы интерфейсов, доку-	Согласие в том, что зафиксирован верный набор требований, существует общее понимание этих требований

Цели	Краткий перечень работ	Артефакты	Критерии оценки
которые будут осуществляться основной проект.		мент дополнительная спецификация	
Выявить хотя бы одно возможное решение: показать вариант архитектуры в отношении некоторых первичных сценариев.	Выявление хотя бы одного архитектурного решения, обеспечивающий заданный уровень функциональности и удовлетворяющий нефункциональным требованиям (выбрать архитектуру и технологию, провести анализ эволюции системы, определить риски и стоимость решений, определить возможный состав программных компонентов, провести анализ возможного приобретения компонентов)	Модель дизайна, модель развертывания, модель анализа, документ «архитектура программного обеспечения»	Согласие в том, что издержки производства, приоритеты, риски и соответствующие процессы разработки приемлемы.
Оценить ресурсы, сроки риски: оценить в целом стоимость и график; идентифицировать	Создать экономическое обоснование проекта создания ИС, которое описывает экономическую ценность продукта (например, окупаемость продукта - ROI), создать перечень рисков выявить степень неопределенность решений проекта.	Черновой вариант главы дипломной работы «экономическое обоснование проекта»,	Согласие в том, что все риски были идентифицированы и существует стратегия уменьшения для каждого из рисков

Цели	Краткий перечень работ	Артефакты	Критерии оценки
<p>потенциальные риски (источники непредсказуемости).</p>		<p>график выполнение работ, документ «Перечень рисков»</p>	
<p>Подготовить опорную конфигурацию проекта: определить какому процессу следовать и какие средства использовать.</p>	<p>Настроить прецедент разработки под конкретные нужды проекта (согласовать его с учебными задачами: выполнение и защита практики, защита дипломной работы). Выбрать инструментальную среды для моделирования (ERWIN, VISIO и т.д.) и для разработки (Eclipse, Delphi, Visual Studio и т.д.). Определить методологию разработки (например, объектно-ориентированные: ICONIX, XP, SCRUM и т.д.)</p>	<p>Календарный план, задание на выполнение дипломной работы, план разработки программного обеспечения, план итераций, план выполнения работ</p>	<p>Утверждение задания на выполнение дипломной работы. Одобрение руководителя в использование выбранных инструментальных средств и методологий</p>

Таблица 2.3 – Жизненный цикл выполнение дипломной работы. Фаза «Уточнение» (одна итерация), веха архитектуры жизненного цикла.

Цели	Краткий перечень работ	Артефакты	Критерии оценки
<p>Создание базовой концепции ИС: достижения глубоко понимания требований.</p>	<p>Завершить описание большинства прецедентов. В конце фазы «Начало» описаны, как правило, только 20% прецедентов. В конце фазы «Уточнение» описаны, как правило, 80% прецедентов. Создать прототипы пользовательских интерфейсов. Просмотреть и протестировать с пользователем прецеденты. Намечить приоритет реализации прецедентов.</p>	<p>Документ «Видение», модель прецедентов, спецификация прецедентов, диаграмма навигации по окнам пользовательского интерфейса. Черновой вариант 3 главы дипломной работы (исследование предметной области и определение функциональных требований к системе).</p>	<p>Видение информационной системы и требования к ней устойчивы. Все заинтересованные стороны согласны с тем, что данное видение может быть выполнено, при условии выполнения текущего плана развития системы в контексте данной архитектуры.</p>

Цели	Краткий перечень работ	Артефакты	Критерии оценки
		ме)	
<p>Спроектировать, реализовать и проверить базовую архитектуру. Максимально быстро определить и утвердить базовую архитектуру.</p>	<p>Выбрать наиболее важных строительных блоков системы и их интерфейсов. Провести анализ возможности их закупки. Выявление архитектурно значимых прецедентов (20-30% прецедентов, которые реализуют наиболее значимые функции системы, или используют наиболее сложные, неизвестные и рискованные элементы требований). Для этих прецедентов выбрать одно или два сценария. Для них создать предварительную схему объектов аналитической модели (кооперация которых реализует эти сценарии). Распределить действия по классам аналитической модели. Создать аналитическую модель классов. Скорректировать модель прецедентов. Доработать классы аналитической модели до классов про-</p>	<p>Модель прецедентов. Диаграмма последовательности и диаграмма кооперации для реализуемых прецедентов. Диаграмма пакетов и компонент. Концептуальная логическая и физическая схема данных. Документ «архитектура программного обеспечения».</p>	<p>Архитектура устойчива. Проверены основные подходы, которые будут использоваться при тестировании и оценке системы.</p>

Цели	Краткий перечень работ	Артефакты	Критерии оценки
	<p>ектной модели.</p> <p>Объединить классы в пакеты. Выполнить проект базы данных.</p> <p>Создать схему параллельного выполнения процессов, потоков и и использования физических ресурсов.</p> <p>Определить архитектурные механизмы.</p> <p>Реализовать критические сценарии и провести модульное тестирование.</p> <p>Проверка того, что критические сценарии обеспечивают:</p> <ul style="list-style-type: none"> • желаемую функциональность; • приемлемую производительность (прецеденты с критической производительностью) • требуемые API интерфейсы. <p>Протестировать все нефункциональные тре-</p>		

Цели	Краткий перечень работ	Артефакты	Критерии оценки
	бования в дополнительной спецификации.		
<p>Снижение существующих рисков и оценка сроков выполнения и стоимости. Демонстрация того, что базовая линия архитектуры позволяет получить видение за разумное время и по разумной цене</p>	<p>Выполнить мероприятия, позволяющие снизить или устранить риски. Изучить основные технические средства и методы, используемые при разработке ИС. Обновить экономическое обоснование создание ИС.</p>	<p>Глава «экономическое обоснование». Документ «Список рисков»</p>	<p>Фактические расходы ресурсов в сравнении с запланированными. Проведены ключевые испытания и основные принципы доказаны; главные элементы риска были выявлены и решены.</p>
<p>Усовершенствование поддержки инфраструктуры.</p>	<p>Инсталляция и развертывания инструментов поддержки процесса разработки ИС. Реализация процессов управления требованиями и управления конфигурацией (код программного обеспечения должен отслеживаться эти</p>		

Цели	Краткий перечень работ	Артефакты	Критерии оценки
	процессом).		
Создания базовой линии уточненного план для фазы построения.	Провести анализ выполнения фазы. Провести корректировку плана проекта. Детализировать план итерации для фазы «Построение». Планирование исследования оставшихся рисков. Планирование порядка реализации оставшихся вариантов использования.	План итерации для фазы «Построение». План проекта	Итеративные планы фазы построения имеет обоснованные и точные детали, что позволяет продолжить работу, опираясь на вероятные оценки.

Таблица 2.4 – Жизненный цикл выполнение дипломной работы. Фаза «Построение » (первая итерация), промежуточная веха начальной функциональной готовности ИС и защита преддипломной практики.

Цели	Краткий перечень работ	Артефакты	Критерии оценки
<p>Минимизировать расходы разработки, оптимизировать ресурсы и избегать ненужных отходов и доработок</p> <p>Быстрее получить приемлемый уровень качества</p>	<p>Планировать построение системы с относительно малыми приращениями, что уменьшает рабочие потоки определения требований, анализа, проектирования, реализации и тестирования до решения нескольких вопросов.</p> <p>Развитие системы управления конфигурацией.</p>		<p>Все еще ли приемлемы расходы ресурсов по сравнению с проектными?</p>
<p>Подготовить продукт программного обеспечения к внедрению в производстве.</p> <p>Получить полезные</p>	<p>Определение оставшихся требований к системе и реализация 80% выявленных функциональных требований. В начале итерации делается акцент на борьбу с самыми крупными рисками. Основные потоки работ (для каждого при-</p>	<p>Модели системы уровня анализа и проектирования.</p> <p>Бета версия системы (программный</p>	<p>Является ли данный выпуск продукции устойчивым и достаточно готовым для внедрения его в пользовательской среде?</p>

Цели	Краткий перечень работ	Артефакты	Критерии оценки
<p>версии (альфа, бета, и другие испытательные выпуски) на столько быстро насколько это возможно</p>	<p>ращения) должен проходить последовательно через все дисциплины: определение требований (детализация и уточнения требований), анализ и проектирование, реализация, внедрение и тестирование. Варианты использования должны реализовываться в соответствии с их приоритетами. Тестирование должно проходить через стадии: планирования тестирования, разработка теста, проведение тестирования целостности, проведение тестирование системы. Модификация базы данных (добавление атрибутов, таблиц, триггеров и хранимых процедур) по мере того, как расширяется реализация новых возможностей.</p> <p>Доработка пользовательских интерфейсов.</p> <p>Часто создавать выпуски системы и проводить внедрение ее. Тестирование системы в пользо-</p>	<p>код).</p> <p>Физическая модель базы данных.</p> <p>Акт передачи системы для опытной эксплуатации.</p>	<p>Все ли заинтересованные стороны готовы к переходу продукта в коллектив пользователей?</p>

Цели	Краткий перечень работ	Артефакты	Критерии оценки
	вательской среде. Подготовить бета версию и передать ее для тестирования пользователями.		
Подготовить и защитить отчет по практике	Развернуть систему для демонстрации научному руководителю. Подготовить отчет по преддипломной практике и обсудить его с научным руководителем. Подготовить презентацию и доклад для защиты.	Отчет по практике, доклад, презентационный материал	Допуск научного руководителя к защите практики. Соответствие отчета по преддипломной практике и уровня знаний студента требованиям ГОС (Государственный образовательный стандарт)
Планирование второй итерации фазы «Построение»	Проведение анализа выполненных работ и оценки ее научным руководителем и комиссией при защите преддипломной практике. Разработка плана итерации для достижения целей фазы построения.	План проекта. Детальный план второй итерации	

Таблица 2.5 – Жизненный цикл выполнение дипломной работы. Фаза «Построение » (вторая итерация), промежуточная веха начальной функциональной готовности ИС и предзащита защита дипломной работы.

Цели	Краткий перечень работ	Артефакты	Критерии оценки
<p>Подготовить продукт программного обеспечения к внедрению в производстве.</p> <p>Получить полезные версии</p>	<p>Реализация оставшихся (20%) прецедентов.</p> <p>Организация работ ведется в соответствии с порядком, описанным в таблице 2.4. Устранение ошибок, выявленных на стадии тестирования бета-версии и при подготовке и защиты преддипломной практике.</p> <p>Реализация изменений выполняется в соответствии с трассируемостью (прослеживаемостью) процесса управления требованиями (например, с применением инструмента Rational Requisite PRO).</p> <p>Предоставление новой версии для проведение испытаний в пользовательской среде (предпо-</p>	<p>Модели системы.</p> <p>Акт о принятия системы к внедрению</p>	<p>Является ли данный выпуск продукции устойчивым и достаточно готовым для внедрения его в пользовательской среде?</p> <p>Все ли заинтересованные стороны готовы к переходу продукта в коллектив пользователей?</p>

	лагается, что внедрение системы будет проводиться вне рамок дипломной работы)		
Подготовка дипломной работы к защите	Подготовка дипломной работы. Описание результатов полученных на различных стадиях выполнения и включение ранее подготовленных глав. Согласование дипломной работы с научным руководителем. Подготовка доклада и презентационного материала к защите. Предзащита диплома.	Первый вариант дипломной работы.	Качество разработанной информационной системы и степень подготовки квалификационной работы к защите (оценивается допуском к предзащите).

Таблица 2.6 – Жизненный цикл выполнение дипломной работы. Фаза «Внедрения» (одна итерация), веха готового продукта (дипломной работы).

Цели	Краткий перечень работ	Артефакты	Критерии оценки
Подготовка работы к защите	Подготовка окончательного варианта дипломной работы. Подготовка доклада и презентационного материала. Представление дипломной работы рецензенту.	Дипломная работа. Отзыв научного руководителя. Рецензия на дипломную работу. Презентационный материал.	Качество разработанной информационной системы. Качество работы выпускника над дипломной работой (оценивается отзывом руководителя). Степень соответствия решения поставленных задач в дипломной работе требованиям ГОС (оценивается рецензентом)
Защита диплома	Доклад перед государственной аттестационной комиссией, ответы на вопросы и замечания рецензента, заключительное слово.	Решение государственной аттестационной комиссии	Степень соответствия подготовки выпускника требованиям ГОС.

Процесс RUP предоставляет не только модель жизненного цикла, но и набор унифицированных артефактов. Использование этих артефактов не является требованием унифицированного процесса. В жизненном цикле, представленном ниже, даются ссылки на артефакты RUP только для того, чтобы дать представление о назначении и содержании требуемых артефактов.

3. Подходы, используемые при разработке ИС

В данной главе будет рассмотрено несколько подходов используемых на различных стадиях разработки информационной системы.

- Планирования разработки ИС – подходы, используемы при определении стратегии развития ИТ и ИС на предприятии.
- Выбора методологии разработки программного обеспечения - структурный и объектно-ориентированный подход.
- Выбор распределенной архитектуры ИС – архитектурные решения при реализации ИС.

В данной главе не преследуется цель дать всеобъемлющее описание различных подходов при разработке информационных систем. Скорее здесь рассматривается только небольшая часть практических вопросов и возможные подходы их решения, которые не получили должного отражения в курсе проектирование информационных систем.

Планирование развития информационных технологий на предприятии

Всем понятно, что использование дорогостоящих информационных технологий может быть оправдано, только выгодами, полученными в бизнесе. Развитие информационных систем должно вестись на предприятии не спонтанно, а обосновываться потребностями бизнеса.

Бизнес стратегия может быть обоснована: анализом бизнес моделей, SWOT-анализом, реинжинирингом бизнес процессов, реализации процессного подхода управления предприятием, анализом информационных ресурсов предприятия, построением архитектуры предприятия и т.д.

Существует много способов планирования развития ИТ на предприятии. В дипломной работе требуется привести экономическое обоснование применение или разработки ИС. Можно воспользоваться известными в экономике подходами, приведенными ниже /39/.

Подход SWOT

SWOT - метод анализа в стратегическом планировании, заключающийся в разделении факторов и явлений на четыре категории: Strengths (Сильные стороны), Weaknesses (Слабые стороны), Opportunities (Возможности) и Threats (Угрозы).

Акроним SWOT был впервые введен в 1963 году в Гарварде на конференции по проблемам бизнес-политики профессором К. Andrews. Первоначально SWOT анализ был основан на озвучивании и структурировании знаний о текущей ситуации и тенденциях /40/.

Преимущества и недостатки - внутренние факторы, которые создают или разрушают ценность. Они могут включать активы, навыки или ресурсы, которыми имеет компания в своем распоряжении, в сравнении со своими конкурентами. Их можно измерить, используя внутренние оценки или внешний анализ бенчмаркинг.

Возможности и угрозы - внешние факторы, которые создают или разрушают ценность. Компания не может их контролировать. Они происходят от конкурентной динамики отрасли/рынка или от демографических, экономических, политических, технических, социальных, правовых или культурных факторов (PEST).

	Положительное влияние	Отрицательное влияние
Внутренняя среда	Strengths	Weaknesses
Внешняя среда	Opportunities	Threats

Можно представить следующий алгоритм процесса стратегического управления.

- Формирования миссии.
- Разработка целей (стратегические цели - это основные направления деятельности организации, ведущие к выполнению ее миссии).

- Стратегический анализ, который должен дать реальную оценку собственных ресурсов и возможностей применительно к состоянию (и потребностям) внешней среды, в которой работает фирма.
- Выбор стратегий из возможного множества вариантов.
- Реализация стратегии.

SWOT предоставляет инструмент качественного стратегического анализа, при этом не требуется мощных и дорогостоящих средств. Выбор эффективных стратегий, соответствующих внутренним параметрам предприятия и его положению на рынке и, в целом, во внешней среде, производится путем построения матриц корреляционного SWOT-анализа. На рисунке 3.1 приводится пример матрицы SWOT анализа.



Рисунок 3.1 Пример матрицы SWOT анализа /41/

Подход BPR

В 1990 Майкл Хаммер, профессор информатики в Массачусетском технологическом институте, опубликовал статью в Harvard Business Review /42/, в котором он утверждал, что обычные методы повышения производительности - рационализация и автоматизация процессов - не привели к серь-

езным улучшениям, которые требуются компаниям. В частности, серьезные инвестиции в информационные технологии принесли разочаровывающие результаты, во многом из-за того, что компании используют технологию только для того, чтобы механизировать старые способы вести дела. Они оставляют в неприкосновенности существующие процессы и используют компьютеры, чтобы просто их ускорить /43/.

Основная идея Хаммера /44/ состоит в том, что при проведении реинжиниринга необходимо освободиться от устаревших бизнес-процессов и принципов их разработки и создать новые. Для этого необходимо рассмотреть основополагающие процессы с кросс-функциональной точки зрения. Под кросс-функциональной точкой зрения понимается обзор бизнес процесса в целом без деления его на части, выполняемые различными функциональными группами. Фактически рассматривается процессный подход, применяемый в функциональном моделировании IDEF0, DFD, ARIS и т.д. Бизнес процесс не привязывается к структурным подразделениям выполняющим его.

Решающее место в реинжиниринге отводится ИТ. По мнению Хаммера компания должна видеть в ИТ не метод автоматизации производства а технологическое средства его изменения. Само по себе использование компьютеров не является реинжинирингом. Более того, неправильное использование компьютеров может заблокировать усилия по коренному изменению бизнес процессов, закрепить старые взгляды и схемы поведению.

Хаммер сформулировал принципы /43/, которые могут быть полезны при создании информационных систем.

- Организовывайте достижение результата, а не выполнение задачи.
- Поручите исполнение процесса тем, кто использует его результат.
- Включайте обработку информации в реальную работу, которая генерирует эту информацию.
- Считайте географически разпыленные ресурсы централизованными.

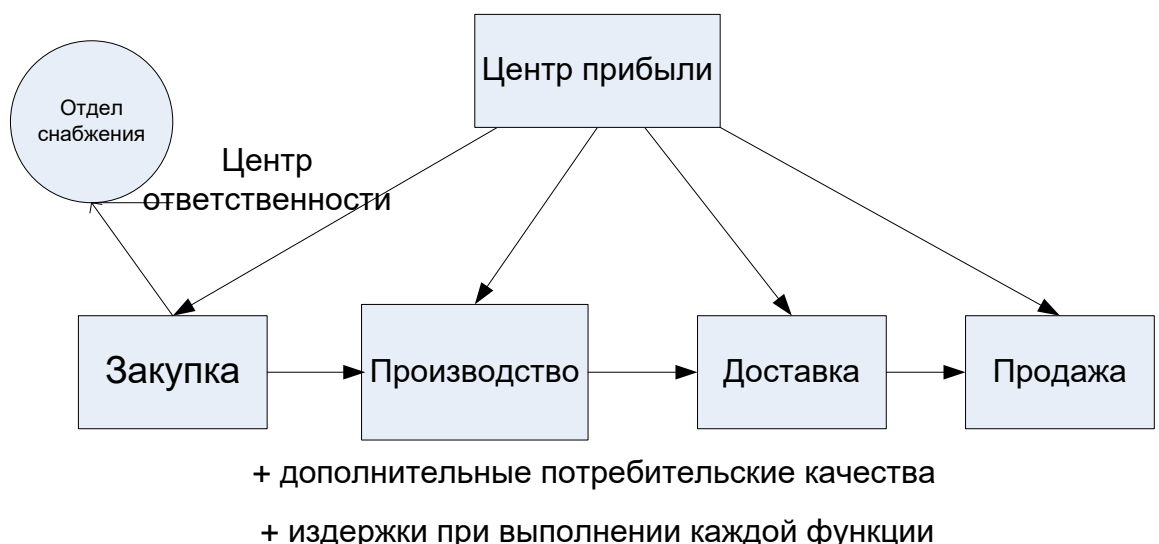
- Связывайте параллельные работы вместо интеграции их результатов.
- Помещайте точку принятия решения туда, где делается работа, и встраивайте контроль в процесс.
- Фиксируйте информацию один раз - у источника.

Подход VCF (Value Chain Framework)

Концепция Value Chain (Цепочка приращения стоимости) Портера (Michael Porter) - это модель, которая помогает проанализировать конкретные действия фирмы по созданию ценности и конкурентного преимущества /45/.

VCF включает (рисунок 3.2) последовательность функций структурных подразделений предприятия (центров ответственности) в том порядке, как они выполняются при создании конкретного вида потребительной стоимости (стоимостного объекта). Отдельные функции выполняются структурными подразделениями, которые имеют статус центра затрат /46/.

Выполнение каждой функции связано с одной стороны, с формированием затрат (издержек), с другой стороны, - с добавлением дополнительных потребительских качеств к создаваемому стоимостному объекту, представляющих ценность для потребителя.



Функции VCF подразделяют на основные (Primary activities) и вспомогательные (Техническая поддержка) элементы (рисунок 3.3).

К основным элементам относятся:

- Входящая логистика. Включает получение, хранение, контроль ТМЗ, планирование транспортных перевозок.
- Операции. Включает обработку, упаковку, сборку, обслуживание оборудования, тестирование и все другие действия, создающие ценность, которые преобразовывают вклады в окончательный продукт.
- Исходящая логистика. Действия, необходимые для доставки конечного продукта клиентам: хранение на складах, выполнение заказа, перевозки, управление дистрибуцией.
- Маркетинг и продажи. Действия, связанные с побуждением покупателей к совершению покупки продукта, включая: выбор канала сбыта, рекламу, продвижение, продажи, ценообразование, менеджмент розничных продаж и т.д.
- Сервисные услуги. Действия, которые поддерживают и повышают ценность продукта, включая: клиентскую поддержку, ремонтные услуги, установку, тренинг, управление запасных частей, модернизацию и т.д.

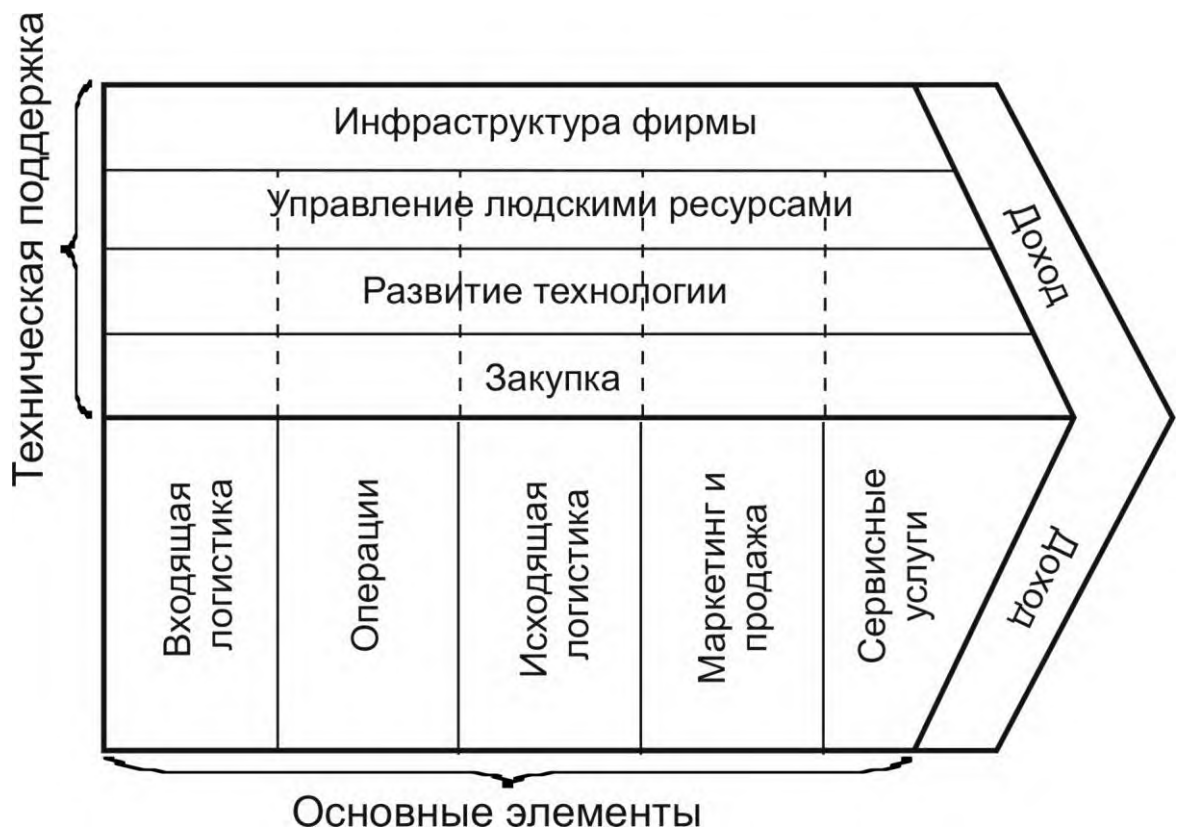


Рисунок 3.3 Фреймворк цепочек приращения стоимости.

К технической поддержки относятся следующие функции.

- Закупки. Закупки сырья, сервис, запасные части, здания, машинное оборудование и т.д.
- Развитие технологии. Включает развитие технологий для поддержки действий в цепочке приращения стоимости. Такие как: Научные исследования и разработки, автоматизация процессов, дизайн, изменение дизайна.
- Управление людскими ресурсами. Действия, связанные с рекрутингом, развитием (образованием), удерживанием и компенсацией сотрудникам и менеджерам.
- Инфраструктура фирмы. Включает общее руководство, менеджмент планирования, правовой, финансовый аспект, бухгалтерский учет, общественные дела, управление качеством и т.д.

Фирма может создавать преимущество по издержкам или путем снижения затрат индивидуальных элементов цепочки приращения стоимости, или путем реконфигурирования цепочки приращения стоимости.

Портер определил 10 факторов затрат, относящихся к действиям цепочки приращения стоимости:

- экономия от масштаба
- обучение.
- использование производственных мощностей.
- связи между действиями.
- взаимоотношения между бизнес группами.
- степень вертикальной интеграции.
- время входа на рынок.
- корпоративная политика в отношении затрат или дифференцирования.
- географическое положение.
- институциональные факторы (регулирование, профсоюзная деятельность, налоги и т.д.).

Для того, чтобы определить влияния информационных технологий на деятельность предприятия, с точки зрения VCF, необходимо проанализировать ее цепочку создания стоимости. Поскольку каждая операция связана с появлением, обработкой и передачей информации, информационные технологии играют решающую роль в формировании цепочки создания стоимости /47/.

Портер и Миллар /39/ выделяют пять шагов, которые должна предпринять организация, чтобы воспользоваться преимуществами, предоставляемыми ИТ.

1. Оценить информационную емкость продуктов и процессов.
2. Оценить роль ИТ в отраслевой структуре.

3. Выявить и ранжировать способы, с помощью которых ИТ создает конкурентное преимущество.
4. Рассмотреть, каким образом ИТ может создать новое направление в бизнесе.
5. Разработать план, направленный на извлечение выгод от использования ИТ.

Подход построение EA (Enterprise Architecture – архитектуры предприятия)

Создание архитектуры предприятия обосновывается принципом стратегическое развитие технологий и ИТ процессов с целью обеспечения поддержки бизнеса. Это, в свою очередь, делает ИТ динамично организованной структурой предприятия, реализующая успешную бизнес – стратегию /48/.

Архитектура обеспечивает стратегический контекст развития ИТ системы в ответ на постоянно изменяющиеся потребности окружающей бизнес - среды. Хорошая (эффективная) архитектура предприятия позволит достигнуть правильного баланса между эффективностью ИТ и новшествами в бизнесе.

Технические преимущества, которые следуют из хорошей архитектуры предприятия, приносят значимые практические результаты:

- более эффективны действия ИТ:
 - низкая стоимость разработки программного обеспечения, поддержки и обслуживания;
 - увеличенная мобильность приложений;
 - улучшенная интероперабельность и удобство управлением системой и сетью;
 - совершенствование возможности решать проблемы масштаба предприятия, подобно безопасности;
 - более легкая модернизация и обмен компонентами системы;

- лучшую отдачу существующих инвестиций, уменьшение риска будущих инвестиций:
 - уменьшенная сложность ИТ инфраструктуры;
 - максимальный возврат инвестиций в существующую ИТ инфраструктуру;
 - гибкость в выборе решения в ИТ: покупка, выполнение или аутсорсинг;
 - полное устранение риска в новых инвестициях и издержек использования ИТ;
- более быстрое, простое и более дешевое приобретение:
 - более простые решения по закупкам, т.к. информация, влияющая на приобретение, доступна в последовательном плане;
 - процесс приобретения осуществляется быстрее, т.к. максимизирована скорость и гибкость в приобретении, без увеличения издержек в архитектуре.

Обычно рассматриваются четыре вида "архитектуры", которые обычно принимаются как подмножества полной архитектуры предприятия:

- бизнес - архитектура;
- архитектура данных;
- архитектура приложений;
- технологическая архитектура.

Комбинация архитектуры данных и архитектуры приложений называется архитектурой ИС.

В качестве примеров архитектурных фреймворков можно привести TOGAF и методологию Захмана (Zachman Framework).

TOGAF

TOGAF (The Open Group Architecture Framework) - является методологией разработки и построения архитектуры предприятия, состоящей из сово-

купности детальных методов и ряда инструментов поддержки, созданных с целью. Данная методология разработана консорциумом OMG (Object Management Group), которая владеет всеми правами на этот продукт. В силу этого, TOGAF может использоваться свободно любой организацией, желающей разработать архитектуру своего предприятия.

Методология предоставляет цикл разработки архитектуры ADM (Architecture Development Method - Метод построения архитектуры) и "виртуального репозиторий" (Континуума предприятия) всех активов архитектуры, как предприятия, так и в целом ИТ индустрии, куда входят: модели, шаблоны, описание архитектуры и других артефакты, которые требуются при построении архитектуры предприятия.

Сам TOGAF предоставляет две эталонные модели, которые могут быть включены конкретной организацией в Континуум предприятия: базовая архитектура TOGAF (TOGAF Foundation Architecture) и эталонная модель объединенной информационной инфраструктуры (Integrated Information Infrastructure Reference Model - III-RM).

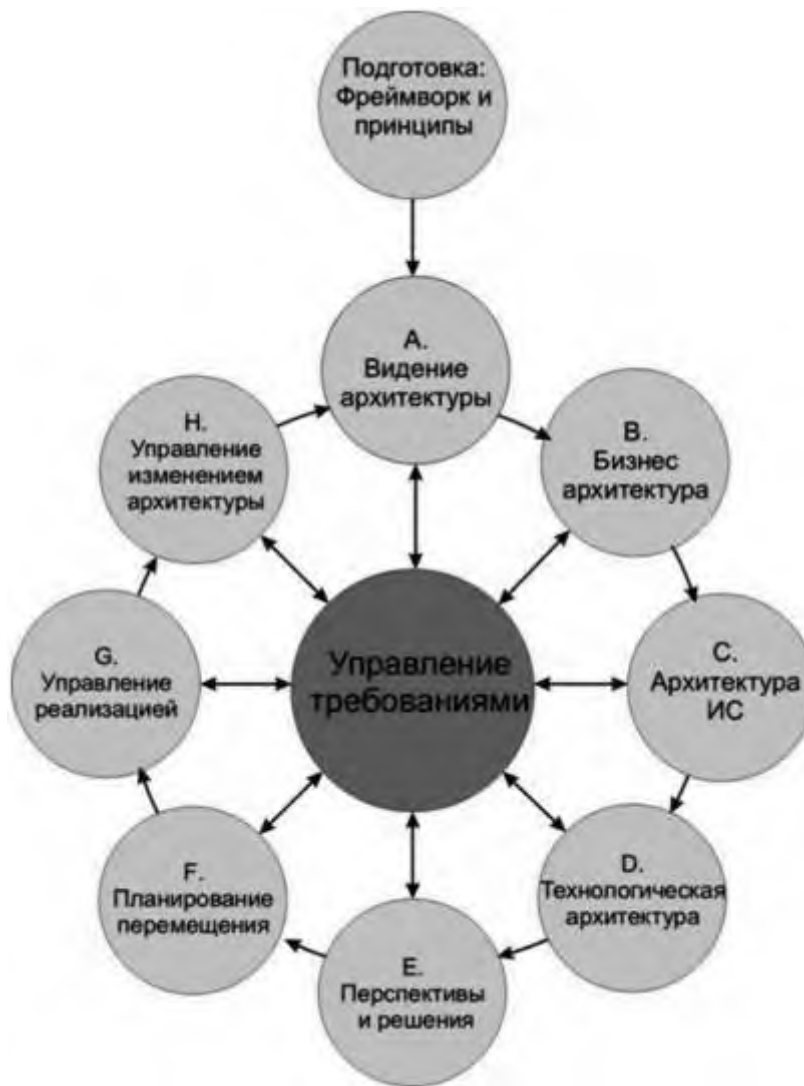


Рисунок 3.4 – Цикл ADM.

Базовая архитектура TOGAF включает:

- TRM (Technical Reference Model – Техническая эталонная модель) универсальных служб и функций, которая предоставляет основу для построения более специализированных архитектур и архитектурных компонентов;
- информационная база стандартов (Standards Information Base - SIB) - информационное ядро соответствующих спецификаций и стандартов.

В общем виде TRM можно представить (рисунок 3.5) в виде трех главных уровней (приложения, платформа приложений и коммуникационная ин-

фраструктура), соединенных двумя интерфейсами (интерфейс платформы приложений и интерфейс коммуникационной инфраструктуры).

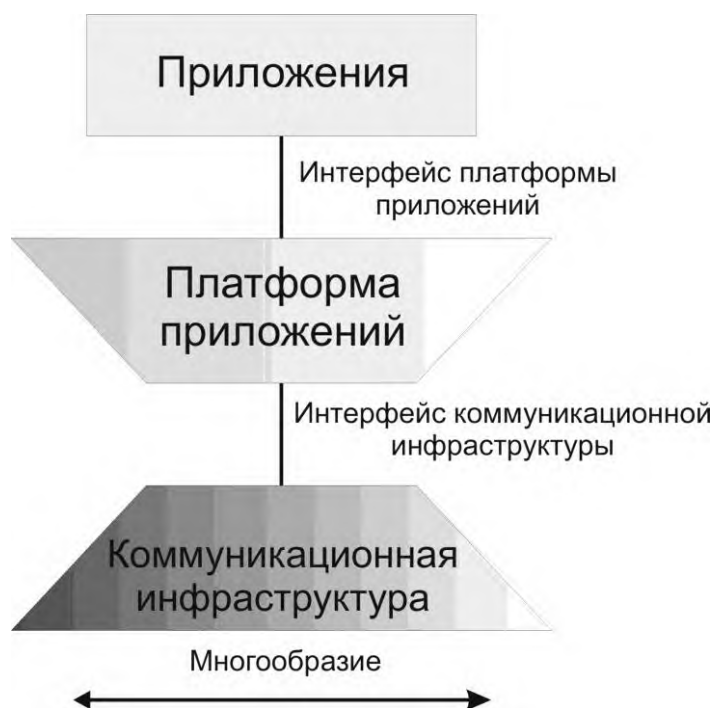


Рисунок 3.5 – TRM – представление высокого уровня.

TOGAF TRM идентифицирует универсальный набор сервисов платформы и обеспечивает классификацию (таксономию), которая объединяет эти сервисы по категориям, основываясь на схожести функциональных возможностей.

Эталонная модель интегрированной информационной инфраструктуры (III-RM) фокусируется на области программного обеспечения приложений. Это дает возможность решить одну из ключевых задач, стоящих при построении архитектуры предприятия: создание интегрированной информационной архитектуры.

Framework Zachman Enterprise Architecture /49-51/

Фреймворк архитектуры предприятия, разработанный Дж. Захманом (John A. Zachman) (иногда просто называют "Модель Захмана"), стал фактическим стандартом классификации и разработки артефактов архитектуры

уровня предприятия. Он построен на основе таких дисциплин, как классическая архитектура, конструирование, инженерия и производство.

В организации модели определяется два измерения (Рисунок 3.6):

- ролевая точка зрения участников проекта создания архитектуры:
 - планировщик, владелец, дизайнер, конструктор, разработчик
- абстракции архитектуры:
 - Что? Как? Где? Кто? Когда? Почему?

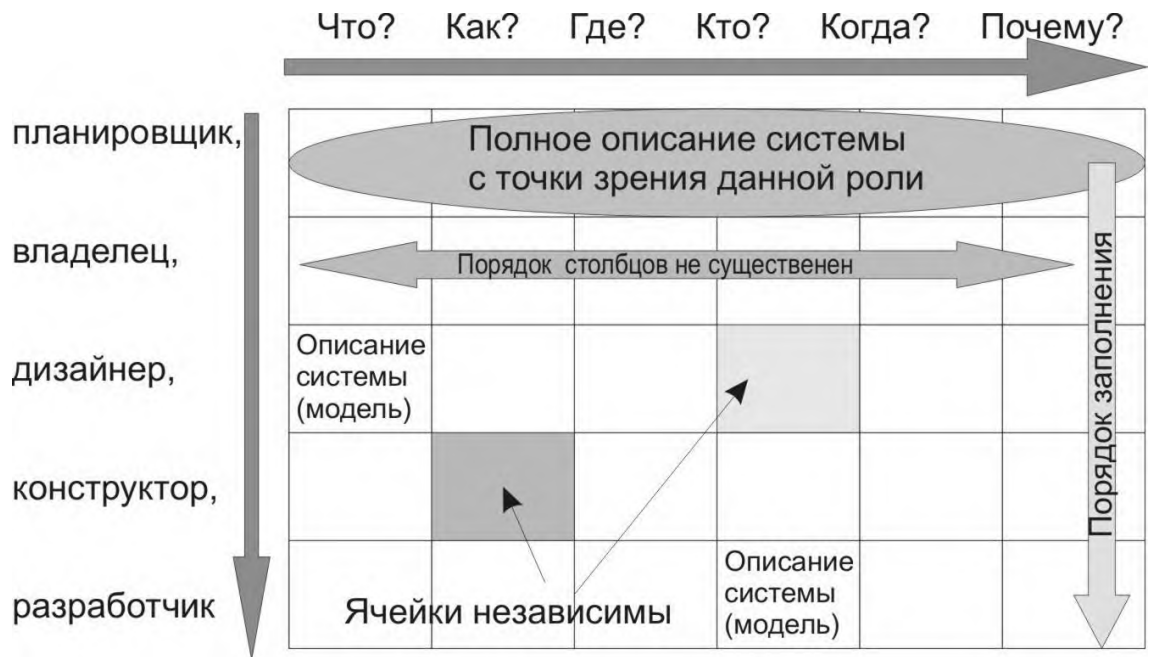


Рисунок 3.6 – Организация модели Захмана.

На рисунке 3.7 и 3.8 показан пример реализации модели Захмана. Конкретные модели и атрибуты представления показаны в качестве примера. В каждом конкретном случае представление, отвечающее вопросу столбца, может быть иным.




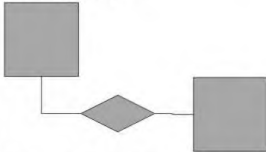
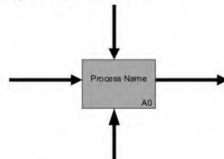
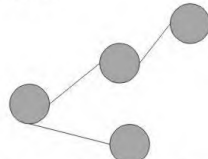
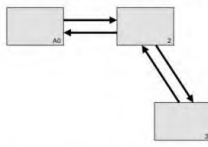
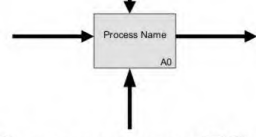
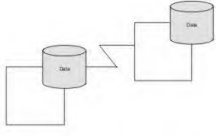
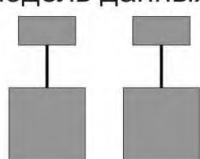
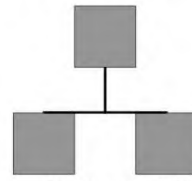




	Данные Что	Функция Как	Сеть Где
<p>Границы (зависит от контекста)</p> <p><i>Планировщик</i></p>	<p>Список важных положений бизнеса</p>  <p>Ent= класс положений бизнеса</p>	<p>Список процессов на предприятии</p>  <p>Proc= класс процессов бизнеса</p>	<p>Территориальное расположения</p>  <p>Node= основные расположение бизнеса</p>
<p>Модель предприятия (зависит от контекста)</p> <p><i>Владелец</i></p>	<p>Семантическая модель</p>  <p>Ent= Бизнес сущн. Reln=Бизнес связи</p>	<p>Модель бизнес процессов</p>  <p>Proc= бизн. процесс I/O=Бизнес ресурс</p>	<p>Логистическая сеть</p>  <p>Node=местоположен. Link=взаимосвязь</p>
<p>Модель системы (логическая)</p> <p><i>Дизайнер</i></p>	<p>Логическая модель данных</p>  <p>Ent= сущности Reln= связи</p>	<p>Архитектура приложений</p>  <p>Proc=процесс ПО I/O=вид пользоват.</p>	<p>Архитектура распределенных систем</p>  <p>Node=I/S функции Link=св-ва связи</p>
<p>Технологическая модель (физическая)</p> <p><i>Конструктор</i></p>	<p>Физическая модель данных</p>  <p>Ent=сегмент табл. Reln=Точка/ключ</p>	<p>Проект системы</p>  <p>Proc=Функции ПК I/O=формат устройств</p>	<p>Архитектура системы</p>  <p>Node="железо"/ПО Link= описание сети</p>
<p>Детализированное представление (не зависит от контекста)</p> <p><i>Разработчик</i></p>	<p>Определение данных</p>  <p>Ent=поле Reln=адрес</p>	<p>Программа</p>  <p>Proc=язык I/O=блок управлен</p>	<p>Архитектура сети</p>  <p>Node=адрес Link= протокол</p>
<p>Действующее предприятие</p>	<p>Данные</p>	<p>Функция</p>	<p>Сеть</p>

Рисунок 3.7 - Левая часть матрицы Захмана



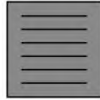
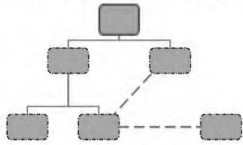
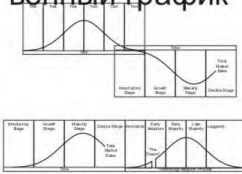
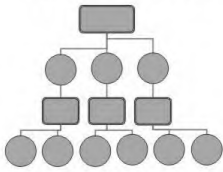
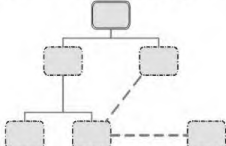
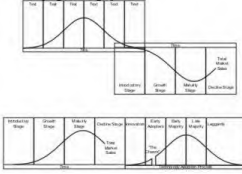
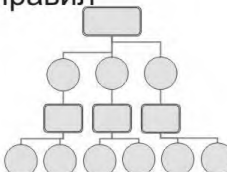
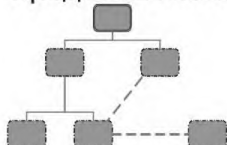
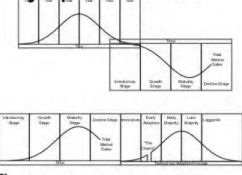
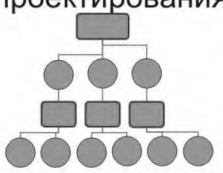



Люди Кто	Время Когда	Мотивация Почему	
<p>Список важных организаций</p>  <p>People= основные организации</p>	<p>Список важных событий в бизнесе</p>  <p>Время= основные события в бизнесе</p>	<p>Список бизнес-целей/стратегии</p>  <p>End/means=гл.цели Бизн. критич. фактор успеха</p>	<p>Границы (зависит от контекста)</p> <p><i>Планировщик</i></p>
<p>Модель делопроизводства</p>  <p>People=Организац. един. Work=рабочий продукт</p>	<p>Гл. Производственный график</p>  <p>Time=Бизнес событие Cycle=Бизнес цикл</p>	<p>Бизнес план</p>  <p>End=бизнес цели Means=бизн. стратег.</p>	<p>Модель предприятия (зависит от контекста)</p> <p><i>Владелец</i></p>
<p>Архитектура пользовательских интерфейсов</p>  <p>People=роли Work=комплект уз.</p>	<p>Технологическая структура</p>  <p>Time=Системн. событие Cycle=технолог. цикл</p>	<p>Модель бизнес правил</p>  <p>End=Структур. прав. Means=прав. действий</p>	<p>Модель системы (логическая)</p> <p><i>Дизайнер</i></p>
<p>Архитектура представлений</p>  <p>People=пользователь Work=формат экрана</p>	<p>Структура управления</p>  <p>Time=выполнение Cycle=цикл кмпоненты</p>	<p>Правила проектирования</p>  <p>End=Состояние Means=Действия</p>	<p>Технологическая модель (физическая)</p> <p><i>Конструктор</i></p>
<p>Архитектура безопасности</p>  <p>People=подлинность Work=работа</p>	<p>Определение временных диаграмм</p>  <p>Time=прерывания Cycle=маш. цикл</p>	<p>Спецификация правил</p>  <p>End=Подсостояние Means=Шаг</p>	<p>Детализированное представление (не зависит от контекста)</p> <p><i>Разработчик</i></p>
Организация	Графики	Стратегия	Действующее предприятие

Рисунок 3.8 - Правая часть матрицы Захмана

Комбинация точек зрения и описаний, представленных в тридцати ячейках таблицы Захмана, представляет собой мощную систематизацию, на основе которой можно выстроить полную архитектуру для разработки ИС. Расположенные по вертикали ракурсы могут отличаться степенью детализации, но, что более важно, они отличаются по существу и используют различные представления модели. Различные модели отражают различные взгляды участников. Аналогично расположенные по горизонтали описания подготовлены, исходя из различных соображений. Каждое из этих описаний призвано ответить на один из шести вопросов /39/.

Подход, основанный на знании предметной области

Требования, основанные на знании предметной области, выявляются посредством изучения стандартов профессиональной и учебной литературы относящейся к этой области. Изучение существующих ИТ решениях и существующих ИС для данной предметной области также может стать богатым источником знаний.

Для того, что бы определиться с видом используемого решения можно воспользоваться классификацией ИС по типу бизнес процессов, предложенная компания "Институт типовых решений - Производство" /52/. По типу бизнес процессов информационные системы, участвующие в автоматизации управления можно разделить на следующие группы:

- ИС стратегического маркетинга и управления предприятием;
- финансовые ИС;
- ИС управления производством;
- ИС управления запасами и сбытом;
- ИС поддержки жизненного цикла продукции.

Внутри каждой группы информационные системы подразделяются, как правило, в зависимости от реализованной концепции или алгоритма. Многие информационные системы реализуют концепции, закрепленные в стандартах (ISO, APICS и др.), или же реализуют алгоритмы, широко используемые в

бизнес процессах, или выделены аналитическими и консалтинговыми агентствами (Gartner, IDC и др.).

Используемая классификация не лишена недостатков. Во-первых, многие информационные системы входят в состав информационной системы, которая интегрирует несколько функций, при этом интегрирование может затронуть информационные системы, позиционированные в различных разделах. Например, Business intelligence (BI) включают Data mining (DM) и OLAP (online analytical processing). Во-вторых некоторые информационные системы не имеют четко разделенных границ. Например, в литературе часто считают, что разделение информационных систем на ERP (Enterprise Resource Planning) системы и MRP II (Manufacturing Resource Planning) системы имеет явно выраженный рекламный характер.

Подходы к разработке программного обеспечения

Структурный подход.

В среде DOS преобладала процедурная логика программного обеспечения, разработанная на языках типа Pascal и Basic. После запуска программа выполняется до завершения в жестко заданной последовательности шагов.

На некоторых этапах программа затребовала ввод исходных данных, которые могли вводиться в интерактивном режиме, что не изменяла детерминированный характер, контролируемый программой. Для этого стиля программирования хорошо зарекомендовал структурный подход в разработке ПО.

При появлении графического пользовательского интерфейса и многозадачной среды выполнения программы все кардинально изменилось. Программа управляется событиями, которые могут возникать случайным образом, например, действиями пользователя (клик мышью, нажатие клавиатуры и т.д.). В среде графического интерфейса, скорее пользователь контролирует выполнение программы, а не программа контролирует действия пользователя.

За каждым событием стоит программный объект, который генерирует или обрабатывает это событие. Современные системы на основе графического интерфейса требуют применение объектно-ориентированных языков программирования, что в свою очередь привело к развитию объектно-ориентированного подхода к разработке программного обеспечения.

В тоже время применение структурного подхода может быть оправдано при разработке современных систем в следующих случаях:

- разработчик не имеет опыт использования объектно-ориентированных парадигм при разработке ПО;
- предполагается разработка с «чистого листа» без поддержки повторного использования существующих компонент;

Структурный подход (structured approach) к разработке систем получил широкое распространение (и был признан стандартом де-факто) в 1980-х годах. Этот подход основан на двух методах: диаграммах потоков данных (data flow diagrams - DFD) для моделирования процессов и диаграммах сущность-связь (entity relationship diagrams - ERD) для моделирования данных.

Структурный подход является функционально-ориентированным и рассматривает DFD-диаграммы в качестве движущей силы разработки ПО /39/. Для описания бизнес процессов можно использовать IDEF0 модели. Сочетание DFD- и ERD-диаграмм дает относительно полные модели анализа, которые фиксируют все функции и данные системы на требуемом уровне абстракции независимо от особенностей аппаратного и программного обеспечения. Затем модель анализа преобразуется в проектную модель, которая обычно выражается в понятиях реляционных баз данных. После этого следует этап реализации /39/.

Не смотря на существенные недостатки структурно подхода, рекомендуется использовать его в особенности тем студентам, которые испытывают значительные затруднения в применении объектно-ориентированного подхода.

Можно предложить следующую последовательность реализации структурного подхода.

1. Изучение предметной области и выявление требований к информационной системе. Описание бизнес процесса (возможно построение функциональной модели AS-IS), выявление документов бизнес процесса, определение проблем, определение функций ИС и выявление ограничений, обоснование разработки.
2. Построение концепции ее реализации и разработка технического задания на ее разработку.
3. Построение функциональной модели (ТО-ВЕ) выявление требований к обеспечивающим частям ИС.
4. Разработка базы данных (серверная часть системы).
5. Разработка пользовательского интерфейса (клиентская часть системы) и интеграция его с серверной частью.
6. Верификация и тестирование ПО.

В главе 4 приводится методология функционального моделирования в нотации DFD и методология проектирования базы данных основанная на ER модели. В шестой главе рассматривается использование структурного подхода при выполнении дипломной работы «Информационная система распределения нагрузки по кафедре».

Объектно-ориентированный подход

Объектно-ориентированный подход к разработке систем получил распространение в 1990-х годах. Ассоциация производителей ПО Object Management Group утвердила в качестве стандартного средства моделирования для этого подхода язык UML (Unified Modeling Language - Унифицированный язык моделирования), который на сегодняшний день поддержан всеми ведущими разработчиками ПО.

По сравнению со структурным подходом объектно-ориентированный подход в большей степени ориентирован на данные, так как он развивается вокруг моделей классов. Использование разработки основанной на применении прецедентов смещает акценты от данных к функциям.

Объектный подход реализует технические преимущества, такие как абстракция, инкапсуляция, повторное использование, наследование, передача сообщений, полиморфизм и т.д.

В тоже время можно выделить существенные проблемы использования объектно-ориентированной парадигмы /39/.

- Этап анализа проводится на еще более высоком уровне абстракции, и - если серверная часть решения по реализации предполагает использование реляционной базы данных - семантический разрыв между концепцией и ее реализацией может быть значительным.
- Управление проектом сложно осуществлять. Менеджеры измеряют степень продвижения разработки с помощью четко определенной декомпозиции работ, элементов комплекта поставки и ключевых этапов. При объектной разработке с помощью «детализации» не существует четких границ между этапами, а проектная документация непрерывно развивается.
- Применение объектного подхода связано с возрастающей сложностью решений, что, в свою очередь, сказывается на таких характеристиках ПО, как приспособленность к сопровождению и масштабируемостью.

В главе 5 приводятся основы использования объектно-ориентированной парадигмы, а в шестой главе демонстрируется использование объектно-ориентированного подхода при выполнении дипломной работы «Информационная система поддержки дистанционного проведения практических и индивидуальных занятий».

Выбор распределенной архитектуры.

Архитектурное проектирование связано с выбором стратегии решений и представлением модульной структуры системы. Стратегия решения призвана разрешить проблемы, связанные с построением клиентской и серверной частей системы, а ПО промежуточного слоя (middleware) необходимо для "склеивания" клиента и сервера. Решение по основным строительным блокам (модулям) только отчасти зависит от выбранной стратегии решения.

Клиент и сервер - логические понятия. Клиент - это вычислительный процесс, который осуществляет запросы к процессу сервера. Сервер - это вычислительный процесс, который обслуживает запросы сервера. Обычно процессы клиента и сервера выполняются на разных компьютерах, но вполне возможно реализация системы «клиент/сервер» на одной машине.

В типичном сценарии клиентский процесс отвечает за управление отображением информации на экране пользователя и за обработку событий, инициированных пользователями. Процесс сервера - это любой компьютерный узел с базой данных, из которой данные могут быть запрошены клиентским процессом.

Архитектуру клиент/сервер можно расширить для представления произвольной распределенной системы. Любой компьютерный узел с базой данных может играть роль клиента в одних операциях, а сервера - в других операциях. Соединение подобных узлов с помощью сети связи дает начало архитектуре системы распределенной обработки.

В системе распределенной обработки клиент может осуществлять доступ к любому количеству серверов. Однако клиенту может быть разрешен доступ одновременно только к одному серверу. Это значит, что он может быть не в состоянии объединить данные от двух или более серверов баз данных в одном запросе. Если это возможно, то архитектура поддерживает систему распределенных баз данных.

Трехзвенная архитектура

Трехзвенная архитектура хорошо укладывается в концепцию подхода MVC и ВСЕ.

MVC (Model-view-controller «Модель-представление-поведение», «Модель-представление-контроллер») — архитектура программного обеспечения, в которой модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента, так, что модификация одного из компонентов оказывает минимальное воздействие на другие компоненты /40/.

Шаблон MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента

Модель (Model). Модель предоставляет данные (обычно для View), а также реагирует на запросы (обычно от контролера), изменяя свое состояние.

Представление (View). Отвечает за отображение информации (пользовательский интерфейс).

Поведение (Controller). Интерпретирует данные, введенные пользователем, и информирует модель и представление о необходимости соответствующей реакции.

Подход ВСЕ (Boundary-Control-Entity - граница-управление-сущность) представляет собой подход к объектному моделированию, основанный на трехфакторном представлении классов. В языке UML на классах определены три стереотипа: boundary (граница), control (управление) и entity (сущность).

- Пограничные классы (boundary class) описывают объекты, которые представляют интерфейс между актором и системой. Они выделяют часть состояния системы и представляют ее пользователю в форме визуального отображения или звукового эффекта. Пограничные классы соответствуют классам, представленным в разработанном GUI-интерфейсе. Погра-

ничные объекты часто сохраняются после однократного выполнения программы.

- Управляющие классы (control class) описывают объекты, которые перехватывают входные события, инициированные пользователем, и контролируют выполнение бизнес-процесса. Управляющий класс представляет действия и виды деятельности прецедентов. Управляющие объекты зачастую не сохраняются после выполнения программы.

- Классы-сущности (entity class) описывают объекты, которые представляют семантику сущностей, принадлежащих проблемной области. Они соотносятся со структурами данных системной базы данных. Объекты-сущности всегда сохраняются после выполнения программы и участвуют во многих прецедентах.

В верно спроектированной иерархии классов актер может взаимодействовать только с пограничными объектами, объекты-сущности могут взаимодействовать только с управляющими объектами и управляющие объекты могут взаимодействовать с объектами любого типа (рисунок 3.10).

Каждому пограничному объекту, который допускает взаимодействие, должен быть поставлен в соответствии связанный с ним управляющий объект. Поэтому некоторые типы архитектуры ПО объединяют две функции в одном классе. В качестве примера можно привести библиотеку VCL (Delphi).

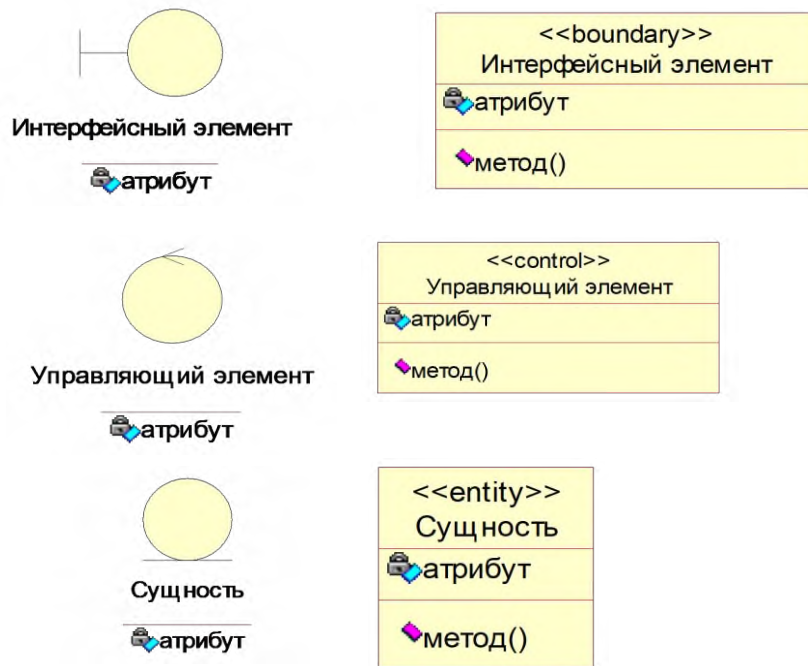


Рисунок 3.9 - Классы boundary (граница), control (управление) и entity (сущность) в нотации UML

Дополнительным преимуществом подхода ВСЕ является его увязка с трехзвенной моделью клиент/сервер, которая отделяет управление данными (объекты-сущности) от представления (пограничные объекты) посредством промежуточного слоя логики приложения (управляющие объекты). Сервер баз данных может даже быть реализован не на объектно-ориентированной, а на реляционной СУБД.

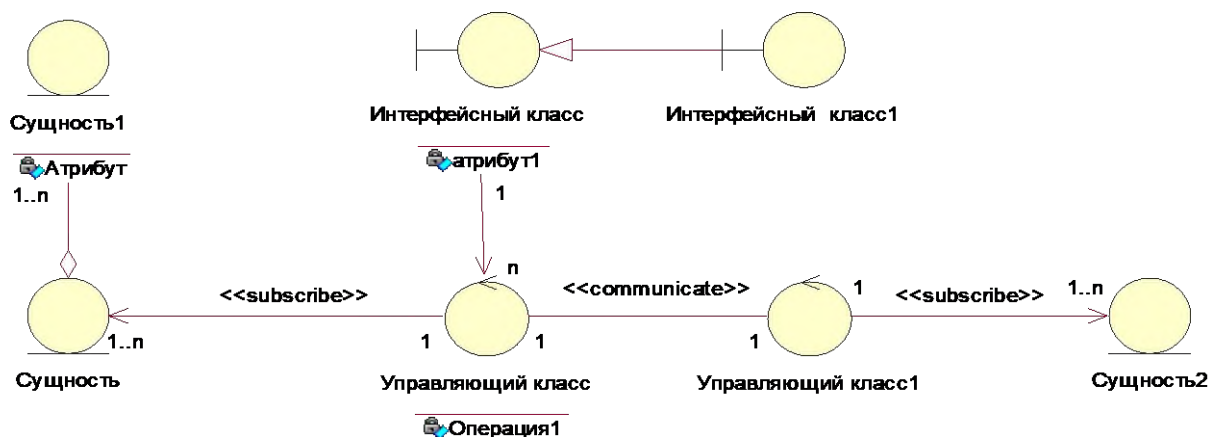


Рисунок 3.10- Пример реализации ВСЕ в диаграмме классов уровня анализа.

Необходимо, подчеркнуть, что классы-сущности располагаются в памяти клиентской программы, а не на сервере базы данных. Чтобы осуществить отображение классов-сущностей в реляционную память СУБД и для хранения другой релевантной информации о структурах баз данных, требуется отдельный уровень (слой) классов, которые могут быть организованы в виде пакета, предназначенного для работы с базой данных. Существует расширение подхода ВСЕ, названное Boundary-Control-Entity-Database (BCED-граница – управление – сущность - база данных). Классы уровня D (Database) ответственны за извлечение данных из базы данных и сохранение данных. К этому уровню классов можно отнести классы Delphi: TADODConnection, TADODTable, TADODQuery (при использовании технологии ADO).

Аналогично клиентскому и серверному процессу прикладной процесс представляет собой логическое понятие, которое может поддерживаться или не поддерживаться специально выделенным для этой цели аппаратным обеспечением. Логика приложения может с равным успехом выполняться на клиентском или серверном узле, т.е. может быть встроена в клиентский или серверный процесс и реализована в виде библиотек: DLL (Dynamic Link Library-динамически компокуемая библиотека), API-интерфейса (Application Programming Interface - интерфейс прикладного программирования), RPC вызовов (Remote Procedure Calls - удаленный вызов процедуры) и т.д.

Если логика приложения скомпилирована с клиентом, говорят об архитектуре толстого клиента. Если она скомпилирована с сервером, говорят об архитектуре тонкого клиента. Возможны также промежуточные архитектуры, в которых логика приложения частично скомпилирована с клиентом, а частично - с сервером.

Логику приложения можно также развернуть на отдельных вычислительных узлах, как показано на рисунке 3.11.

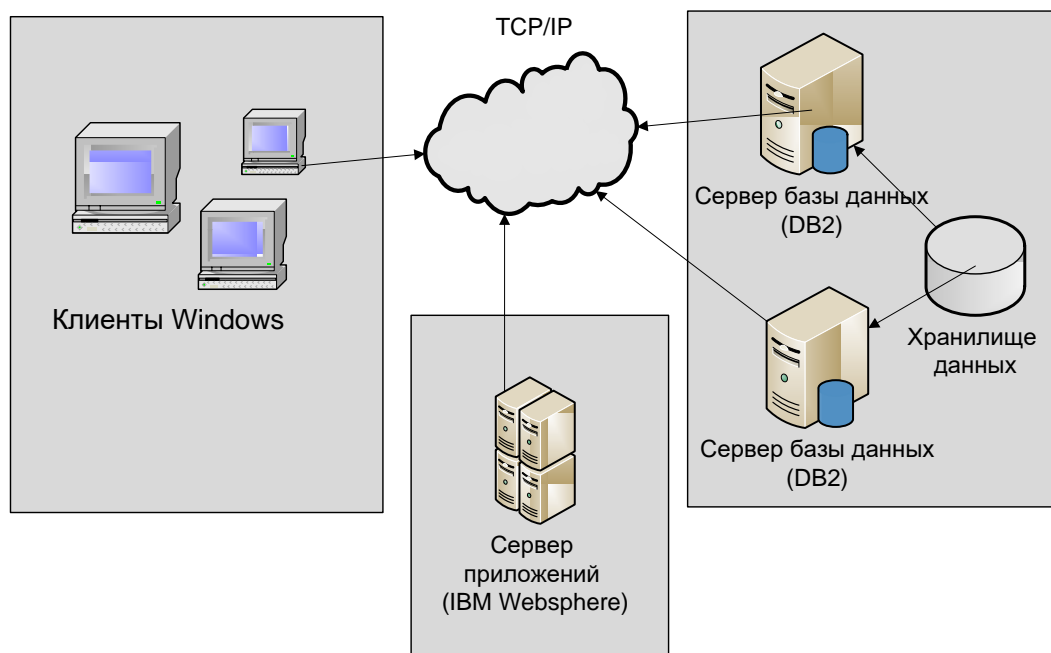


Рисунок 3.11 – Пример развертывания трехзвенной архитектуры /39/.

При реализации приложения необходимо решить вопросы реализации: пользовательского интерфейса, логики ответственной за обработку GUI интерфейса (презентационная логика), прикладные функции, логики ответственной за бизнес правила масштаба предприятия (интегральной логики), функции доступа к базе данных (рисунок 3.12).

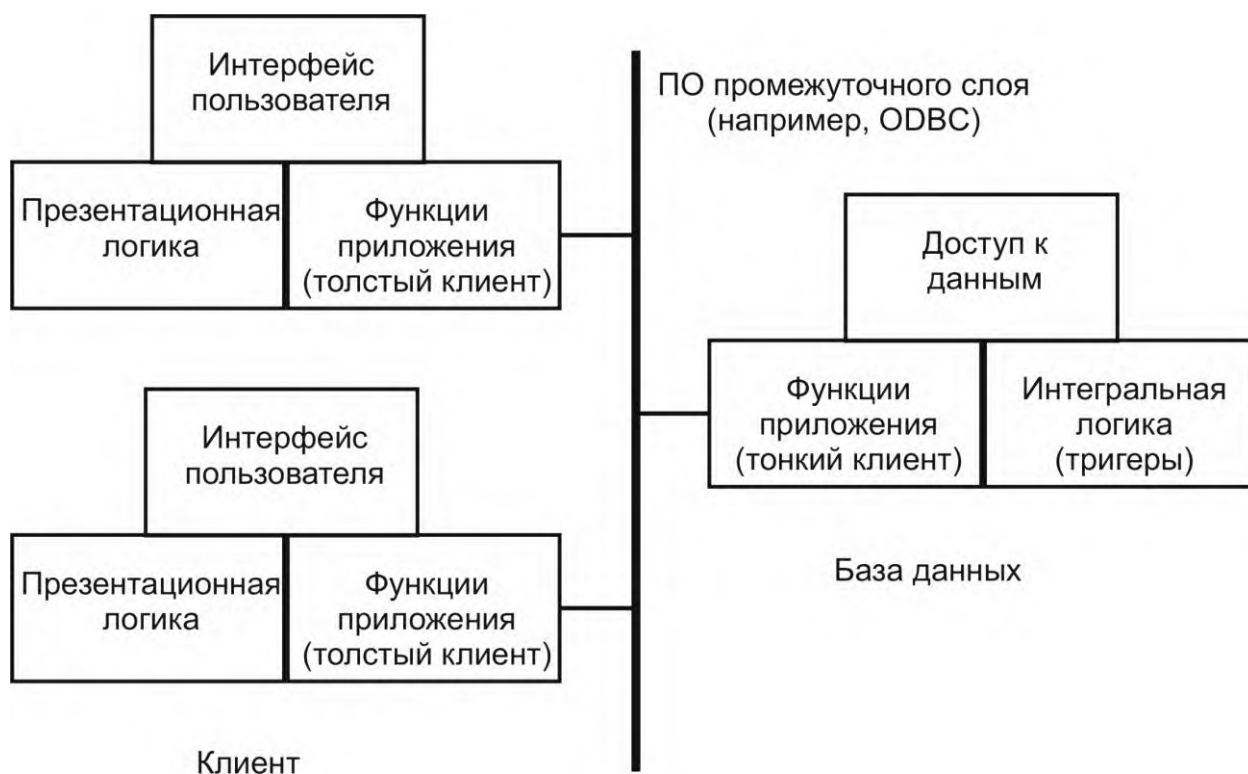


Рисунок 3.12 – Взаимодействие приложение – база данных /39/.

Программы баз данных называются хранимыми процедурами. Хранимые процедуры хранятся в самой базе данных (они являются постоянными или персистентными). Их можно вызвать из клиентской программы (или из другой хранимой процедуры) с помощью обычного оператора вызова процедуры/функции.

Существуют хранимые процедуры специального вида - триггеры, которые нельзя вызвать явно. Триггер срабатывает автоматически при попытке изменить содержимое базы данных. Триггеры используются для реализации бизнес-правил масштаба предприятия (интегральная логика).

Реализация Web- приложений /39/

Web-приложение - разновидность системы клиент/сервер с Web-узлом. Браузер Internet-клиента отображает Web-страницы на экране компьютера. Web-сервер доставляет Web-страницы браузеру. Web-страницы могут быть статическими (неизменяемыми) или динамическими. Web-страница может представлять форму, заполняемую пользователем. Чтобы пользователь мог одновременно просматривать несколько Web-страниц экрана разделяется приложением на фреймы.

Чтобы управлять логикой приложения и отслеживать состояние приложения, Web-приложение может включать сервер приложений. Обычный метод отслеживания состояния реализуется хранением в браузере куков (cookie - короткие символьные строки, представляющих состояние интерактивного пользователя).

Чтобы придать страницам, отображаемым на клиентской машине, динамичность, используются сценарии или апплеты. Сценарий или скрипты (script – может быть написан, например, на языке JavaScript) представляет собой программу, выполняемую браузером в режиме интерпретации. Апплет (applet) - это скомпилированная компонента, которая выполняется в контексте браузера, однако, имеет лишь ограниченный доступ к другим ресурсам клиентской машины (из соображений безопасности).

Web-страница может также включать сценарии, выполняемые сервером. Подобная страница называется серверной страницей (server page). Серверная страница имеет доступ ко всем ресурсам сервера баз данных. Серверные страницы управляют клиентскими сеансами, размещают cookies в среде браузера и строят клиентские страницы т.е. строят страничные документы из серверных бизнес-объектов и отправляют их назад клиенту.

Чтобы обеспечить доступ сценариев, содержащихся в серверных страницах, к базам данных, используются стандартные библиотеки доступа к данным. К типичным технологиям, позволяющим реализовать эту возможность, относятся ODBC (Open Database Connectivity - открытый интерфейс доступа к базам данных), JDBC (Java Database Connectivity- интерфейс доступа к базам данных Java-приложений), RDO (Remote Data Objects - интерфейс доступа к удаленным объектам), ADO (ActiveX Data Object - набор высокоуровневых интерфейсов, позволяющих разработчикам обращаться к данным на любом языке программирования на основе ActiveX).

Технологией, дающей возможность функционировать Web-серверу, являются содержащие сценарии страницы HTML - активные серверные страницы (Active Server Pages - ASP) или серверные страницы Java (Java Server Pages - JSP). Для создания Web-страниц можно использовать технологию написания клиентских сценариев (JavaScript или VBScript), документов XML (eXtensible Markup Language), Java-апплетов, управляющих элементов JavaBean или ActiveX

Для получения Web-страниц с Web-сервера клиенты используют протокол HTTP (Hypertext Transfer Protocol - протокол передачи гипертекстовых файлов). Страница может содержать сценарий или скомпилированные и непосредственно выполняемые модули DLL (Dynamic Link Library - динамически компокуемая библиотека), например, ISAPI (Internet Server Application Programming Interface - интерфейс прикладного программирования Internet-сервера), CGI (Common Gateway Interlace - Общий шлюзовый интерфейс) или Java-сервлеты.

Cookie играют роль примитивного механизма поддержки соединения между клиентом и сервером в системе, которая иначе называется Internet-системой без установления соединения. Более сложный механизм соединения клиента с сервером превращает Internet в распределенную объектную систему. В распределенной объектной системе объекты идентифицируются с помощью уникальных OID и взаимодействуют за счет получения OID друг друга. Главными механизмами при этом выступают технологии CORBA, DCOM и EJB. При использовании данных технологий объекты могут взаимодействовать без использования протокола HTTP или Web-сервера в качестве посредника.

Архитектура развертывания может представлять четырехзвенную структуру: клиентский Web-браузер, Web-сервер, сервер приложений, сервер баз данных.

Браузер клиентского узла можно использовать для отображения статических или динамических Web-страниц, которые могут включать сценарии и апплеты, загружаемые и выполняемые в рамках браузера. Клиентский браузер можно оснастить дополнительными функциональными возможностями, такими как элементы управления ActiveX или JavaBeans.

Web-сервер обрабатывает запросы на страницы, поступающие от браузера, и динамически генерирует страницы и программный код для выполнения и отображения на клиенте. Web-сервер также обеспечивает настройку и параметризацию сеансов работы пользователя.

Сервер приложений необходим в том случае, когда в реализации используются распределенные объекты. Он управляет бизнес-логикой. Бизнес-компоненты публикуют свои интерфейсы для других узлов через интерфейсы компонент, такие как CORBA, DCOM или EJB.

Бизнес-компоненты, инкапсулируют постоянные объекты, хранимые в базе данных, чаще всего в реляционной базе. Они взаимодействуют с сервером баз данных через протоколы связи с базами данных, например, такими

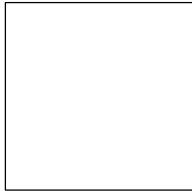


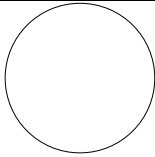




как JDBC или ODBC. Узел базы данных обеспечивает масштабируемое хранилище данных и многопользовательский доступ к нему.

4. Методологии структурного моделирования

Моделирование потоков данных /53/

К синтаксическим элементам DFD относятся объекты диаграммы и правила отображения их на диаграмме. К объектам диаграммы относятся: процессы, потоки, внешние сущности, накопители данных (хранилища). Существует две распространенные нотации DFD, представленные в таблице 4.1.

Таблица 4.1 - Изображение различных объектов DFD в нотациях Гейн-Сарсона и Йордона

Объект модели	Нотация Гейн- Сарсона	Нотация Йордона
Внешняя сущность		
Процесс		
Хранилище (накопитель данных)		
Поток данных		

Основные структурными элементами диаграммы являются процессы и потоки данных. Цель процесса состоит в преобразование входных потоков в выходные. Модель DFD состоит из трех типов документов: графические диаграммы, спецификация процессов и глоссарий (словарь данных).

Спецификация процесса определяет:

- входные и выходные потоки данных,
- предусловие и постусловие,
- последовательностью действий преобразующих входной поток в выходной.

Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, Проверить стаж работы) или отглагольное существительное (Проверка стажа работы).

Стрелки (потоки) не могут быть «висячими». Они начинаются и заканчиваются на объекте (объект может быть один и тот же). В качестве объекта могут быть все остальные элементы DFD. Поток должен иметь уникальное имя, которое идентифицирует его в словаре данных. Название потока должно быть именем существительным и кратко характеризовать данные потока (например, паспортные данные клиента). Потоки могут объединяться в другие потоки, в тоже время, данные в потоки могут быть объединены в пачки (адрес состоит из названия города, улицы и т.д.).

Хранилище (накопитель данных) позволяет выявить данные, которые будут сохраняться вне процессов. Данные, которые он содержит, может использоваться в любое время после ее определения (после поступления с входными потоками), при этом данные могут выбираться в любом порядке. Основное отличие накопителя от процесса то, что основная его цель хранить данные без изменения. Данные во входном и выходном потоке процесса обязаны быть отличными, накопителя – одинаковыми. При этом не уточняется способ помещения и извлечения данных в накопитель, нас не интересует, происходит ли извлечение данных для чтения (копирования) или для изъятия и другие подобные вопросы. Имя накопителя должно идентифицировать его содержимое и быть существительным во множественном числе. Накопитель должен иметь уникальный номер. Допускается использовать накопитель в разных диаграммах. Рекомендуется связь с накопителем показывать непосредственно на диаграмме, а не через родительскую диаграмму.

Так как в системе используются материальные потоки (материальные носители данных или информации), то соответственно может допускаться хранение этих носителей. Входные и выходные потоки хранилищ материальных носителей должны быть только материальные.

Внешняя сущность это объект вне контекста системы, являющаяся источником или приемником данных нашей системы (контекстного процесса), например: клиент, отдел кадров, Министерство образования и науки. Внешняя сущность, являясь объектом окружающей среды, определяет границу нашей системы. Нас интересует взаимодействие системы с окружающей средой, но не интересует взаимодействие объектов окружающей среды. Не допускаются потоки между внешними сущностями.

В DFD не определяется понятие узла диаграммы и соответственно не существует ссылочного кода, как в нотации IDEF0. Рекомендуется использовать правила построения, используемые в IDEF0 для следующих элементов:

- поля диаграммы (рамка диаграммы) и ссылочный код, при этом нужно использовать рамку, которую предоставляет применяемый CASE инструмент (Visio, BPWIN) ;
- потоки данных (в IDEF0 это правила относительно изображения стрелок)

Словарь данных требует применения формализованных языковых конструкций в связи с тем, что данные используются в алгоритмах и в электронных хранилищах. Ниже приведены возможные языковые конструкции, используемые системными аналитиками для организации словаря данных.

:=	Составлен из
+	и
()	дополнительный элемент(может присутствовать или отсутствовать)
{ }	повторение элементов

[]	альтернативные варианты
**	комментарии
@	ключевое поле в хранилище
	отдельные альтернативные элементы выбора в конструкции []

Так, например, определение адреса в потоке данных:

- адрес := название страны + (название федерального округа) + (название типа субъекта федерации) + (название субъекта федерации) + (название района субъекта федерации)+ название типа населенного пункта + название населенного пункта + (название района населенного пункта) + название типа объекта населенного пункта + (название объекта населенного пункта) + номер дома + (номер корпуса)+ (номер квартиры);
- название типа населенного пункта := [город | районный центр | поселок | село | рабочий поселок | и т.д];
- имя субъекта федерации := { слова }.

Потоки данных не отражают семантику данных. Рекомендуем, по возможности, иллюстрировать поток данных формой документа, с которой он ассоциирован. Так, например, при поступлении студента в университет он заполняет анкету. Поля этой анкеты образуют поток от внешней сущности «студент» к процессу, скажем «прием документов». В словаре данных вместе с формальным описанием потока данных на структурированном языке, описанном выше, требуется приложить (включить в приложение словаря и сослаться на него) форму анкеты студента.

Методика построения DFD диаграммы.

Разработка модели в DFD представляет собой пошаговую, итеративную процедуру. На каждом шаге итерации разработчик предлагает вариант модели, который подвергают обсуждению, рецензированию и последующему редактированию, после чего цикл повторяется.

Построение контекстной диаграммы. Задача контекстной диаграммы – определение границ системы (наша система - процесс). Для этого необходимо определить процесс (хотя бы его название) и идентифицировать все внешние сущности, с которыми он может взаимодействовать.

Определение информационных потоков между внешними сущностями и системой. Рекомендуется сразу идентифицировать потоки в глоссарии. Если количество потоков большое и мешает восприятию диаграммы, необходимо их объединить. Данные сгруппированные в пакеты не следует разъединять. В последующем структура информации- пакет, может обеспечить выявление транзакций баз данных.

Декомпозиция системы на процессы. При этом идентифицируются процессы, составляющие систему с их кратким описанием. На этом этапе не требуется создания спецификации процессов, а требуется только их выявить и согласовать со специалистом в предметной области. Рекомендуется не обращать внимание на то, что процессы могут находиться на разном уровне абстракции. На этой стадии преобладает в анализе движение сверху вниз.

Определение потоков данных между процессами. На этом этапе возможно группировка части процессов, или включения их в другой процесс. Не требуется акцентировать свое внимание на хранилищах. Необходимо добиться согласования потоков родительских и дочерних диаграмм. На этом этапе начинается проектирование снизу вверх.

Далее осуществляется группировка процессов со сходными задачами, и определяются хранилища. Продолжается проектирование снизу вверх (Рисунок 4.1)

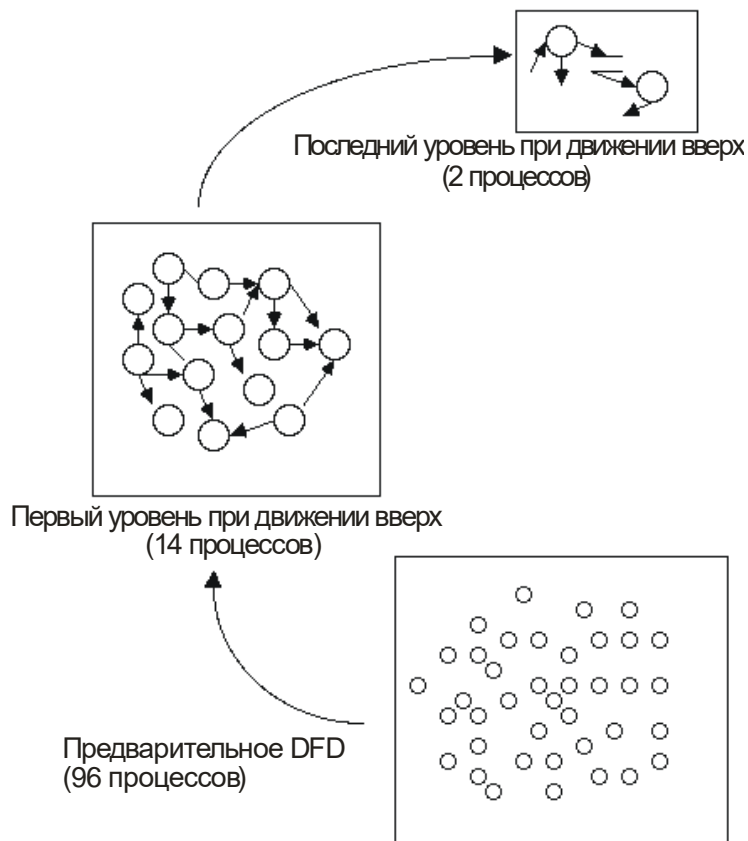


Рисунок 4.1 - Проектирование снизу вверх по Йордону.

На всех этапах требуется вести глоссарий и спецификацию процессов.

Как видите, такое проектирование постоянно контролирует количество процессов нижнего уровня, это препятствует лавинообразное размножение диаграмм. Декомпозиция и объединение процессов производится совместно с декомпозицией и объединение потоков.

Выявление хранилищ. Хранилища системы определяют после декомпозиции контекстной диаграммы (Рисунок 4.2).

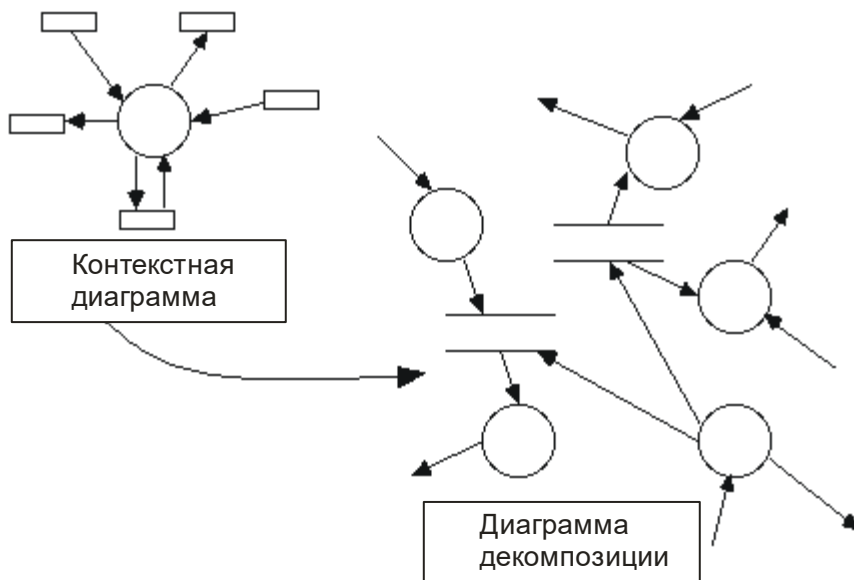


Рисунок 4.2 – Выявление хранилищ при движении сверху вниз

При группировке процессов по деятельности в ситуации, показанной на рисунке 4.3.

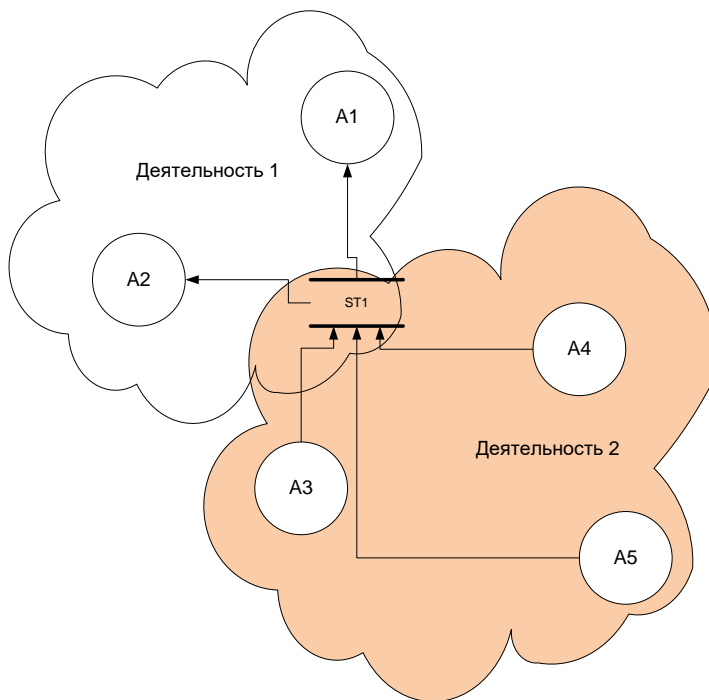


Рисунок 4.3- Группировка процессов

Взаимодействие их через хранилище необходимо производить, так как показано на рисунке 4.4

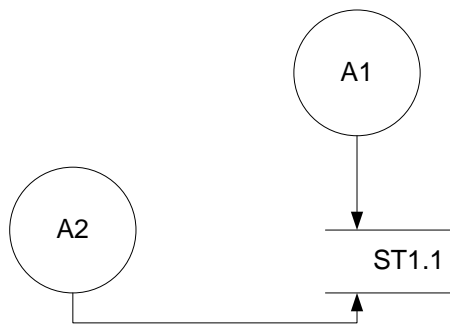


Диаграмма 1

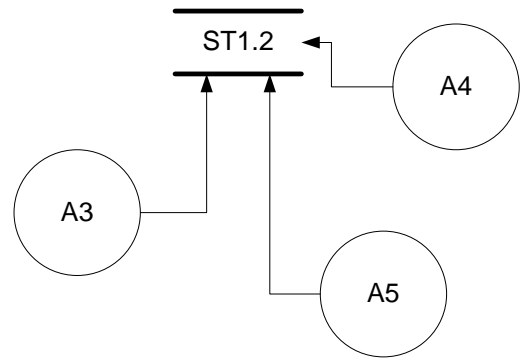
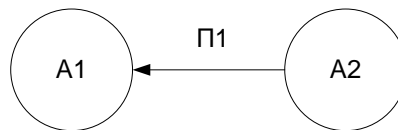


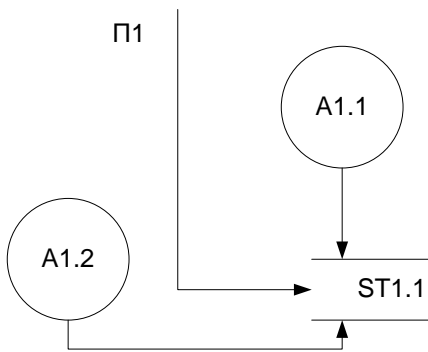
Диаграмма 2

Рисунок 4.4 – Представление одного и того же хранилища на различных диаграммах после группировки процессов.

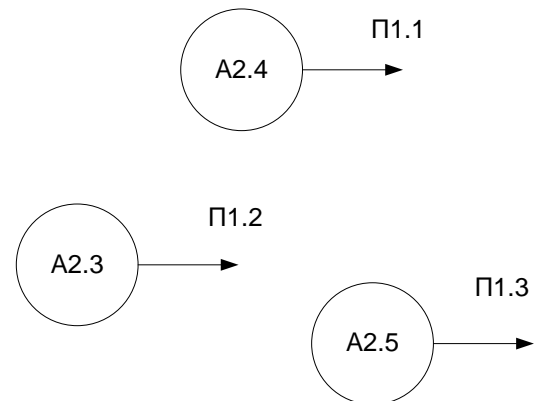
Одно и то же хранилище может присутствовать на различных диаграммах. Такое взаимодействие предпочтительнее связывание процессов, через граничные потоки, показанное на рисунке 4.5.



Родительская диаграмма



Дочерняя диаграмма 1



Дочерняя диаграмма 2

Рисунок 4.5- Взаимодействие процессов через родительскую диаграмму (не рекомендуется).

Моделирование данных /54-56/

Концептуальное²³ проектирование базы данных

Создание локальной концептуальной модели данных может выполняться на основании:

- описания предметной области с точки зрения хранилища DFD (спецификации процессов, сопряженных с хранилищем, и входных и выходных потоков данных хранилища),
- представления о предметной области каждого пользователя системы.

В первом случае схема данных используется для детализации хранилищ DFD. Во втором случае, для описания предметной области на стадии бизнес анализа.

Для построение концептуальной модели рекомендуется использовать ERD в нотации П. Чена /54,55/.

Нотация Чена

Тип сущности (далее сущность), это множество сущностей одного типа. Тип сущности отражает объектное множество²⁴ реального мира.

Экземпляр сущности является конкретным элементом типа сущности. В модели тип сущность представляется прямоугольником (Рисунок 4.6). Название типа сущности – существительное. Экземпляр сущности не имеет графического представления.

²³ Фаза концептуального проектирования базы данных начинается с создания концептуальной модели данных предприятия, полностью независимой от любых деталей реализации: выбранный тип СУБД, состав программ приложения, используемый язык программирования, конкретная вычислительная платформа и т.д.

²⁴ Объектное множество это множество объектов одного типа

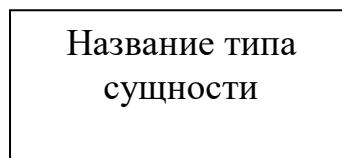


Рисунок 4.6 – Тип сущности.

Конкретизация - объектное множество, является подмножеством другого объектного множества. **Генерализация (обобщение)** - объектное множество, является надмножеством другого объектного множества. Представление обобщения конкретизации в ER модели приведено на рисунке 4.7.

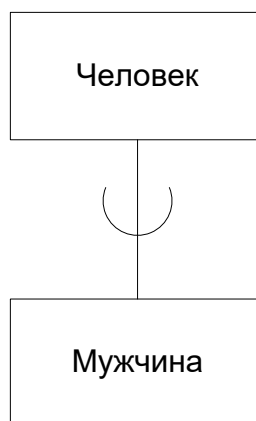


Рисунок 4.7 - Представление конкретизации и обобщения.

Необходимо отметить, что экземпляр сущности «Мужчина» одновременно является экземпляром сущности «Человек».

Отношением называется связь между объектами, принадлежащая разным объектным множествам. Под **типом связи** понимается осмысленную ассоциацию между сущностями разных типов. В ER модели **тип связи** (далее **связь**) изображается ромбом (рисунок 4.8). Тип связи называется глаголом или отглагольной формой. Направление связи не отражается на ER диаграмме, хотя может и присутствовать в порядке слов описывающих эту связь (связь «находится замужем» в языковых конструкциях имеет направление). Экземпляр связи - ассоциация между сущностями, включающая по одному экземпляру сущности из каждого участвующего в связи типа сущности.

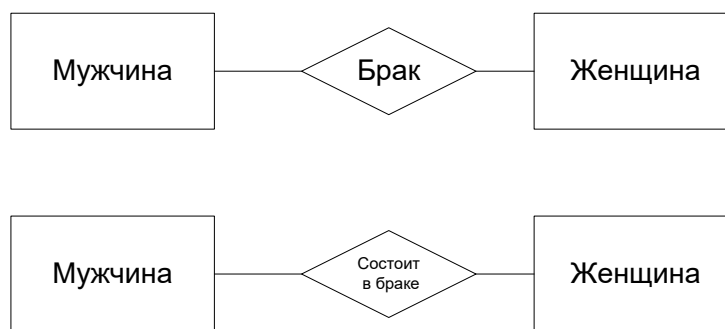


Рисунок 4.8 - Тип связи.

Основное отличие типа связи от типа сущности, то, что тип связи не существует без типа сущности, а тип сущности может быть не связан с другими сущностями.

Считается правомочным участием типа связи в другом типе связи, в качестве составной сущности («связь со связью»). Для того, чтобы диаграмма была читаема, рекомендуется именовать составную сущность, как это показано на рисунке 4.9.

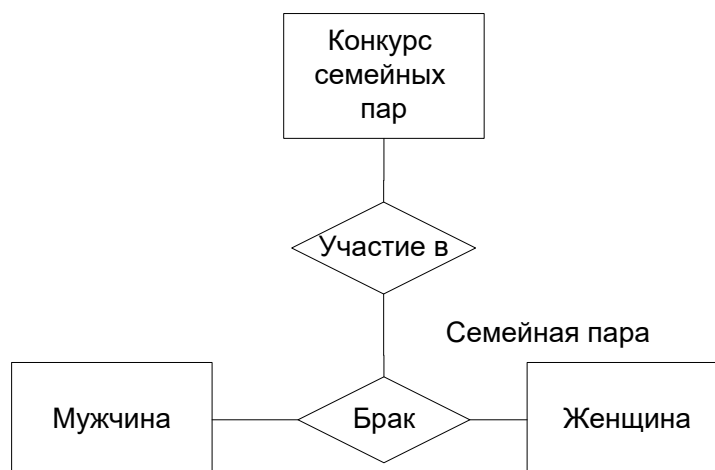


Рисунок 4.9 – Рекомендуемое отображение составной сущности (семейная пара) и ее участие в связи.

Атрибут - свойство типа сущности или типа связи. Набор значений, которые могут быть присвоены к атрибуту, называется **домен атрибута**.

Можно выделить следующие виды атрибутов (рисунок 4.10):

- простой атрибут - состоит из одного компонента данных (не зависит от других компонентов и не имеет собственной структуры);

- составной атрибут - состоит из нескольких компонентов, каждый из которых характеризуется независимым существованием;
- однозначный атрибут - содержит одно значение для одного экземпляра сущности;
- многозначный атрибут - содержит несколько значений для одного типа сущности;
- производный атрибут - представляет значение, производное от значения связанного с ним атрибута или некоторого множества атрибутов, принадлежащих некоторому (не обязательно данному) типу сущности.



Рисунок 4.10 – Различные виды атрибутов.

Сущности подразделяются на два вида (рисунок 4.11): слабые и сильные. **Слабый тип сущности** это отображение множества объектов реального мира, присутствие которых в данной модели зависит от объектов другой сущности. **Сильный тип сущности** отображение множества объектов реального мира, присутствие которых в данной модели не зависит от объектов другой сущности.

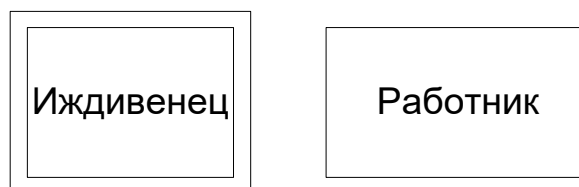


Рисунок 4.11- Изображение слабого и сильного типа сущности.

Тип связи между слабой и сильной сущностью слабый (рисунок 4.12).



Рисунок 4.12 – Слабый тип связи

Мощность связи (показатель кардинальности) обозначает максимальное количество элементов одного типа сущности связанных с одним элементом другого типа сущности. Наиболее распространенными являются бинарные связи с показателями кардинальности «один к одному» (1:1), «один ко многим» (1:M) и «многие ко многим» (M:N) (Рисунок 4.13).

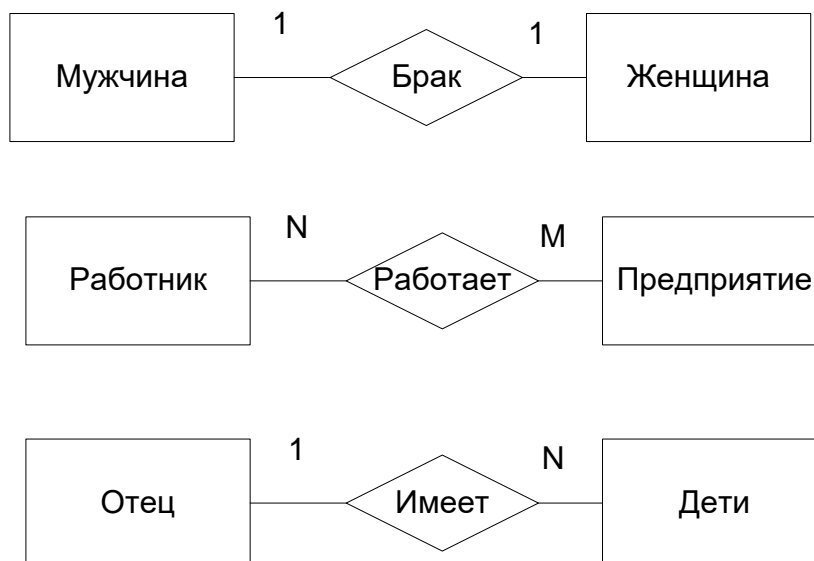


Рисунок 4.13 – Бинарные связи с различной мощностью

Степень связи количество сущностей, которые охвачены данной связью (рисунок 4.14). Такие типы связи называют n-арными. Мощность n-арной связи определять бессмысленно, так как мощность понятие бинарной связи. Обычно проставляют мощность много, обозначая возле каждой связи буквы N, M, P и т.д..

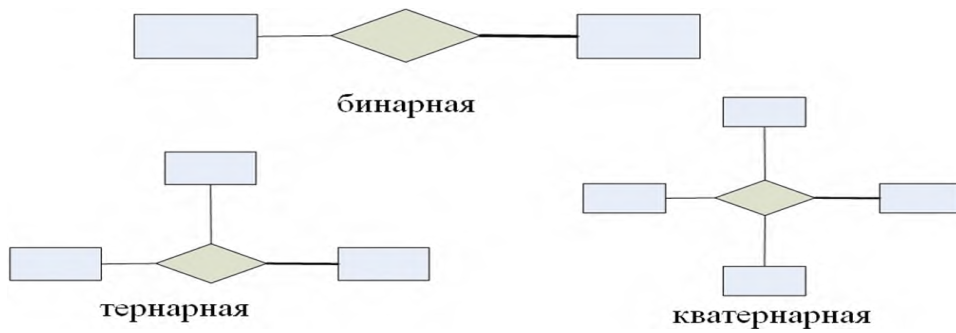


Рисунок 4.14 – Степень связи.

Рекурсивная связь - это ассоциацию объектов множества с другими объектами этого множества. Для лучшего понимания рекурсивной связи используется ролевые имена (рисунок 4.15).



Рисунок 4.15 – Рекурсивная связь.

Степень участия (рисунок 4.16) определяет, зависит ли существование некоторой сущности от участия в связи некоторой другой сущности. Та сущность, которая имеет **полное участие в связи**, представляет множество экземпляров сущности, каждое из которых находится в связи с каким-либо экземпляром другой сущности. С другой стороны не может быть определен экземпляр сущности, для которого нет пары из числа экземпляра другой сущности. **Частичное участие** предполагает отрицание этого ограничения.

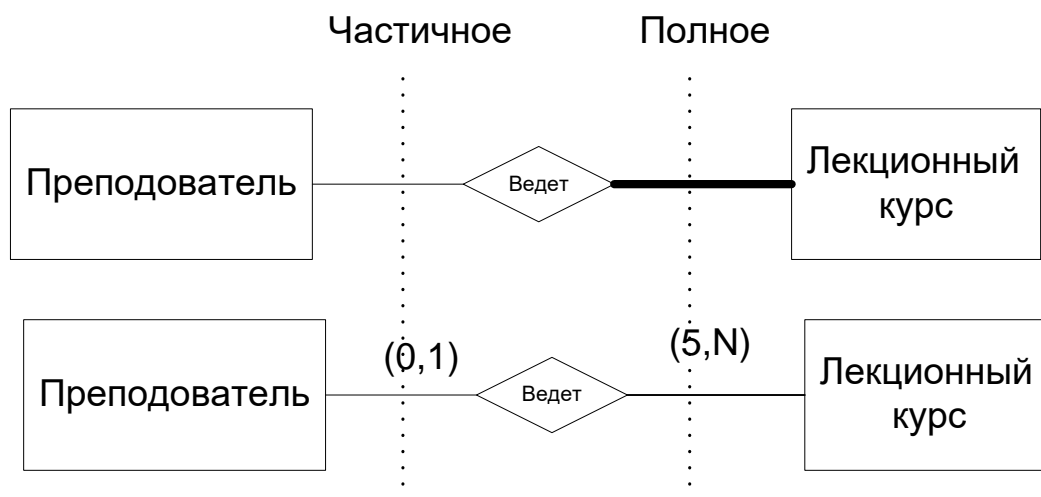


Рисунок 4.16 – Два способа отображения степени участия.

Особый вид отношений генерализация/ специализация (для краткости будем называть генерализация) давал возможность воплотить парадигму ООП в ER модели. В ERD (Enhanced Entity-Relationship) генерализация дает возможность в суперклассе собрать все атрибуты общие для нескольких подклассов, а в подклассах оставить только специальные атрибуты (рисунок 4.17).

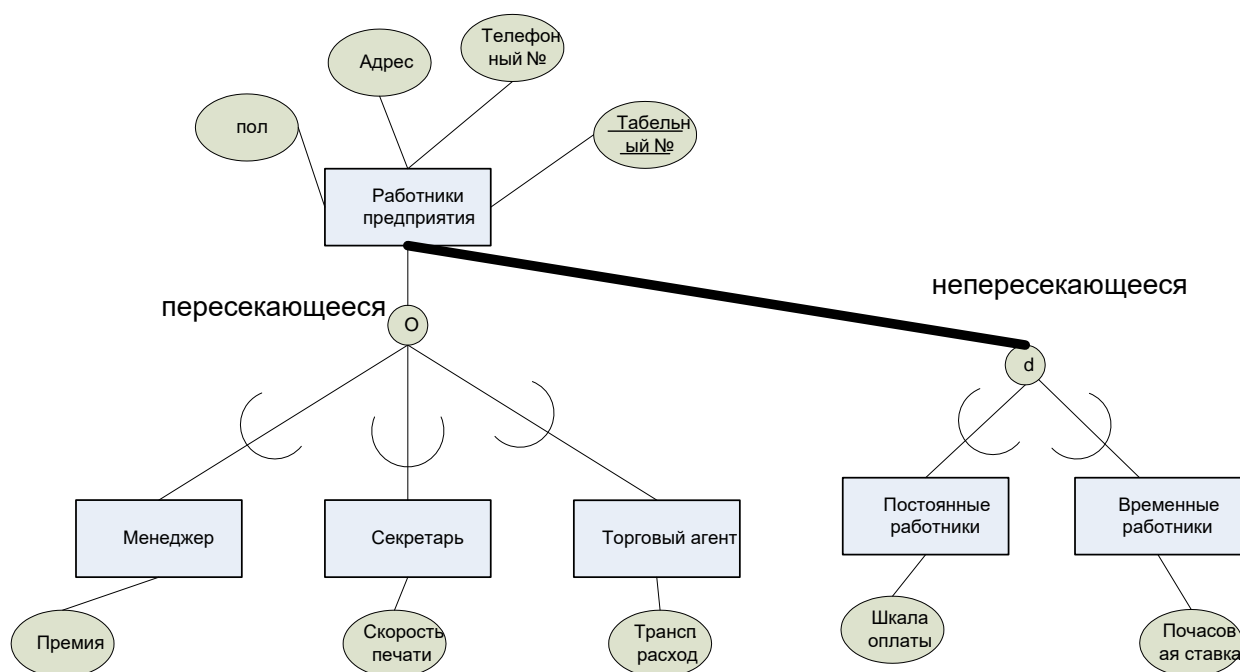


Рисунок 4.17 – Пример использование генерализации/ специализации.

Перекрывание атрибутов не допускается. Так как экземпляр принадлежит одновременно суперклассу и подклассу мощность этой связи равна один

к одному. Могут быть следующие варианты: один экземпляр суперкласса присутствует только в одном подклассе – специализации не пересекаются (в узле ставится буква d- disjoint), один экземпляр суперкласса присутствует только в разных подклассах – специализации пересекаются (в узле ставится буква o- non disjoint).

Для реализации множественного наследования в EER модели введено понятие категоризация - один подкласс (категория) связан с двумя суперклассами (рисунок 4.18).

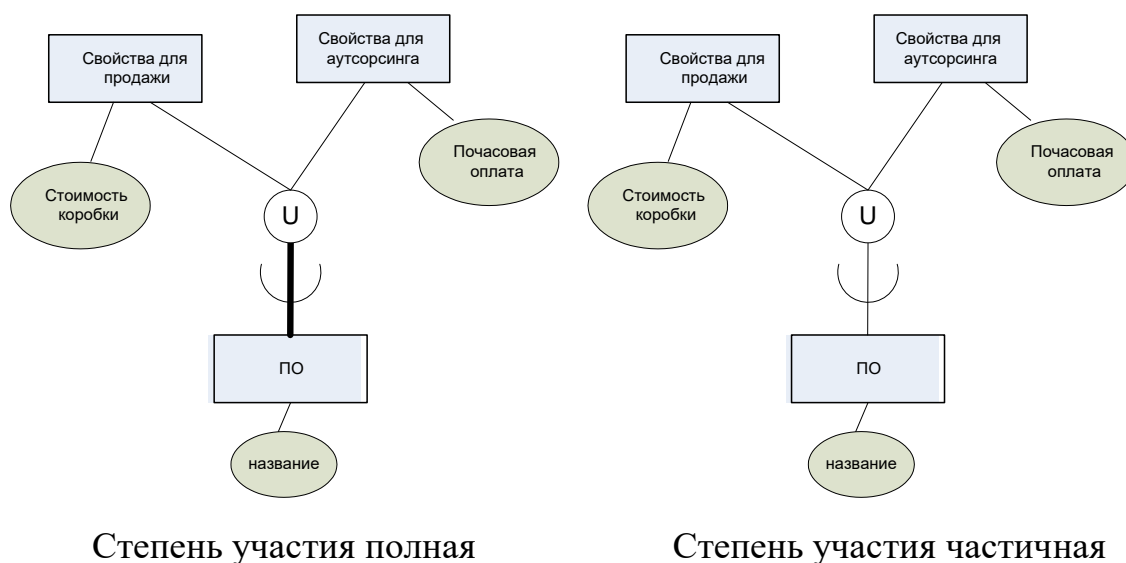


Рисунок 4.18 - Пример использование категории.

Методология концептуального проектирования

Предполагается, что к началу этого этапа предметная область изучена и построена функциональная модель в нотации DFD. Входной информацией на этом этапе являются все бизнес документы и DFD модель (диаграмма, словарь данных и спецификация процесса).

Нужно обратить внимание на то, что детализация хранилищ DFD не может быть получена исходя только из описания входных или выходных потоков. Этот факт легко понять, если обратиться к шагу определения связей между типами сущностей ERD. На этом этапе необходимо выделить глаголы и глагольные обороты, соединяющие типы сущностей, именно эти глаголы и определяют связи ERD. В словаре данных глаголов нет! Следовательно,

определить связи, используя только словарь данных невозможно. Если вспомнить отличие данных от информации (мы считаем, что информация это осмысленные данные) и учесть, что схема данных отражает семантику (смысл) информации, то становится понятно, что поток данных не может являться источником определения семантической связи этих данных.

Разработчик модели использует данные из различных источников. Природа этих данных в значительной степени будет зависеть от фазы моделирования. Каждый пользователь и разработчик имеет свое представление данных. Разработчик должен увидеть сущность основных значений и структуру данных. Для этого он сравнивает модели различного представления и вырабатывает обобщенное представление. Каждый документ можно рассматривать некоторым представлением структуры данных. Можно составлять схему данных для всех документов используемых в бизнес процессе и на основании обобщения этих представлений создавать глобальную модель данных предметной области. В главе 6 рассматривается именно такой алгоритм создания модели данных.

В данном пособии предлагается также рассматривать локальное представление данных с точки зрения хранилища, с которым они ассоциированы потоками данных на DFD модели. При этом принимается во внимание, что семантика информации отражена в спецификации бизнес процессов сопряженных с хранилищами.

Шаг1. Определение типов сущностей.

"Тип сущности" в контексте ERD отражает ряд «объектов внешнего мира», через данные, связанные с ними. «Объектом внешнего мира» может быть: человек, материальное тело, явление, документ, идея, понятие, момент времени, место и т.д. У множества элементов, представленного типом сущности, есть единый набор атрибутов или характеристик. Например, у всего множества служащих есть табельный номер, имя и другие общие атрибуты. Индивидуальный элемент множества сущностей обозначается сущностью или экземпляром типа сущности (экземпляр сущности). Один из путей иден-

тификации типа сущности является идентификацией всех существительных, встречаемых в описании предметной области. Можно определить к сущностям те существительные, которые используются со словами «число» или «код» (код изделия, число сотрудников). К типам сущностей нужно отнести существительные, которые обозначают предметы или вещи внешнего мира. Если же существительные обозначают информацию об этих предметах или вещах, то их нужно отнести впоследствии к атрибутам сущностей.

В виду того, что «тип сущности» довольно громоздкий термин его заменяют термином «сущность», в этом случае отдельный экземпляр типа сущности называется экземпляром (объектом или примером) сущности.

Чтобы отделять «сущность» от «не сущности», разработчик модели должен ответить на следующие вопросы по каждому претенденту на сущность (из числа существительных).

- Можно ли его описать? (Есть ли у него качество?)
- Можно ли привести его примеры?
- Можно ли примеры отделить друг от друга или идентифицировать?
- Оно описывает что-то или говорит о чем то? (Если в этом вопросе ответ «Да», то существительное обозначает атрибут, а не сущность).

Выделенному типу сущности должно быть дано имя (уникальное для всей ER модели). Тип сущности называется существительным в единственном числе. В методологии IDEF1X рекомендуется для каждого типа сущности определить: имя, описание (понимание его в данном бизнес процессе) и список синонимов.

В процессе разработки модели могут быть идентифицированы новые типы сущности или же некоторые типы сущности могут быть удалены.

Шаг 2. Определение типов связей.

Связь определяет отношение между сущностями. Можно применить подход, используемый при определении сущностей. В описание предметной области нужно выделить глаголы, которые соединяют одну или несколько сущностей. Например: «Студент **учиться** на специальности Прикладная информатика в экономике», «Студент **зачислен** приказом от 12.09.08 в группу 123».

Необходимо проверить:

- выделены все ли связи явно или не явно присутствующие в описание предметной области (проверка может выполняться как при изучении документов предметной области, так и при опросе специалистов);
- возможность замены n-арных связей бинарными (семантика n-арной связи не соответствует семантики множества бинарных связей).

После определение связи нужно определить кардинальное число, степень участия, ролевые имена сущностей в связи (по необходимости). Имя связи должно нести смысловое значение понятное пользователю.

Довольно простым способом в определении схемы данных является построение ее на основании документов предметной области. В тоже время нужно обращать внимание на то, что документ может не представить полную информацию о семантики предметной области, поэтому необходимо пользоваться и другой информацией.

Шаг 3. Определение атрибутов и связывание их с типами сущностей и связей.

Целью данной фазы привязка атрибутов к сущностям или связям. Пул атрибутов был определен в модели DFD входными и выходными данными хранилищ. Необходимо, что бы все эти атрибуты были распределены между идентифицированными типами сущностей и типами связей.

Для каждого атрибута должен быть определен его домен, который включает формат атрибута и допустимые значения.

Шаг 4. Определение атрибутов, являющихся потенциальными и первичными ключами.

При выборе первичного ключа нужно соблюдать следующие правила:

- использовать потенциальный ключ с минимальным набором атрибутов и минимальную длину;
- использовать потенциальный ключ, который обладает минимальной вероятностью потерей уникальности в будущем;
- использовать потенциальный ключ, который проще использовать пользователю.

На этом этапе определяется сильная и слабая сущность.

Шаг 5. Определение специализации/ генерализация типов сущностей (по необходимости).

Определение суперклассов, которые ассоциируются с атрибутами и связями общими для всех подклассов, и подклассов, с которыми ассоциируются атрибуты и связи, характеризующие особенности подкласса.

Каждый шаг документируется диаграммой ERD и сопроводительной документацией, где описываются особенности разрабатываемой модели на каждом шаге (см. выше).

Логическое проектирование

Фаза логического проектирования базы данных заключается в преобразовании концептуальной модели данных в логическую модель данных предприятия с учетом выбранного типа СУБД. В учебном пособии рассматривается реляционная СУБД.

Нотация IDEF1X

IDEF1X включает следующие компоненты в свою нотацию:

Сущности

- Сильные сущности (Identifier-Independent Entities)

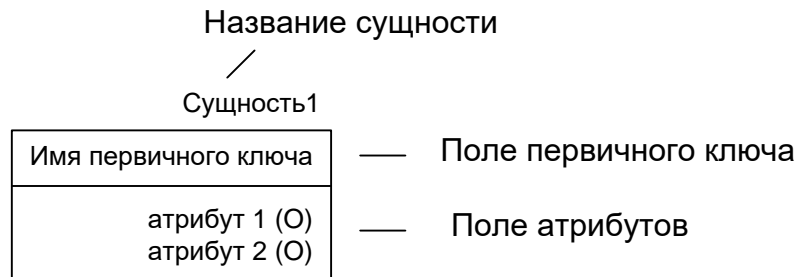


Рисунок 4.19- Сильная сущность

- Слабые сущности (Identifier-Dependent Entities). Слабая сущность возникает при определении идентифицирующей связи. Реализуется слабая сущность за счет миграции первичного ключа сильной сущности в составной атрибут первичного ключа слабой сущности (отображается внешним ключом). В этом случае слабая сущность без идентификации сильной сущности не может быть определена (атрибут первичного ключа не может быть Null).

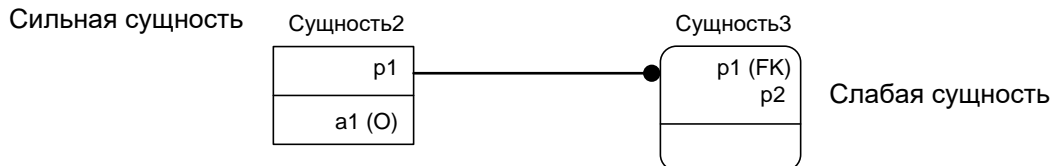


Рисунок 4.20 - Сильная и слабая сущности соединены идентифицирующей связью.

Связи

Изображается линией, соединяющей две сущности. Жирная точка показывает сторону связи «много». Сущность с примыкающей жирной точкой называется дочерняя, другая – родительская. В реляционной базе данных связь реализуется внешним ключом в дочерней таблице, ссылающимся на первичный ключ родительской. В силу этого CASE средства изображают связь, как «один-ко-многим» (линия с одной жирной точкой) при любой мощности связи.

- *Идентифицирующие* (Identifying Connection) связи. Изображается сплошной линией. Дает возможность определить слабую сущность (см. выше).
- *Неидентифицирующие* (Non-Identifying Connection) связи. Изображается пунктирной линией.

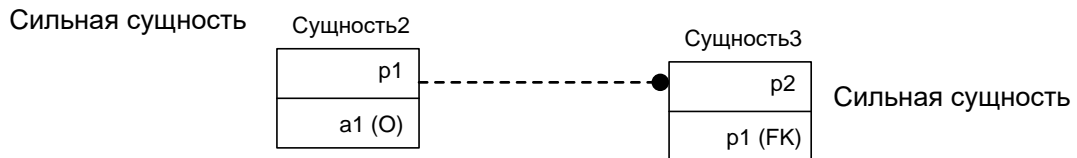


Рисунок 4.21 Неидентифицирующая связь.

- Полное и неполное участие. Если задано не полное участие, то возле родительской сущности появляется ромб, а в дочерней таблице возле атрибута внешнего ключа появляется «O» (Рисунок 4.22)

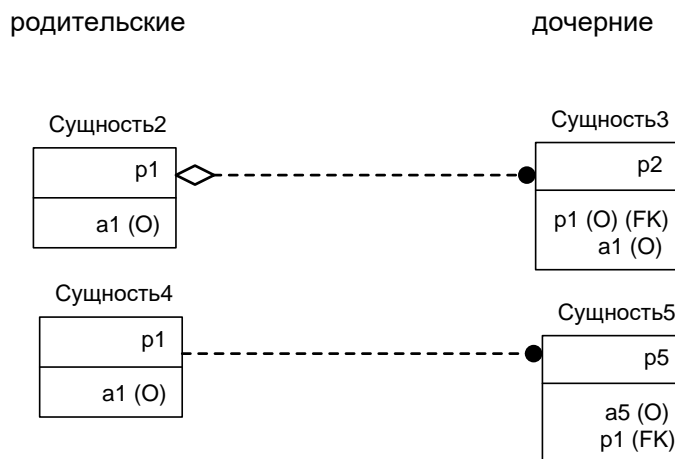


Рисунок 4.22 – Неполное (верхняя связь) и полное участие.

- Категоризация. Правило описание категории даны ниже.
- *Неспецифицированные* связи. Изображается линией с двумя жирными точками (по концам), она должна быть исключена в процессе моделирования.

Атрибуты/ Ключи

- *Атрибуты* (рисунок 4.19). Буква «O» показывает, что для этого атрибута допускается значение «Null».

- *Первичный ключ* (рисунок 4.19).
- *Альтернативный ключ*. Каждому альтернативному ключу в рамках сущности присваивается номер, и напротив каждого атрибута, который принадлежит данному ключу ставится метка (АК<номер ключа>)
- *Внешний ключ*. Указание в скобках FK (foreign key), говорит о том, что данный атрибут внешний ключ.

Отчет (описание). Может включать описание модели данных, выполненное как на концептуальном уровне (см. методологию концептуального проектирования), так и на логическом уровне (см. ниже).

Категоризация.

Использование отношений категоризации (рисунок 4.23, 4.24) подчиняется следующим правилам:

- каждая категория может иметь не более одной родовой сущности;
- категория может быть родовой сущностью для других категорий;
- никакая категория прямо или косвенно не может быть родовой сущностью для самой себя;
- родовая сущность может иметь несколько кластеров категорий. Для каждого кластера должен использоваться собственный дискриминатор;
- дискриминатор полного кластера категорий не может быть обязательным.

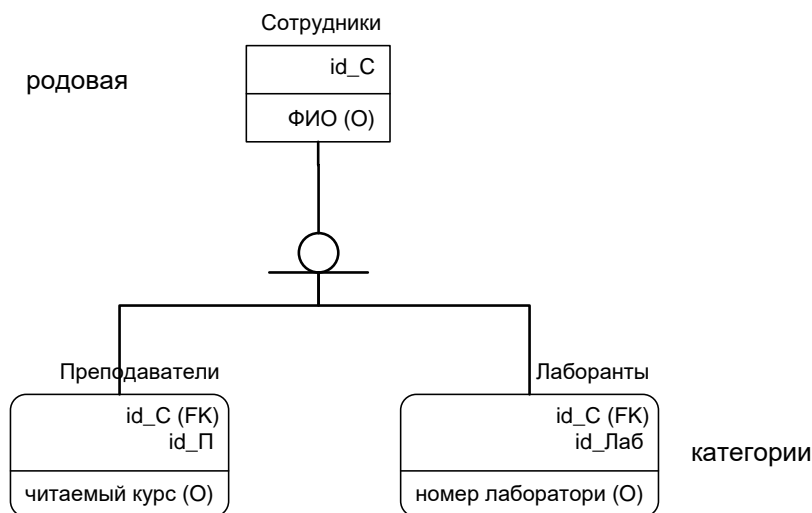


Рисунок 4.23 - Пример неполной категории. Необязательное присутствие родовой сущности в сущности категории - сотрудник может не относиться, ни к категории преподаватель, ни к категории лаборант.

Если некоторая сущность разбивается на категории, то ее называют родовой сущностью, а сущность, соответствующую категории, называют сущностью-категорией или просто категорией. Очевидно, что каждому экземпляру категории одновременно является экземпляром родовой сущности.

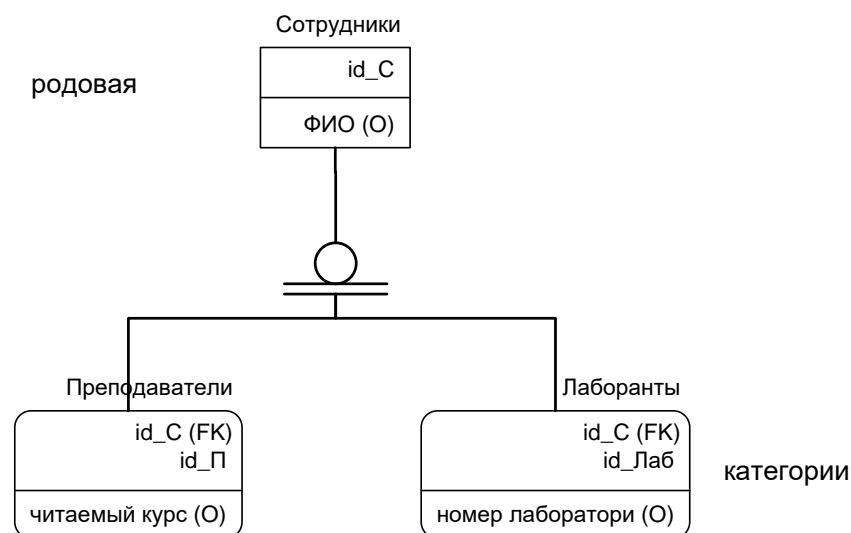


Рисунок 4.24 - Пример полной категории. Обязательное присутствие родовой сущности в сущности категории - сотрудник должен быть или преподавателем или лаборантом.

Кластер категории: набор одной или более взаимно исключающих отношений классификации для той же самой родовой сущности. Дискримина-

тор категории: признак в родовой сущности группы категории. Значения дискриминатора указывают, какой сущности категории принадлежит экземпляр родовой сущности. Категории должны включать экземпляры родовой сущности с одним значением дискриминатора (рисунок 4.25).

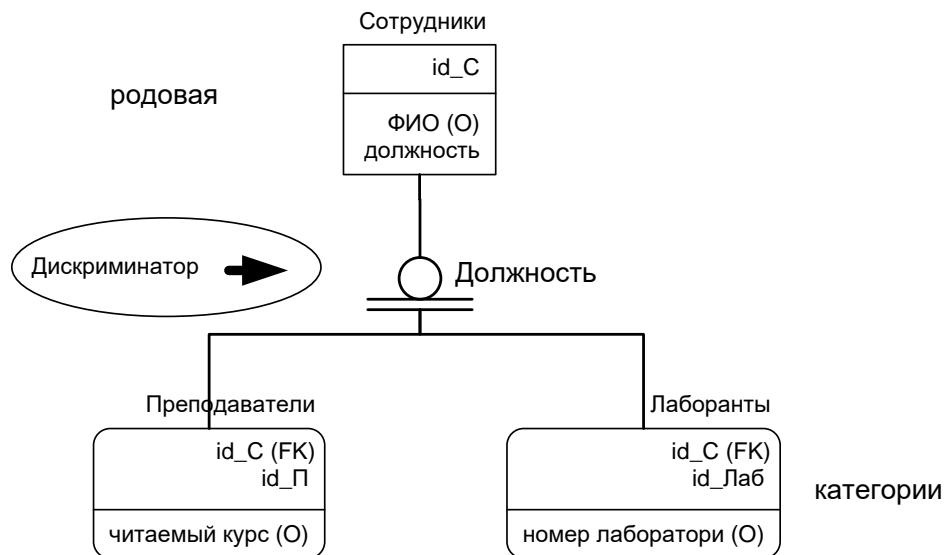


Рисунок 4.26 – Использование дискриминатора

Методология логического проектирования

Состоит из двух этапов: построение локальной логической модели и построение глобальной логической модели. В пособии рассматривается построение логической модели, основанной на использовании реляционной базы данных.

Этап 1. Построение локальной логической модели. Входом являются локальные концептуальные модели, выходом - локальные логические модели.

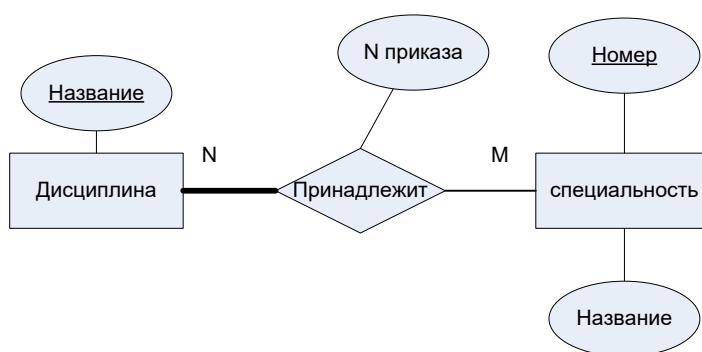
Шаг 1. Преобразование локальной концептуальной модели данных в локальную логическую модель.

Предлагается на этом шаге использовать CASE средства, использующие нотацию IDEF1X или подобную ей. Необходимо преобразовать те элементы концептуальной модели, которые не поддерживаются реляционной моделью, выполняя следующие действия /55/:

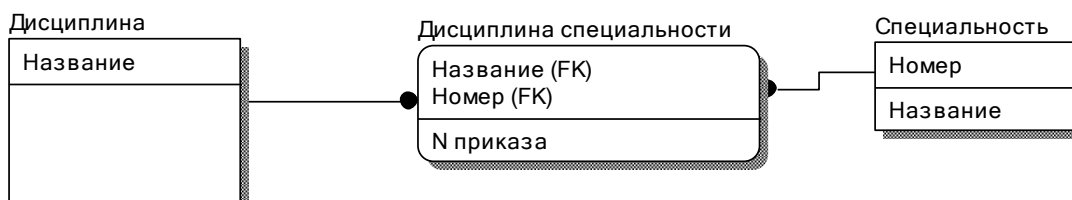
- удаление связей типа m:n;

- удаление сложных связей;
- удаление рекурсивных связей;
- удаление связей с атрибутами;
- удаление множественных атрибутов;
- перепроверка связей типа 1:1;
- удаление избыточных связей;
- Определение связи суперкласс/ подкласс.

Удаление связей *многое ко многим*, а также связи с атрибутом, за счет введения дополнительной слабой сущности.



а

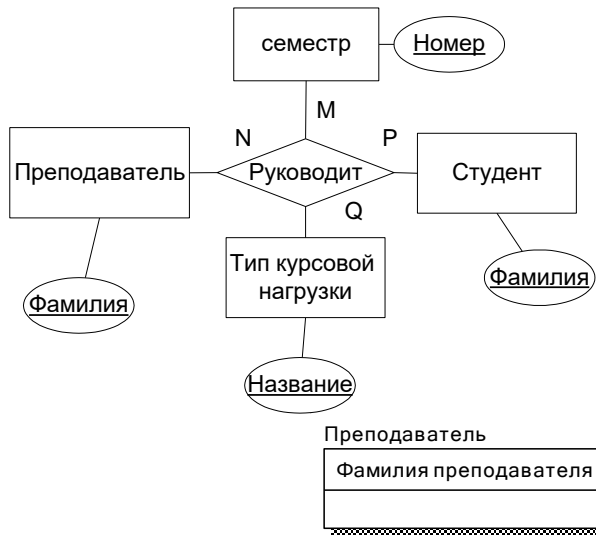


б

а- концептуальная модель; б- логическая модель

Рисунок 4.26 - Связь заменяется слабой сущностью.

Удаление сложных связей.



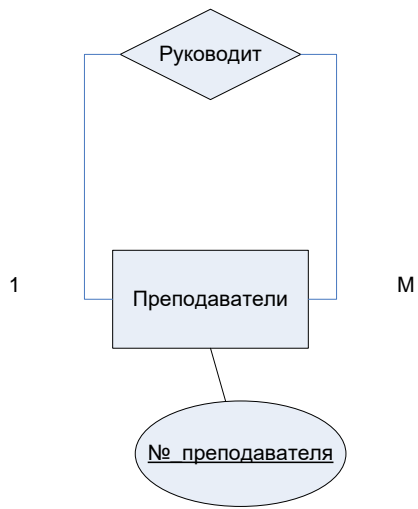
- Концептуальная модель



Логическая модель

Рисунок 4.27- Удаление сложной связи, за счет введение слабой сущности

Удаление рекурсивных связей.



Концептуальная модель



Логическая модель

Рисунок 4.28 – Удаление рекурсивной связи

Удаление множественных атрибутов.

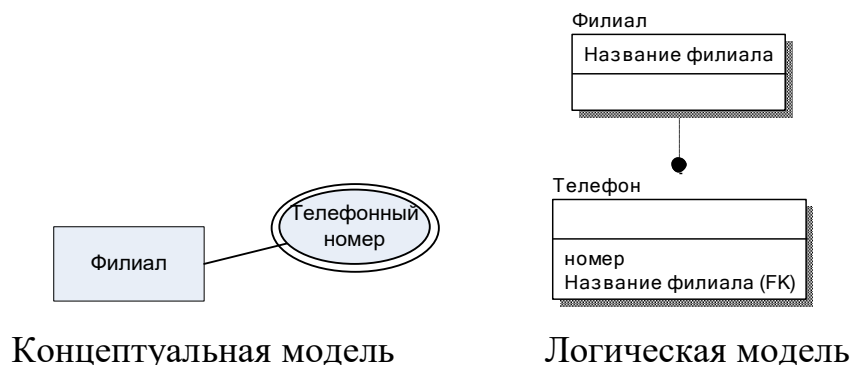


Рисунок 4.29 – Удаление множественных атрибутов

Перепроверка связей типа 1:1.

В процессе определения сущностей могли быть созданы две различные сущности, которые на самом деле представляют один и тот же объект в предметной области. Например, могли быть созданы две сущности, Преподаватель и Тьютер, которые на самом деле представляют один и тот же тип объекта. В этом случае целесообразно объединить атрибуты в одной сущности. Первичный ключ, объединенной сущности выбирается должен совпадать с первичным ключом одной из сущностей.

Слияние в единое отношение предпочтительнее в том случае, если данные сущности не принимают участия в других типах связей друг с другом. Не следует объединять сущности, если участие одной из них в связи частичное. Сущность, которая частично участвует в связи, определяется как родительская, а та сущность, которая участвует в связи полностью, определяется как дочерняя (в этой сущности внешний ключ ссылается на первичный ключ родительской сущности). В том случае если участие в связи обеих сущностей частичное, то выбор дочерней и родительской сущности произвольный.

В том случае если не целесообразно объединять сущности (отображают различные объекты реального мира), но в тоже время участие в связи обеих сущностей полное, выбор дочерней и родительской сущности произвольный.

Удаление избыточных связей.

Если две сущности соединены более чем одной связью напрямую или же опосредованно через другие сущности, то некоторые связи могут быть избыточными.

Связь является избыточной, если одна и та же информация может быть получена не только из нее, но и с помощью других связей.

В том случае, если экземпляры различных сущностей связаны напрямую или опосредованно через различные связи, то не будут являться избыточными связями те, в которых участие сущности частичное.

Определение связи суперкласс/подкласс.

Для реализации такой связи можно воспользоваться средствами CASE инструмента. Использование нотации IDEF1X предполагает реализацию понятия категория. Это понятие отличается от понятия категория в концептуальной модели. Категория в IDEF1X воплощает парадигму специализация/генерализация концептуальной модели с некоторыми ограничениями, описанными выше. На рисунке 4.30 и 4.31 показано преобразование концептуальной модели в нотации Чена в нотацию IDEF1X.

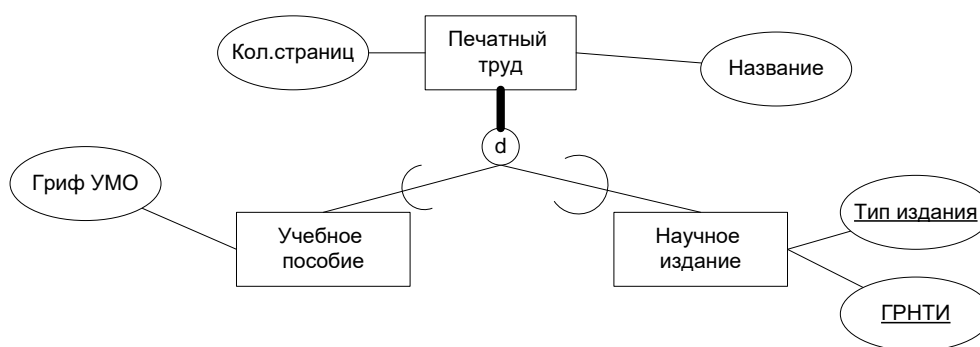


Рисунок 4.30 – Пример генерализации в концептуальной модели.

CASE средство фактически преобразовало категоризацию в вид описанный на языке DBDL /54/ (рисунке 4.32). В тоже время реализация генерализации возможна путем создания двух независимых сущностей «Учебное пособие» и «Научное издание» (рисунке 4.33), а также созданием общей сущности «Печатное издание» (рисунке 4.34).

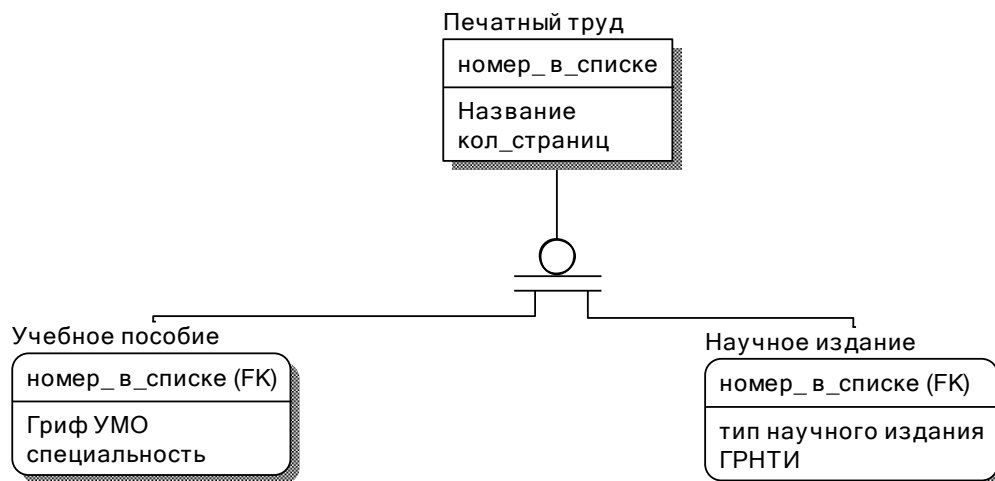


Рисунок 4.31 - Реализации модели концептуальной модели (рисунок 4.30) на логическом уровне.

Печатный труд (номер в списке, название, кол.стр.)

Primary Key номер в списке

Учебное пособие (Номер в списке, Гриф УМО)

Primary Key Номер в списке

Foreign Key Номер в списке **references** *Печатный труд* (Номер в списке)

Научное издание (Номер в списке, тип научного издания, ГРНТИ)

Primary Key Номер в списке

Foreign Key Номер в списке **references** *Печатный труд* (Номер в списке)

Рисунок 4.32 – Реализация генерализации (способ 1)

Учебное пособие (Номер в списке, Гриф УМО, название, кол.стр.)

Primary Key Номер в списке

Научное издание (Номер в списке, тип научного издания, ГРНТИ, название, кол.стр.)

Primary Key Номер в списке

Рисунок 4.33 – Реализация генерализации (способ 2)

Печатный труд (номер в списке, название, кол.стр., Гриф УМО, тип научного издания, ГРНТИ)

Primary Key номер в списке

Рисунок 4.34 – Реализация генерализации (способ 3)

Шаг 2. Проверка модели с помощью правил нормализации:

- приведение к первой нормальной форме (1НФ), позволяющее удалить из отношений повторяющиеся группы атрибутов;
- приведение ко второй нормальной форме (2НФ), позволяющее устранить частичную зависимость атрибутов от первичного ключа;
- приведение к третьей нормальной форме (3НФ), позволяющее устранить транзитивную зависимость атрибутов от первичного ключа;
- приведение к нормальной форме Бойса-Кодда (НФБК), позволяющее удалить из функциональных зависимостей оставшиеся аномалии.

Шаг 3. Проверка модели в отношении транзакций пользователей.

Модель DFD дает представление о технологии работы пользователя с информационной системой. При этом определяются действия пользователя, а также потоки данных записи или чтения данных из хранилища. Все это дает возможность на этом этапе работы создать прототипы пользовательского интерфейса и создать список транзакций²⁵ базы данных. Нужно обратить внимание, что транзакции определяются не только обращением пользователя, но и обращением системы к базе данных.

²⁵ Транзакция - группа последовательных операций, которая представляет собой логическую единицу работы с данными. Транзакция может быть выполнена целиком либо успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта /40/.

На этом этапе нужно сформулировать все транзакции и проверить их по локальным логическим моделям данных.

В гл 6 рассматривается транзакция: составление списка преподавателей, которые в данный год будут работать на кафедре. В соответствии с DFD моделью прототип пользовательского интерфейса, соответствующий этой транзакции, представлен на рисунке 4.35. На рисунке 4.36 представлена диаграмма на логической схеме данных.

Кафедра: Информационных систем Год учебного плана: 2007

ФИО	Должность	Степень	Штатный	Вну. совместитель	Вне. совместитель	ДОУ	Ставка
Иванова Вера Петровна	Профессор	д. пед. н.	1	0,2	0	0	1,2
Сидорова Ольга Сергеевна	Доцент	к. э. н.	0	0	0,5	0	0,5
Петрова Ирина Ивановна	Ассистент		0	0	0	1	1

ФИО преподавателя: Сидорова Ольга Сергеевна
 Должность: Доцент
 Степень: к. э. н.
 Штатный: 0
 Внут. совместитель: 0
 Вне. совместитель: 0,5
 ДОУ: 0

Рисунок 4.35 – Прототип пользовательского интерфейса.

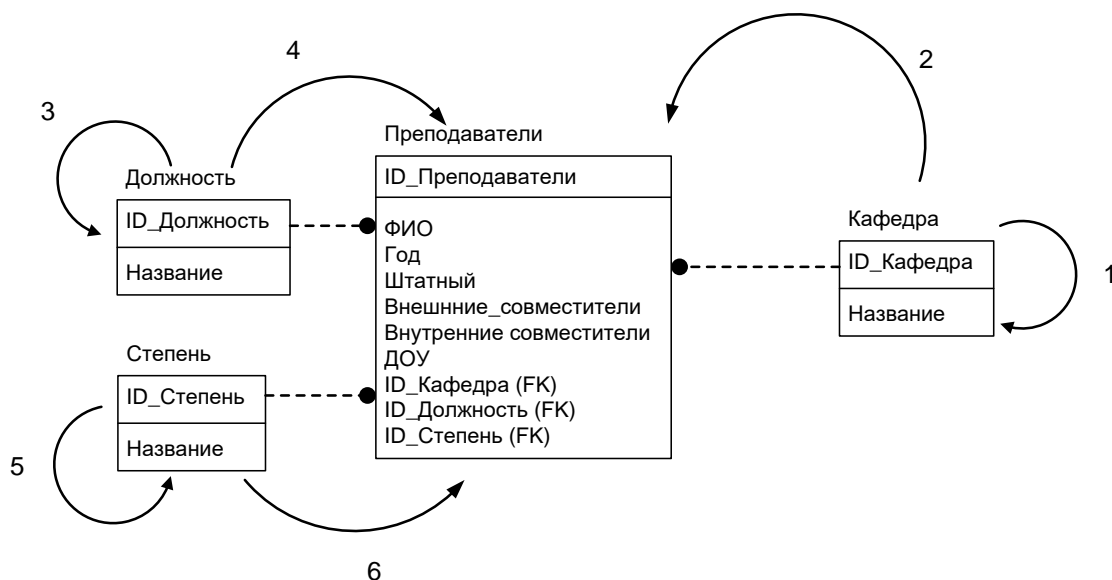


Рисунок 4.36 – Представление транзакции на ER диаграмме.

Шаг 4.. Определение требований поддержки целостности данных.

Необходимо выявить правила, обеспечивающие в представлении пользователей правильность данных в любой момент времени. Определяется пять основных типов требований ограничения целостности данных /55, 56/:

- обязательные данные (выявление неключевых атрибутов, для которых запрещается Null) - ограничения должны фиксироваться при занесении сведений об атрибуте в словарь данных;
- ограничения для доменов атрибутов (определение допустимых значений атрибутов) – ограничения устанавливаются при определении доменов атрибута;
- целостность сущностей (первичный ключ любой сущности не могут содержать null) – учитывается при определении первичного ключа;
- ссылочная целостность (если внешний ключ содержит некоторое значение, то оно обязательно должно присутствовать в одной из строк поля первичного ключа родительского отношения);
- бизнес правила (ограничения, действующие на данном предприятии, например, преподаватель «ассистент» не может читать лекции).

Шаг 5. Обсуждение разработанных локальных логических моделей данных с конечными пользователями.

Этап 2. Построение глобальной логической модели данных /55, 56/.

Шаг 1. Слияние локальных логических моделей данных в единую глобальную модель данных.

Выявить сущности в различных локальных моделях, отображающие одни и те же объекты и явления внешнего мира. Правильное документирование выявленных сущностей должна обеспечить:

- идентификацию сущностей с различными именами, но отображающих одно и то же множество объектов внешнего мира (для каждого имени сущности указываются синонимы)
- идентификацию сущностей названных одним именем, но обозначающих различные множества объектов внешнего мира.

Произвести слияние сущностей, отображающие одни и те же множества объектов внешнего мира. Остальные сущности включаются в глобальную модель без слияния.

Слияние общих связей. Под «общими связями» понимаются связи между одинаковыми сущностями (для которых было выявлено слияние) и имеющие одно и то же назначение. Связь между двумя сущностями может быть опосредована другой сущностью, которая, например, была введена на первом этапе, для того, что бы разрешить ограничения реляционной базы данных. В силу этого, необходимо для каждой пары «слитых» сущностей проанализировать все типы связей (в том числе и через третьи сущности) на выявления общих связей.

Остальные связи включаются без изменения.

Шаг 2. Проверка глобальной логической модели данных.

Проверка:

- на наличие пропущенных сущностей и связей;
- проверка ограничений целостности.

Шаг 3. Проверка возможностей расширения модели в будущем.

Шаг 4. Обсуждение глобальной логической модели данных с пользователями.

Физическое проектирование базы данных /55/

Процесс создания описания конкретной реализации базы данных, размещаемой во вторичной памяти. Предусматривает описание структуры хранения данных и методов доступа, предназначенных для осуществления наиболее эффективного доступа к информации.

Фаза физического проектирования базы данных предусматривает принятие разработчиком окончательного решения о способах реализации создаваемой базы. Поэтому физическое проектирование обязательно производится с учетом всех особенностей используемой СУБД.

Этап 1. Перенос глобальной логической модели данных в среду целевой СУБД.

Шаг 1. Проектирование основных таблиц в среде целевой СУБД. Данный этап можно выполнить в среде CASE средства, которое позволяет выполнить физическое проектирование и осуществить импорт разработанной структуры в СУБД.

Шаг 2. Реализация бизнес-правил предприятия в среде целевой СУБД. Реализация бизнес правил реализуется триггерами и хранимыми процедурами.

Этап 2. Проектирование физического представления базы данных.

Шаг 1. Анализ транзакций.

Шаг 2. Выбор файловой структуры.

Шаг 3. Определение вторичных индексов.

Шаг 4. Анализ необходимости введения контролируемой избыточности данных.

Шаг 5. Определение требований к дисковой памяти.

Этап 3. Разработка механизмов защиты.

Шаг 1. Разработка пользовательских представлений (видов).

Шаг 2. Определение прав доступа.

Этап 4. Организация мониторинга и настройка функционирования системы

5. Применение методологии ООП при разработке ИС

Можно выделить следующую последовательность этапов (в терминологии RUP - «дисциплин») разработки ПО и возможные модели, используемые в них /57-62/:

- Моделирование бизнес процессов
 - Диаграммы бизнес - прецедентов
 - Диаграммы деятельности (диаграммы рабочих потоков)
 - Диаграмма бизнес - сущностей.
- Определение требований к системе
 - Диаграммы прецедентов
 - Диаграммы деятельности (описание потоков прецедентов)
 - Диаграмма классов
 - Диаграмма последовательности (взаимодействие объектов системы)
 - Диаграмма состояния (для классов которые меняют свое состояние в течение жизненного цикла)
- Анализ и проектирование
 - Диаграмма классов уровня анализа
 - Диаграмма последовательности
 - Дополнительная спецификация
 - Диаграмма прецедентов для событий (по необходимости)
 - Диаграмма классов уровня проектирования
 - Диаграмма коммуникации
 - Диаграмма состояний (по необходимости)
 - Диаграмма компонентов
 - Диаграмма развертывания
- Проектирование базы данных (входит в «Анализ и проектирование»)
 - Диаграмма классов

- Схема данных

- Реализация
- Тестирование
- Внедрение

В данном приложении даются краткие рекомендации по составлению объектных моделей на каждом этапе (дисциплине) выполнения проекта. Здесь не рассматриваются вопросы, связанные с жизненным циклом проекта²⁶.

Бизнес моделирование

Моделирование бизнес процессов необходимо /57/:

- исполнитель плохо знаком с бизнес процессом;
- в организации ранее проводилось обратное проектирование бизнес процессов;
- организация требует проведение обратного проектирования бизнес процессов;
- разрабатывается ПО, которое будет использоваться во многих подразделениях предприятия;
- в организации присутствуют плохо документированные процессы.

Моделирование бизнес процессов не нужно, когда:

- есть ясная картина структуры, целей и задач бизнеса;
- разрабатывается ПО, которое не будет широко использоваться в организации и не повлияет на остальной бизнес;
- рабочие потоки в организации понятны и хорошо документированы;

²⁶ Напомним, что перечисленные дисциплины должны выполняться последовательно в каждой итерации. В течение каждой итерации формируется планируемые инкременты системы, которые на последующих итерациях могут увеличивать свою функциональность.

- на моделирование бизнес процесса нет времени (требуется обоснование того, что отсутствие бизнес-моделирование не повлияет на качество разработки).

Цели бизнес-моделирования /63/:

- выявление проблем в организации и определение потенциальных улучшений;
- оценка воздействия организационного изменения;
- единое понимание организации у клиентов, разработчиков, пользователей и других сторон;
- получение требований к программному обеспечению, необходимых для поддержания организации;
- определить, как будущая система вписывается в организацию.

Можно выделить два этапа, которые требуется осуществить при выполнении бизнес – моделирования: понимание и описание существующего бизнес процесса и исследование автоматизации процессов.

На первом этапе требуется построить диаграмму бизнес прецедентов и, при необходимости, детализировать сценарий прецедентов диаграммами деятельности.

На втором этапе требуется выполнить следующие задачи:

- определение заинтересованных сторон;
- согласование целей;
- определение возможных ограничений;
- формулировка проблемы;
- расстановка приоритетов;
- разработка документа «Бизнес-видение»;
- оценка результатов.

При этом выявляются возможные технологические пути решения обнаруженных проблем. Для этого определяется уровень автоматизации, возможные технологии, выявляются системные акторы и прецеденты.

На рисунке 5.1 приведен пример диаграммы бизнес – прецедентов. На рисунке 5.2 приводится пример диаграммы бизнес объектов.

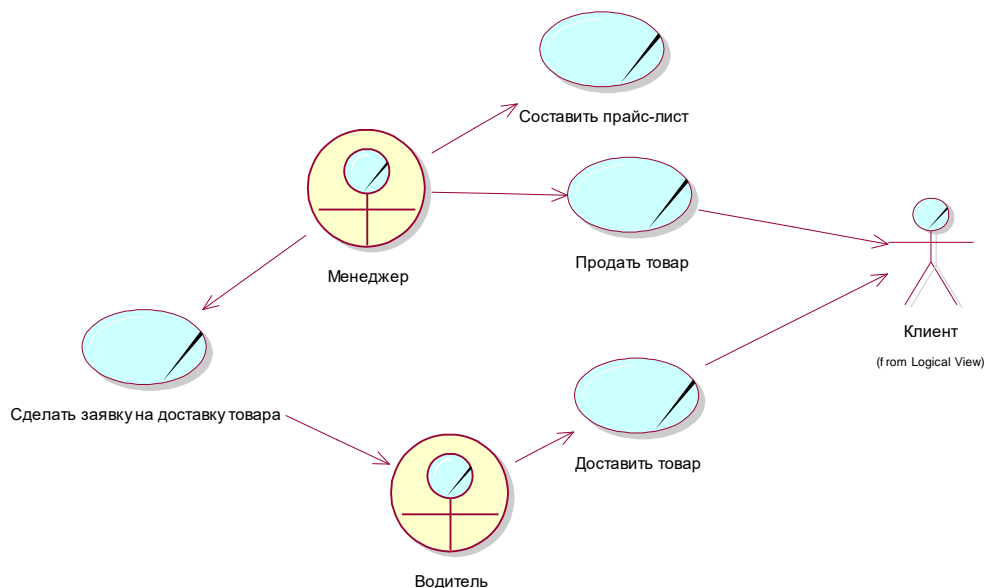


Рисунок 5.1 Диаграмма бизнес прецедентов «Продажа мебели в рознице».

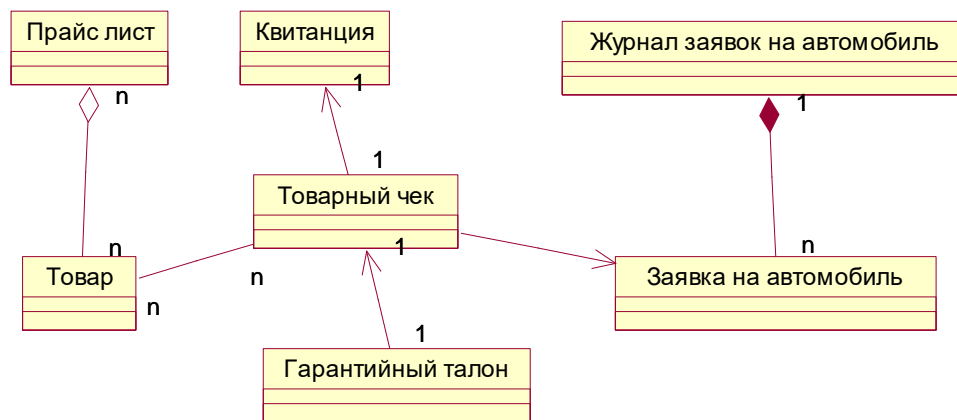


Рисунок 5.2 - Диаграмма бизнес объектов (в данной диаграмме учитывается, что товар не имеет серийных номеров, т.к. осуществляется продажа мебели).

На рисунке 5.3 приведена диаграмма деятельности. Данный пример демонстрирует, на каких стадиях формируются документы.

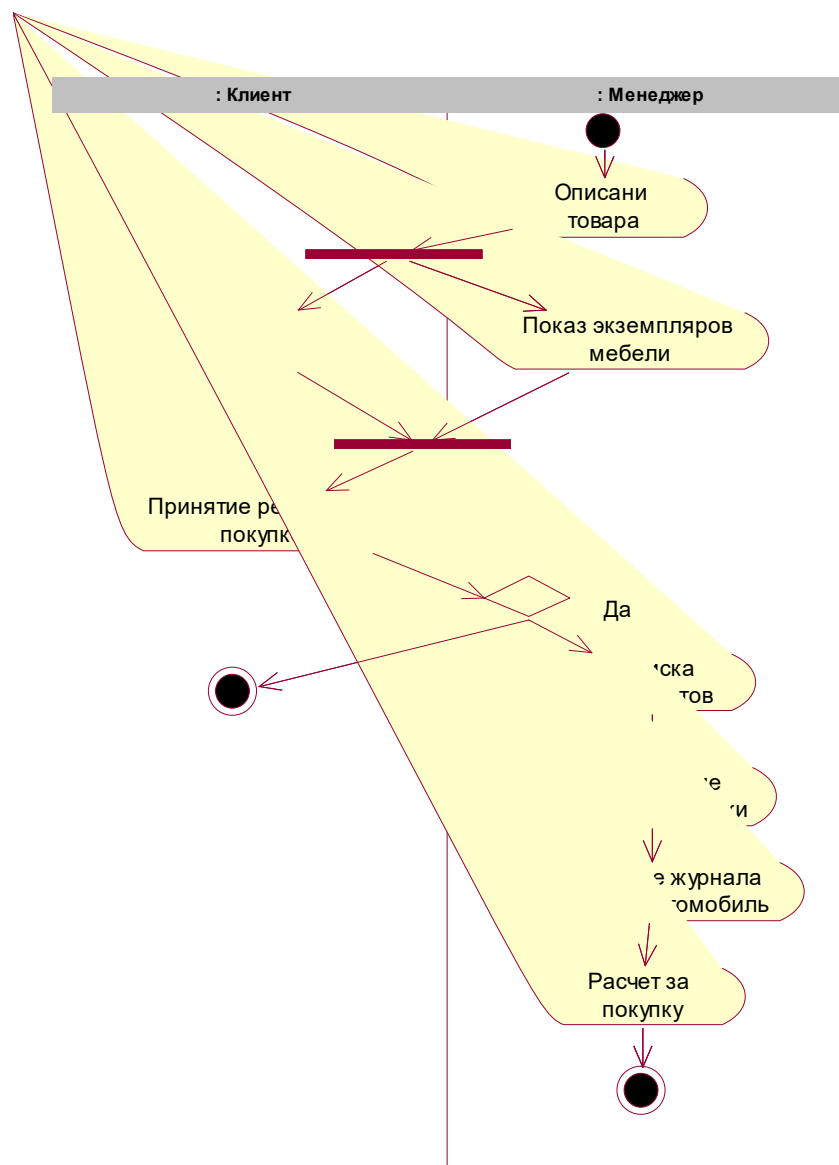


Рисунок 5.3 – Диаграмма действий основного потока прецедента «Продажа товара».

Мы не приводим спецификации процессов. Моделирование процесса показало, что в этом процессе невозможно применить автоматизированные средства, т.к. в процессе не используется обработка информации.

Определение требований к системе

На этом этапе необходимо: выявить проблемы, которые необходимо решить с помощью системы; определить заинтересованные стороны; выявить, собрать и проанализировать требования. Входом этапа являются: бизнес-правила, модель бизнес прецедентов. Выходом - функционал системы,

представленный с точки зрения заинтересованных сторон (описание функций системы в терминах, используемых в предметной области).

К основным задачам, решаемых на этом этапе, следует отнести:

- выявить требования заинтересованных сторон;
- определение границ системы;
- описание основных функциональных возможностей системы;
- построение **модели прецедентов** (для системы);
- определение пользовательского интерфейса системы, исходя из потребностей и целей пользователей;
- детализация требований с помощью **диаграмм последовательности** и **диаграмм состояния** (используются реже);
- создание терминов системы (составления глоссария).
- создание плана итераций (план итераций может быть выражен в неформализованном виде);
- определение нефункциональных требований - требований, не специфицированных моделью прецедентов (подготовка документа «Дополнительная спецификация»);
- разработка документа «Видение» (может использоваться любой документ с подобным функционалом – Приложение Д);

На рисунке 5.4 приведен пример пакета прецедентов, связанных с актором «Преподаватель». На этом же рисунке показана граница системы. В таблице 5.1 дана спецификация одного из прецедентов, а на рисунке 5.2 детализируется описание этого же прецедента с помощью диаграммы последовательности. Как видно из представленного материала описание требований к системе ведется в терминах предметной области.



Рисунок 5.4 – Пакет, в котором сгруппированы прецеденты связанные с актором «Преподаватель».

Таблица 5.1 – Пример спецификации прецедента.

Спецификация прецедента «Назначить задание студентам»	
Краткое описание	Прецедент описывает последовательность событий при назначении пользователем задания для выполнения.
Предусловия	Наличие необходимых записей в справочниках “Группы”, “Дисциплины”, а также наличие в системе заданий, необходимых для назначения.
Основной поток событий	<ol style="list-style-type: none"> 1. Пользователь с правами доступа “Преподаватель” на форме “Назначение заданий” выбирает дисциплину, группу, задание и подтверждает назначение данного задания заданной группе (нажимает “Назначить”). 2. Система проверяет корректность назначения задания. 3. Система присваивает каждому назначенному заданию статус выполнения “Нет полученных писем”.
Альтернативный поток	<p>A2: При повторном назначении задания система выдает пользователю соответствующее сообщение.</p> <p>A3: В случае отсутствия связи с сервером БД пользователю выдается сообщение об ошибке.</p>
Постусловия	Отметка в журнале успеваемости о назначении задания, запись об отправленном письме в журнале сообщений.
Точки расширения	–
Точки включения	–

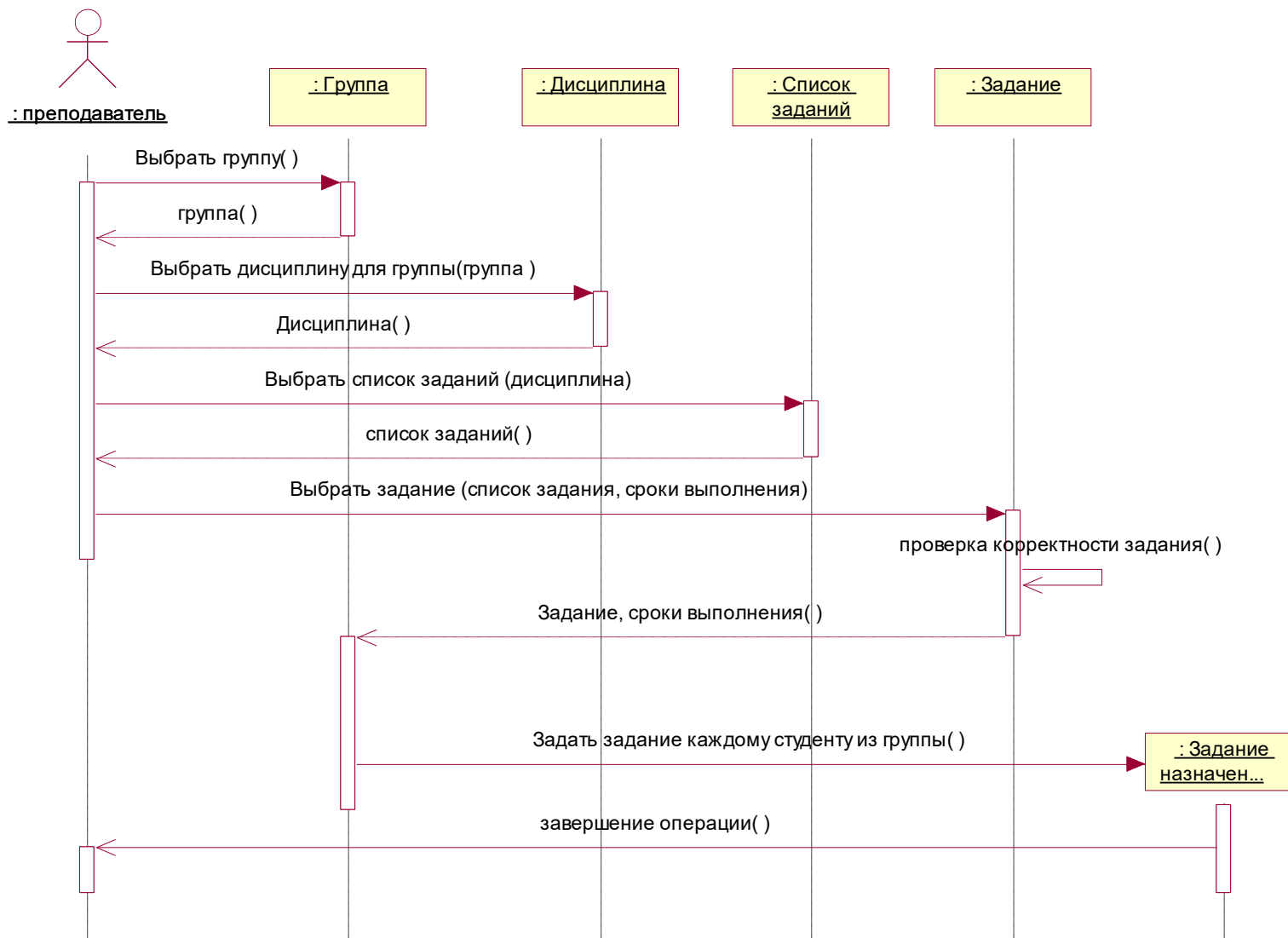


Рисунок 5.5 – Диаграмма последовательности для прецедента «Назначить задание студенту».

Очевидно, что для того, чтобы построить диаграмму последовательности требуется определить классы системы. В данном случае использовалась диаграмма сущностей определенная на этапе бизнес моделирования, но при этом она была дополнена классом задание назначенное студенту.

В силу того, что в последствие на стадии анализа была изменена диаграмма классов, модель классов на этом этапе не приводится.

Анализ и проектирование

В RUP определяется четыре рабочих процесса, которые выполняются в рамках этой дисциплины: создание архитектуры, анализ поведения, проектирование компонентов и проектирование базы данных (выполняется параллельно с проектированием компонентов). В тоже время некоторые авторы разделяют этап анализ и проектирования. Так, например, Дж. Рамбо и М.Блах к этапу анализа приложения относят действия, описанные в определении требованиями. Цель анализа, как считают эти авторы, состоит в определении проблемы, не давая преимуществ, какому либо конкретному варианту реализации /61/. В дальнейшем будет использоваться подход RUP, используемый для небольшой команды /63/.

В фазе «Начало» определяются одно из возможных архитектурных решений, которое может использоваться в данном проекте. В фазе «Уточнение» в том случае если архитектура системы не задана внешними факторами, требуется выбрать варианта архитектуры. При этом выполняются следующие шаги (выходные модели обозначены в скобках):

- описание контекста системы (высокоуровневой модели взаимодействия);
- архитектурный анализ (модель анализа²⁷, модель проекта²⁸, диаграмма развертывания);

²⁷ Модель анализа включает классы анализа и любые связанные с ними артефакты. Модель анализа в течение проекта может меняться. Она кратко фиксирует концептуальное описание системы.

- анализ прецедентов (модель реализации прецедентов, модель анализа);
- анализ операций (операции и их реализация).

На этом этапе, основываясь на анализе модели прецедентов и диаграмме бизнес-сущностей, требуется:

- определить классы анализа из поведения прецедента;
- распределить поведение в классах анализа;
- описать права;
- описать атрибуты и ассоциации;
- согласовать реализации прецедентов.

Модель анализа специфицируется **диаграммой классов** (пример представлен на рисунке 5.6), которые могут объединяться в пакеты (на высоком уровне абстракции можно представить **диаграмму пакетов**). Для каждого архитектурно-значимого прецедента и каждой системной операции описать ее выполнение в виде последовательности шагов, описанных в терминах действий и взаимодействий подсистем. Описать взаимодействия подсистем с помощью **диаграмм последовательности** (рисунок 5.7).

Классы анализа, представляющие результат анализа прецедентов, описывают концептуальных исполнителей поведения. На уровне проектирования (Design), классы анализа развиваются в различные виды элементы дизайна, описанные ниже:

- классы, представляют набор детально описанных функций;
- подсистемы – набор описанных в широком смысле функций, возможно также составленных из набора подсистем, но в конечном счете наборе классов;

²⁸ Модель проекта - абстракция реализации системы. Она используется, чтобы зафиксировать проект программной системы и включает: все классы проекта, подсистемы, пакеты, взаимодействие и связи между ними.

- активные классы, представляющие потоки в системе;
- интерфейсы, представляющие абстрактно объявленные функции, предоставляемые классом или подсистемой.

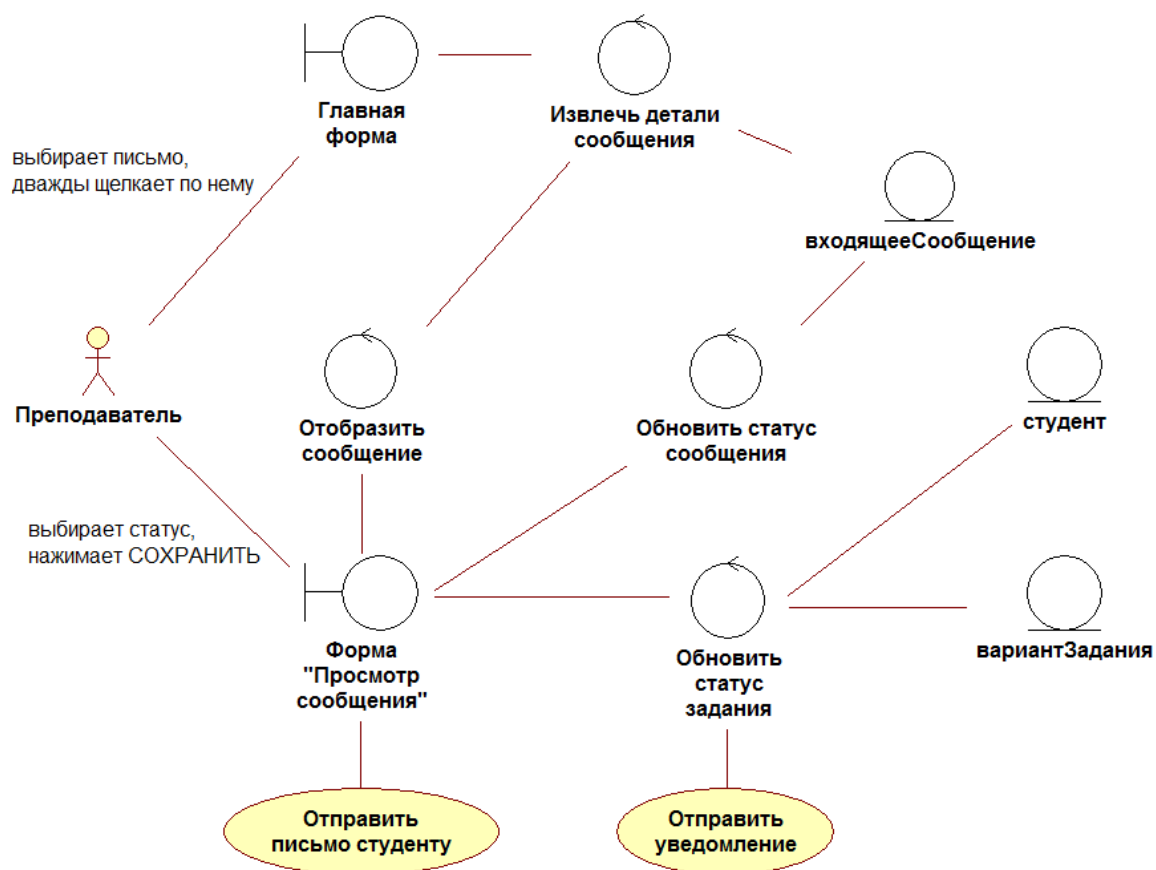


Рисунок 5.6 – Диаграмма классов (модель пригодности) для прецедента проверить задание.

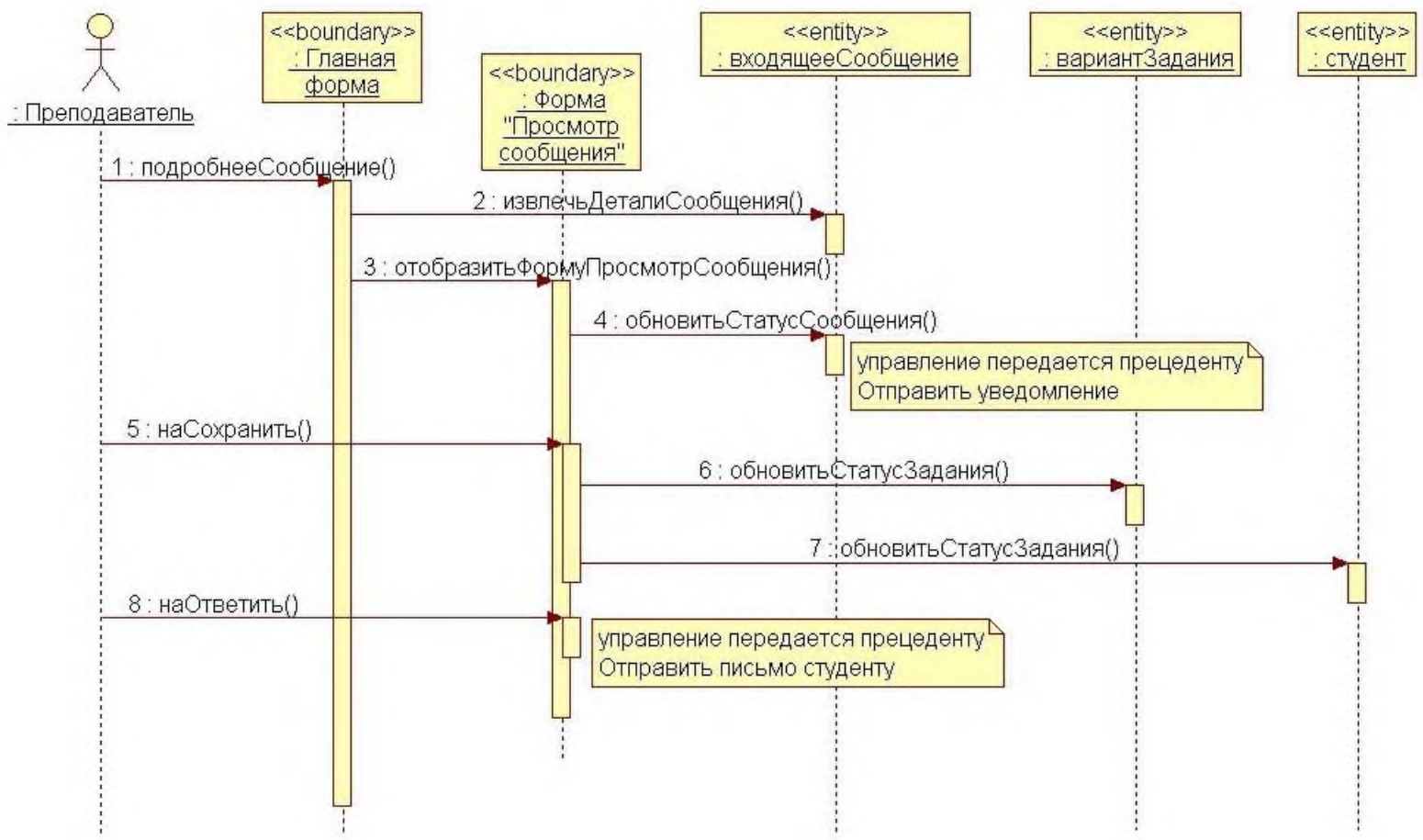


Рисунок 5.7 – Диаграмма последовательности для прецедента «Проверить задание»

Кроме этого на этом этапе определяются события²⁹. Выявление таких деталей дает возможность исследовать различные аспекты уровня проектирования

События и сигналы дают возможность описать асинхронные триггеры поведения, на которые должна отвечать система.

Классы и подсистемы позволяют группировать функции в относительно независимые модули. Класс – реализация атомарной функции, подсистемы – блоки, составленные из классов или других подсистем. Подсистемы используются для представления рабочих продуктов группы разработчиков как единый функциональный модуль, а также как логический элемент дизайна. Активные классы используются для представления управляющих потоков в системе, позволяя моделировать параллельное выполнение. Активные классы обычно используются совместно с другими классами, но не обязательно, например, для моделирования общего поведения.

Интерфейсы позволяют исследовать и фиксировать «места соединения» системы, определяя взаимодействие составных частей системы.

На рисунке 5.8 представлен пример диаграммы классов уровня проектирования.

Проектирование пользовательского интерфейса выполняется с использованием тех строительных блоков системы, которые были определены на этот момент. Для отображение навигационной карты интерфейсов можно использовать диаграмму состояний /39/. В этом случае стереотипы состояния будут обозначать типы окон (окно просмотра строк, Web-страница, окно сообщений и т.д.) и другие GUI-объекты. Состояния могут включать действия, которые отображают элементы управления окном (ко-

²⁹ События – привлекающие внимание явления, возникающие во времени и пространстве, обычно (но не всегда) требующие некоторого ответа системы /63/

мандная кнопка, выпадающее меню, пункт всплывающего меню, полоса прокрутки и т.д.).

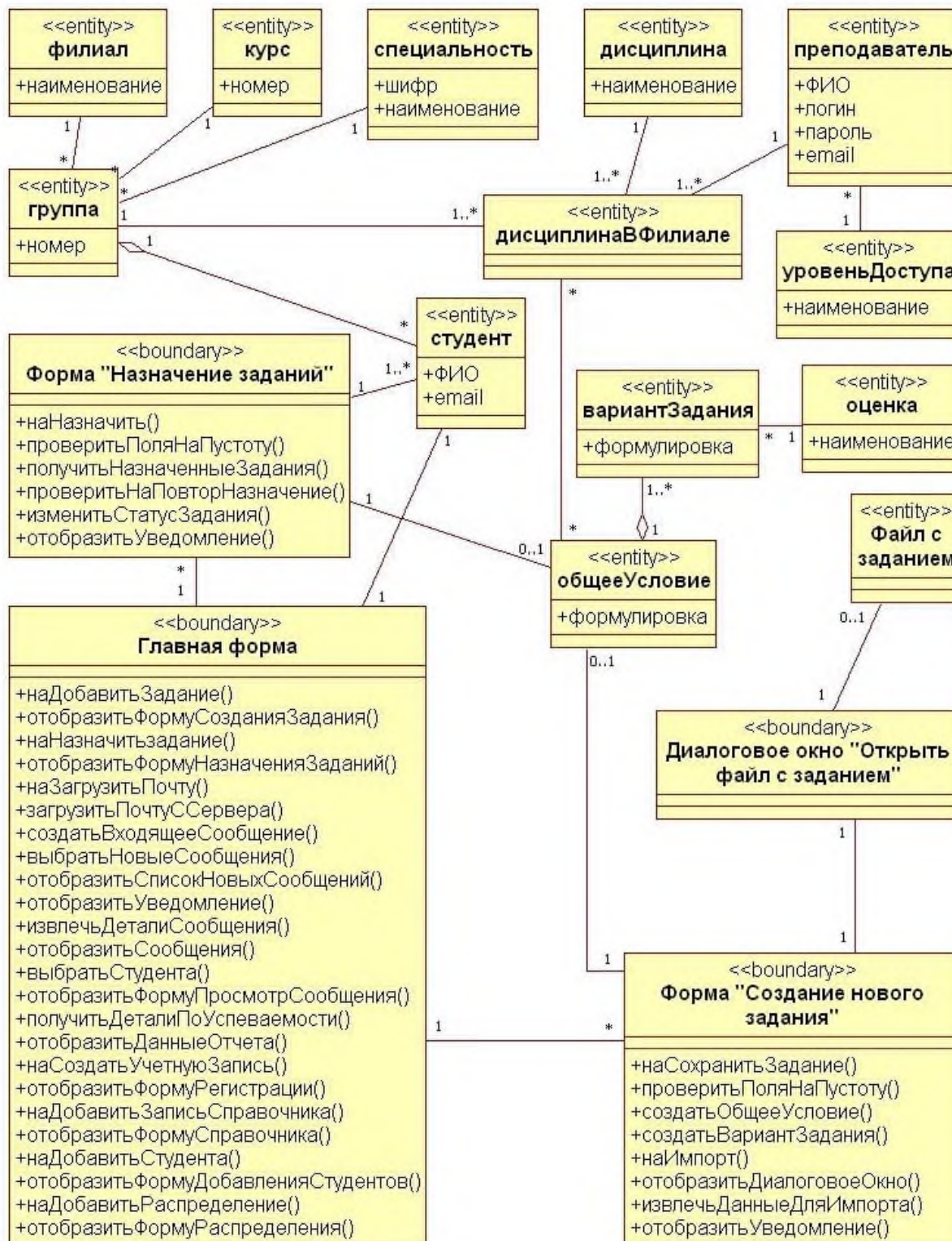


Рисунок 5.8 - Диаграмма классов уровня проектирования для пакета «Операции с заданиями».

Проектирование компонентов системы может строиться на различных принципах, зависящих от выбранной реализации системы. Организация системы в управляемую структуру начинается на этапе формирования пакетов прецедентов, которое может быть развита в пакеты классов. Пакеты классов могут строиться на основе объединения их в кооперации, реализующие соответствующие прецеденты. В тоже время можно объединять в соответствии со стереотипам трехзвенной архитектуры (пакеты «boundary», пакеты «control», пакеты «entity»). В силу того, что пакеты объединяют классы, которые «сильно» взаимодействуют друг с другом, и «слабо» взаимодействуют с классами другого пакета, полученную пакетную организацию можно продолжить при реализации компонентов.

Компоненты являются физическими частями системы, которые будут размещаться в соответствующих узлах системы. Узлы системы определяются техническим решением развертывания системы. Так, например, для реализации Web-приложения требуется: клиентский Web-браузер, Web-сервер, сервер приложений и сервер баз данных. Компоненты будут организовываться в зависимости от выбранной архитектуры развертывания. На этом этапе разрабатывается **диаграмма развертывания** (рисунок 5.9) и **диаграмма компонентов**.

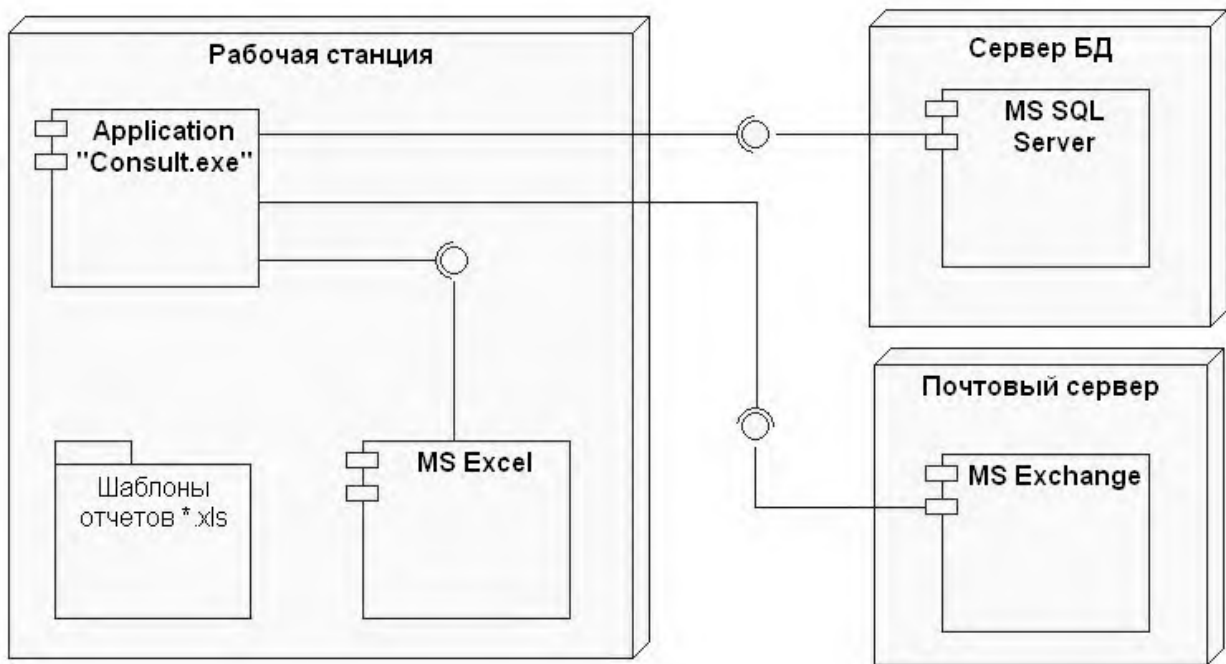


Рисунок 5.9 – Пример диаграммы развертывания.

Проектирование базы данных

Эта деятельность включает:

- идентификация постоянных классов в проекте;
- проектирование структуры базы данных, которая дала возможность реализовать хранение постоянные классов;
- определение стратегий и механизмов сохранения и восстановления постоянных данных, обеспечивающие требуемую производительность.

База данных и механизмы хранения и поиска постоянных данных, реализуются и тестируются как часть полной реализации компонентов и подсистем приложения.

К постоянным классам можно отнести классы (модели проектирования) со стереотипом «entity», а также некоторые классы со стереотипом «control», содержащие данные о состоянии системы. Необходимо принимать во внимание, что реляционная модель не поддерживает операции

классов, а модель классов не поддерживает триггеров и хранимых процедур.

Далее будет описан механизм преобразования модели классов в реляционную модель данных.

Отображение (миграция) классов.

Для того чтобы классы можно было отобразить в реляционные таблицы, необходимо, чтобы атрибуты класса были бы атомарными. Обычно классы «entity» удовлетворяют этим требованиям. В тоже время классы ассоциаций содержат атрибуты–списки или атрибуты-массивы.

Класс «entity» просто преобразуется в таблицу с соответствующим названием, в которой поля соответствуют атрибутам, атрибут выполняющий роль первичного ключа или Oid становится полем первичного ключа.

В том случае если существует множественный атрибут, для него нужно создать отдельную таблицу, в которой, кроме того, должно быть создано поле с внешним ключом, ссылающимся на первичный ключ родительской таблицы (откуда был вынесен множественный атрибут).

Отображение ассоциации.

Любую ассоциацию кратности «один к одному» или «один ко многим» можно непосредственно выразить с помощью помещения внешнего ключа в одну из таблиц (на стороне много) для установления соответствия первичному ключу другой таблицы (на стороне один). Для ассоциации один к одному выбор размещения первичного и внешнего ключа схожа с

Класс, который частично участвует в ассоциации, определяется как родительский, а тот который, который участвует в связи полностью, определяется как дочерний (в этой таблице располагается внешний ключ ссылается на первичный ключ родительской таблицы). В том случае если участие в ассоциации обеих классов частичное, то выбор дочернего и родительского класса произвольный.

Для рекурсивных ассоциаций "один к одному" и "один ко многим" создается отдельная таблица, как показано 5.10 (таблица подчиненный преподаватель)



Рисунок 5.10 – Миграция рекурсивной ассоциации.

Каждая ассоциация «многие ко многим» (не зависимо от того, рекурсивна она или нет) требует введения дополнительной таблице как показано на рисунке 5.11 (таблица «дисциплина специальности»).

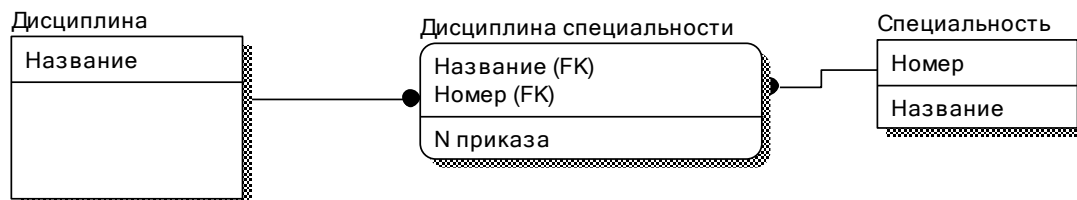


Рисунок 5.11 – Миграция ассоциации многое ко многим между классами «Дисциплина» и «Специальность»

Отображение агрегаций.

Реляционная модель не различает отношений ассоциации и агрегации за исключением случаев их процедурной реализации с помощью триггеров или хранимых процедур, Основные принцип отображения ассоциаций применимы и к отображению агрегаций. Композиция может быть реализована за счет использования идентифицирующей связи.

Отображение обобщения.

Для отображения обобщения можно использовать методику, описанную в приложении И для преобразования связи суперкласс/ подкласс (раздел логическое моделирование).

Для отображения диаграммы классов в реляционную модель можно воспользоваться инструментами предоставляемыми CASE. Так, например, Rational Rose предоставляет средство Data Modeler.

6. Применение проектных методологий при выполнении дипломных работ

Часто тема предлагаемая руководителем, можно рассматривать только как начальную идею, которая должна быть проработана, прежде чем приступить к формированию задания. Формулировка идеи - это начала жизненного цикла информационной системы. К сожалению, в очень редких организациях существует описание бизнес процесса. Для каждой темы, представленной в этой главе, приводится предварительное описание бизнес процесса для того, что бы соотнести определение требований к информационной системе с реальными потребностями.

В данной главе рассматриваются примеры использованием структурного (тема 1- «Информационная система распределения нагрузки по кафедре») и объектно-ориентированного подходов (тема 2 - «Информационная система поддержки дистанционного проведения практических и индивидуальных занятий. Рамки пособия не дают возможность привести полностью дипломные работы и тем более весь материал, который был получен студентами при разработке данных тем. Приводятся только фрагменты дипломных работ, поэтому может возникнуть впечатление их не проработанности. В тоже, как нам кажется, приведенный материал иллюстрирует последовательность выполнения проектов, и форму представления результатов.

ТЕМА 1 «Информационная система распределения нагрузки по кафедре»

Данная тема выполнялась в Тюменском государственном университете в рамках дипломной работы по специальности «Прикладная информатика в экономике» студентом группы 335-2 В. С. Луценко. В этой главе

приводятся фрагменты этой дипломной работы, иллюстрирующие применение структурного подхода.

Описание задачи

В Тюменском государственном университете каждый заведующий кафедрой ежегодно проводит работу по планированию учебной нагрузки на кафедре. Возникла идея создать информационную систему поддержки этой деятельности. Данная система может рассматриваться как один из модулей системы управления учебным процессом в университете.

Планирование учебной нагрузки на кафедре производится в несколько этапов.

- Первый этап (апрель) – планирование штата кафедры. Регламентируется выходной документ «Расчет часов и штата по кафедре» (Приложение Д, Рисунок Д.1, Таблицы Д.1, Д.2).
- Второй этап (май - июнь) – распределение нагрузки между преподавателями. На этом этапе нужно определиться (ориентировочно) со списком преподавателей, дисциплинами, которые они должны будут вести (в первую очередь необходимо определить лекторов по дисциплинам). Выходной документ не регламентируется.
- Третий этап (август - сентябрь) – распределение нагрузки по кафедре. На этом этапе должна быть распределена вся аудиторная нагрузка между преподавателями. Выходной документ не регламентируется.
- Четвертый этап (сентябрь – октябрь)- составление плана работы кафедры. На этом этапе для каждого преподавателя должно быть составлено «Учебное поручение преподавателю кафедры» (таблица Д.8), а также, план работы кафедры (Таблицы Д.3 ... Д.6).

- Пятый этап (весь учебный год) - внесение изменений распределения нагрузки, связанных с: изменением контингента студентов, увольнением преподавателя или приходом нового (Таблица Д.8).
- Шестой этап (июнь) – подготовка отчета кафедры (Таблицы Д3...Д7).

Фактически, рассматривается два представления учебной нагрузки, выполняемой на кафедре: учебная нагрузка по дисциплинам и компонентам (рисунок Д.1) и учебная нагрузка по преподавателям (таблица Д.6). Учебная нагрузка по отдельной дисциплине или компоненту определяется суммированием часов, отводимых для их проведения. Так, например, количество часов, отводимых на дисциплину, соответствует сумме часов на проведение лекций, практических занятий и/или лабораторных работ, консультаций по дисциплине, рецензирования контрольных работ, консультаций по курсовым работам, индивидуальных консультаций, консультациям перед экзаменом, прием экзаменов, прием зачетов. Для каждой дисциплины множество приведенных характеристик соответствует требованиям учебного плана (количество часов лекционного курса, количество часов практических и лабораторных занятий на одну группу или на одну подгруппу, наличие курсовой и контрольной работы, проведение экзамена или/и зачета). Учебный план, кроме того, что определяет количество часов и формы контроля по дисциплине, определяет кафедру для проведения каждой дисциплины (Рисунок Д.2). Норма времени по каждому виду нагрузки определяется соответствующим приказом по университету. В приложении Д приведена таблица приложения к этому приказу (Таблица Д.12). Норма времени определяется в виде формулы:

$$\text{Количество часов для каждого компонента} = K_i * P,$$

где K_i – коэффициент, определяемый приказом;

$$P = \{ \text{часы учебного плана} \mid \text{количество студентов} \mid \text{количество} \}$$

групп | количество потоков | количество контрольных работ | количество недель практики и т.д. }.

В настоящем положении приводится 31 норма. Возможно добавление или удаление соответствующих норм, изменение коэффициентов K_i , а также расширение множества P .

Контингент студентов показан в форме, представленной в таблице Д.10 (приложение Д). Для каждой специальности на факультетах или в институтах распоряжением (таблица Д.11) закрепляется распределение студентов за выпускающими кафедрами для прохождения курсовых и дипломных работ. Этим же распоряжением закрепляется распределение студентов по специализациям. Контингент аспирантов и соискателей определяется приказом ректора об их зачислении с указанием научного руководителя.

Для проведения Государственных экзаменов и Государственной аттестации приказом ректора назначается комиссия.

При распределении нагрузки по преподавателям обычно используют следующие правила:

- лектор проводит консультации по дисциплине, консультации перед экзаменом и принимает экзамен;
- преподаватель, ведущий практику, проводит у этих же студентов консультации по курсовой работе по предмету, индивидуальные консультации и прием контрольных работ и прием зачетов (в некоторых случаях, зачеты принимает лектор).
- лектор ведет практические занятия хотя бы в одной группе;
- студенческая группа на практических занятиях не делится на подгруппы, кроме случаев, закрепленных приказом по факультету/институту (в приказе перечисляются дисциплины, на которых допускается деление группы на подгруппы, в случае если в группе больше 20 студентов);

- могут использоваться другие правила, в которых один вид нагрузки ассоциируется с другим видом нагрузки по этой же дисциплине.

Анализ требований к системе

Анализ проблем - это осознание реальных проблем и потребностей пользователей и предложение решения, позволяющих удовлетворить эти потребности. Цель анализа проблем - добиться лучшего понимания решаемой проблемы до начала разработки /31/.

Изучение бизнес - процесса в нашем случае облегчается тем, что в нем участвуют, обычно, один или два человека (заведующий кафедрой и/или лаборант кафедры).

Постановку проблемы можно отразить по следующему шаблону.

- Проблема воздействует на <перечислить заинтересованные лица, на которых она воздействует, и указать результат воздействия проблемы>.
- Решение проблемы может состоять <указать возможный вариант решения>.
- Выигрыш от решения проблемы может состоять <указать эффект, который может быть достигнут>.

В данном случае были выявлены следующие проблемы:

- Заведующий кафедрой многократно выполняет одну и ту же работу.
- Работа, как правило, выполняется либо на листе бумаге, либо в Excel. Результат работы требуется переносить в выходные документы вручную.
- Алгоритм распределения нагрузки не определен (каждый заведующий кафедрой использует свои трудно формализуемые методы), поэтому распределение нагрузки требует большого времени, даже в том случае, если кафедра малочисленная.

- Вызывает трудность привлечения лаборанта к этому процессу. Как правило, лаборант переносит данные, подготовленные заведующим кафедрой, в шаблон выходного документа.
- Высокая вероятность возникновения ошибок (либо нагрузка не распределена, либо одни и те же поручения закреплены за различными преподавателями).
- Трудно проследить причину невыполнения нагрузки преподавателем.
- Трудно прогнозировать нагрузку на следующий семестр.
- Изменение контингента студентов или изменения в штате преподавателей требует повторное выполнение всей работы.

Для выделения основных проблем и корневых причин их возникновения, можно построить диаграмму «рыбного скелета» (диаграмма Ишикавы).

На рисунке 6.1 приведена диаграмма причинно-следственных связей для данной предметной области. Эта диаграмма показывает, что для уменьшения времени распределения учебной нагрузки по преподавателям (определение учебных поручений) и уменьшения ошибок, возникающих в этом процессе, требуется создать информационную среду, обеспечивающую:

- Единое хранение информации с генерацией разнородных выходных форм;
- Реализацию проверки соответствия распределенной нагрузки правилам;
- Автоматический расчет нагрузки в соответствии с распределенными учебными заданиями;
- Сохранение планируемой нагрузки (возможно в нескольких вариантах) в течение всего учебного года.

Цель создания системы определяется следствием диаграммы Иши-кавы - достижения эффекта в бизнесе: «Уменьшение затрат времени при планировании нагрузки по кафедре, а также, снижение ошибок, возникающих в результате этого процесса».

Данный проект может воздействовать на заведующего кафедрой, преподавателей кафедры, лаборанта кафедры и учебный отдел. Предполагаемыми пользователями системы являются заведующие кафедрой и лаборанты кафедр.

Следующий этап определения требований - определение границ системы. В различных нотациях описания бизнес - процесса граница системы описывается различным инструментом. В DFD диаграммах это внешние сущности, в UML – это акторы, в IDEF0 это все то, что находится за контекстным процессом и т.д.

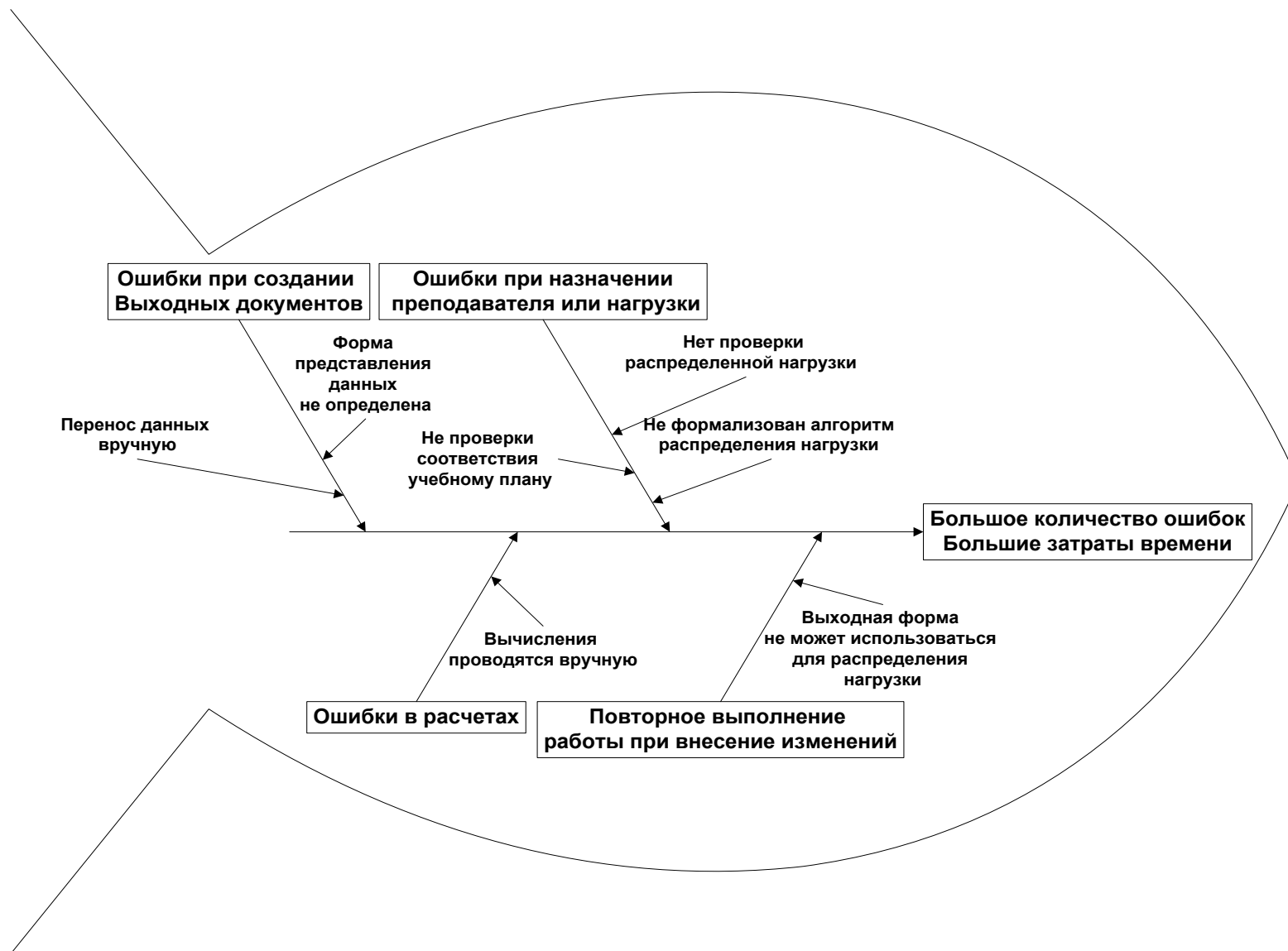


Рисунок 6.1 - Диаграмма причинно-следственных связей.

Каждая из диаграмм отвечает на определенные вопросы. Но в тоже время, на этапе анализа проблем нам необходимо разделить мир на два класса: наша система и окружающая среда.

Если мы рассматриваем структурные модели IDEF0 и DFD, то в этом случае мы рассматриваем систему как бизнес-функцию, в которой используется информационная система. Определение границ системы в этом случае осуществляется спецификацией функций, в которых участвует наша информационная система.

Если мы рассматриваем модель UML, то в этом случае наша система – совокупность прецедентов, с которыми взаимодействуют акторы. Она отображается на диаграмме прецедентов.

И в том, и в другом случае нам нужно получить ответ на вопросы /31/:

- Кто будет поставлять, использовать и удалять информационную систему?
- Кто будет управлять системой?
- Кто будет осуществлять сопровождение системы?
- Кто будет использовать систему?
- Откуда система получает информацию?
- Какие внешние системы будут взаимодействовать с системой?

Построение функциональной модели (ТО-ВЕ) выявление требований к обеспечивающим частям ИС.

Диаграммы потоков данных в контексте «ТО-ВЕ» дает возможность определить точки взаимодействия пользователя с информационной системой. На рисунке 6.2 и 6.3 приведены два уровня декомпозиции этой диаграммы, а на рисунке 6.4 приводится фрагмент словаря данных. В нашем случае выделен один пользователь (роль - заведующий кафедрой), который осуществляет сохранение всех требуемых данных в системе и он же проводит назначение нагрузки преподавателям. Объектами автоматизации являются

ся расчет часов учебной нагрузки по кафедре и формирование заявки на чтение курсов.

Организационное обеспечение и информационное обеспечение демонстрируется диаграммой DFD и спецификацией ее процессов (не приводится).

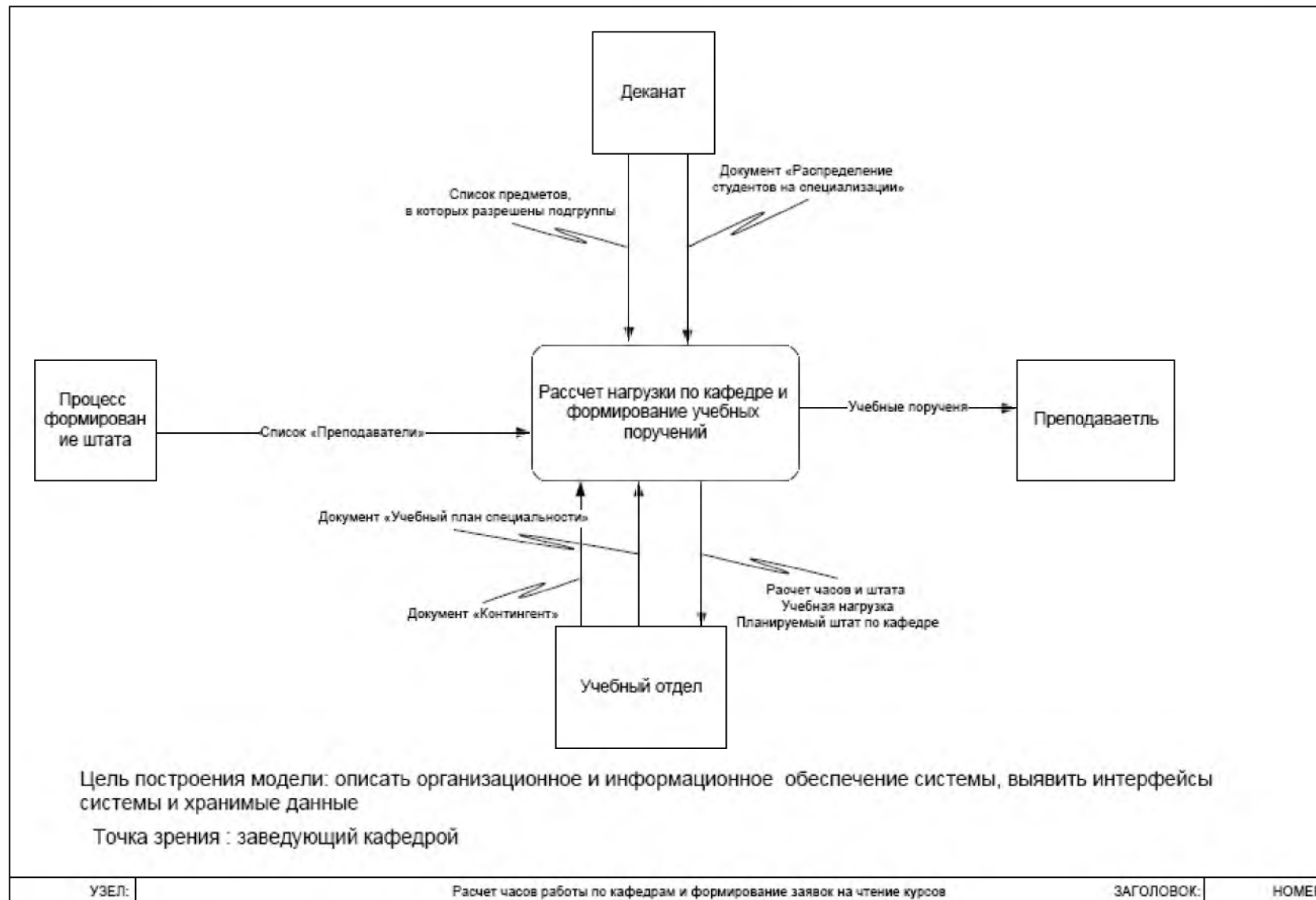


Рисунок 6.2 - Контекстная диаграмма «Расчет часов работы по кафедре и формирование заявок на чтение курсов».

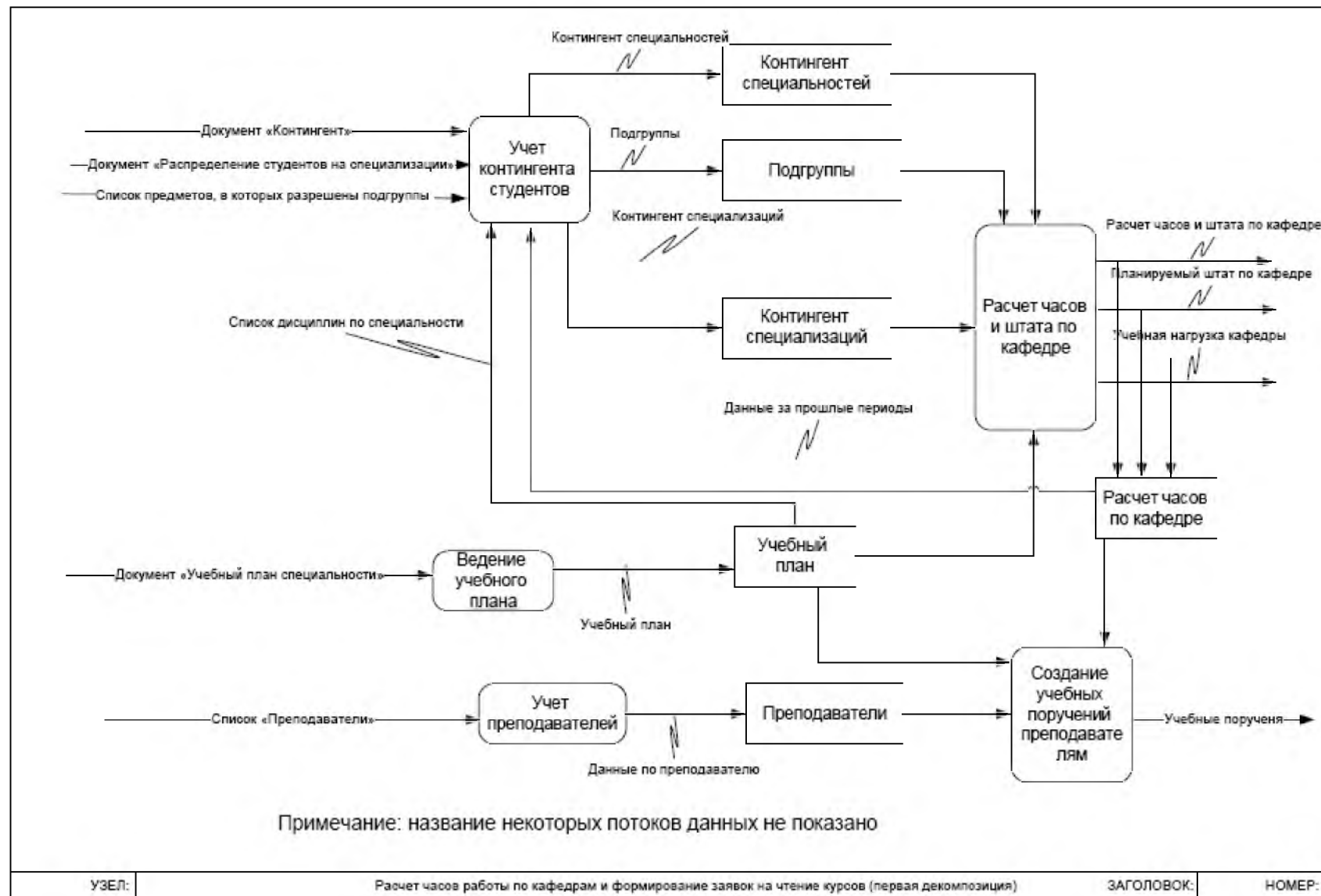


Рисунок 6.3 - Декомпозиция диаграммы .

Словарь данных

@ИМЯ = Учебный план

@ТИП = простой, дискретный, внешний, поток данных

@БНФ = /код специальности + название специальности + № семестра + название кафедры + название дисциплины + учебный год + факультатив + название цикла + название компонента + вид занятий + кол-во часов + форма отчетности + кол-во недель в семестре/

@ИМЯ = Контингент специальностей

@ТИП = простой, дискретный, внешний, поток данных

@БНФ = /Название факультета + название специальности + № курса + год специальности + кол-во студентов + кол-во групп + кол-во потоков/

Рисунок 6.4 Фрагмент словаря данных

Схема данных

Для проектирования базы данных используется методология "сущность-связь". Первым шагом в процессе проектирования баз данных является выделение сущностей, их атрибутов, после устанавливают связи между сущностями.

На рисунках 6.5 ... 6.9 приводятся мже приводятся

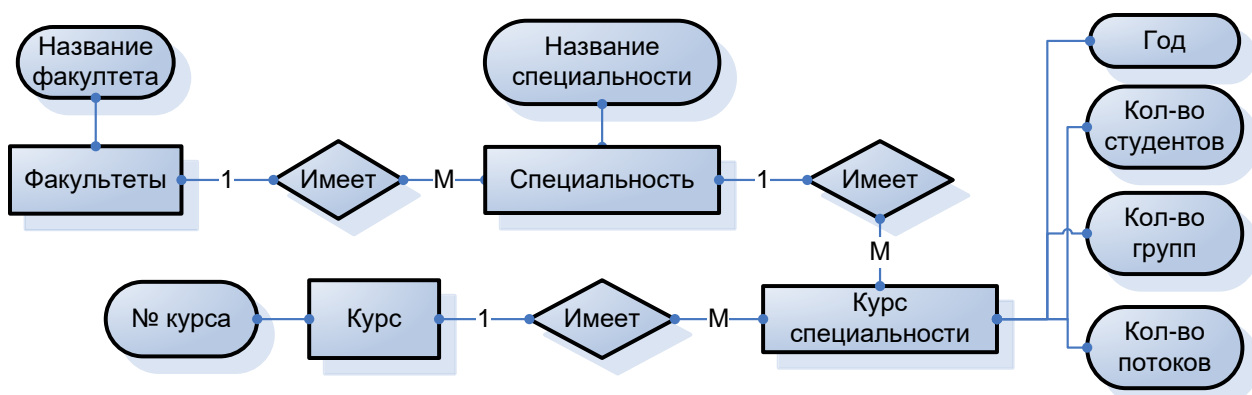


Рисунок 6.5 Диаграмма ERD – «Контингент специальностей»

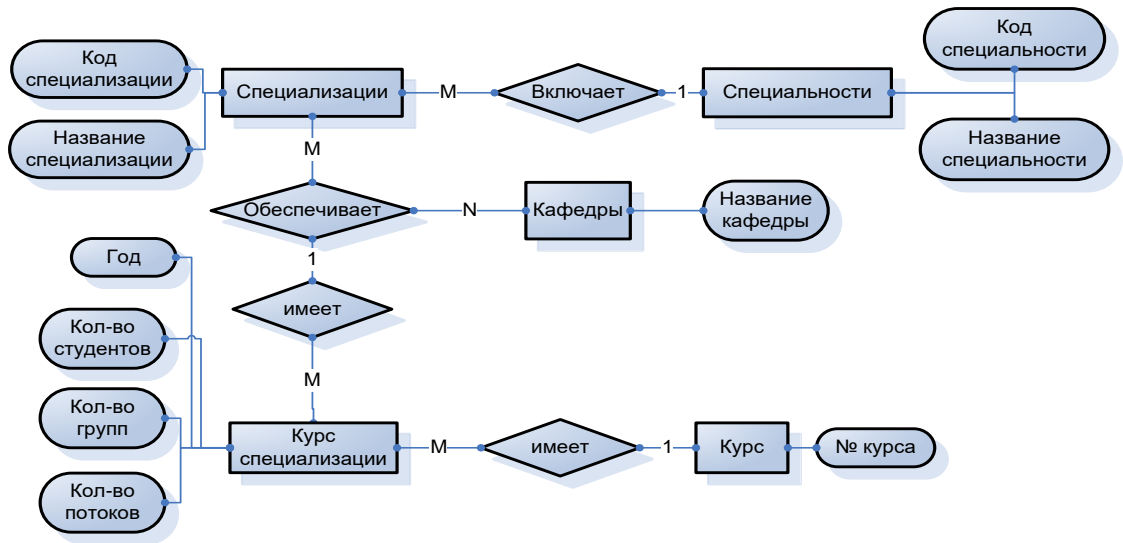


Рисунок 6.6 - Диаграмма ERD «Контингент специализаций»

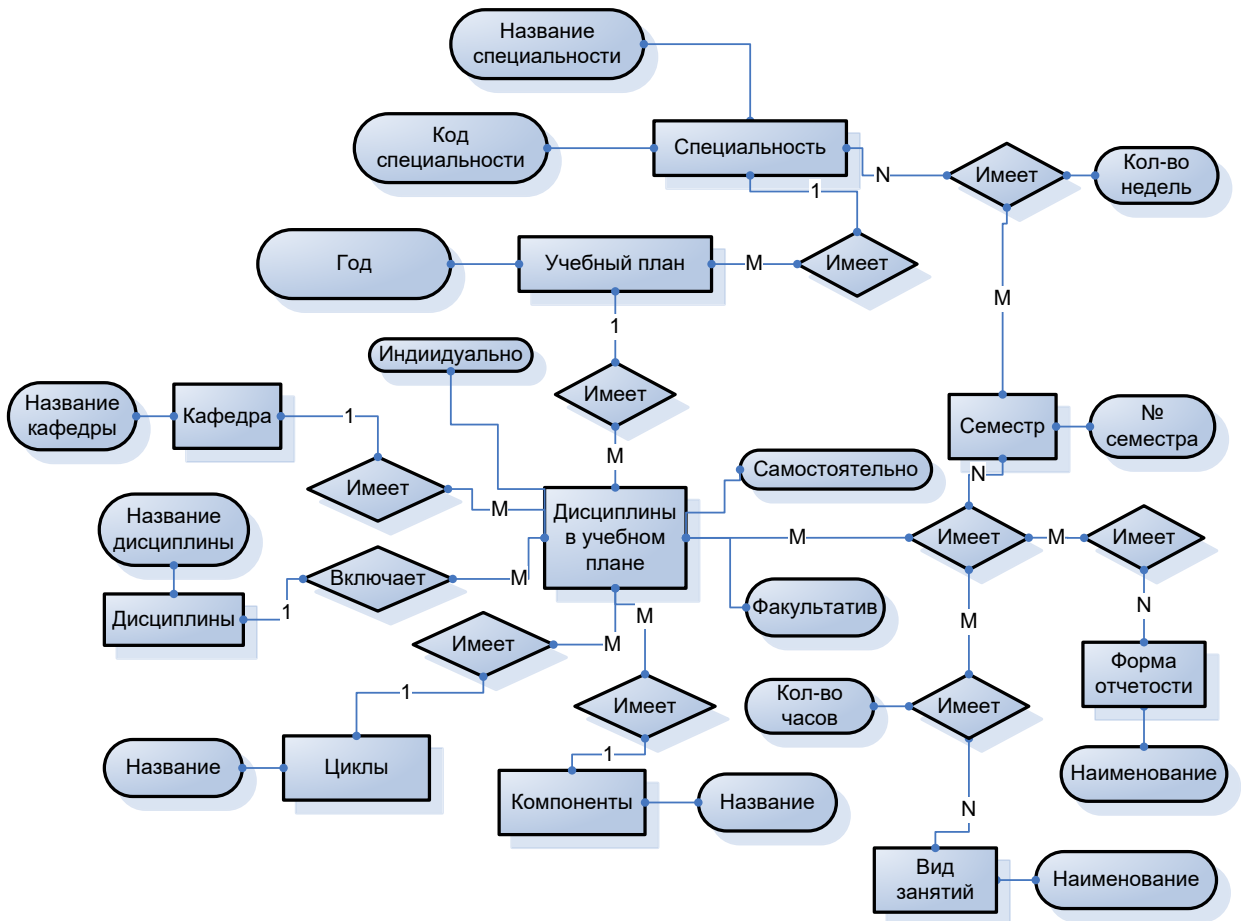


Рисунок 6.7 Диаграмма ERD «Учебный план»

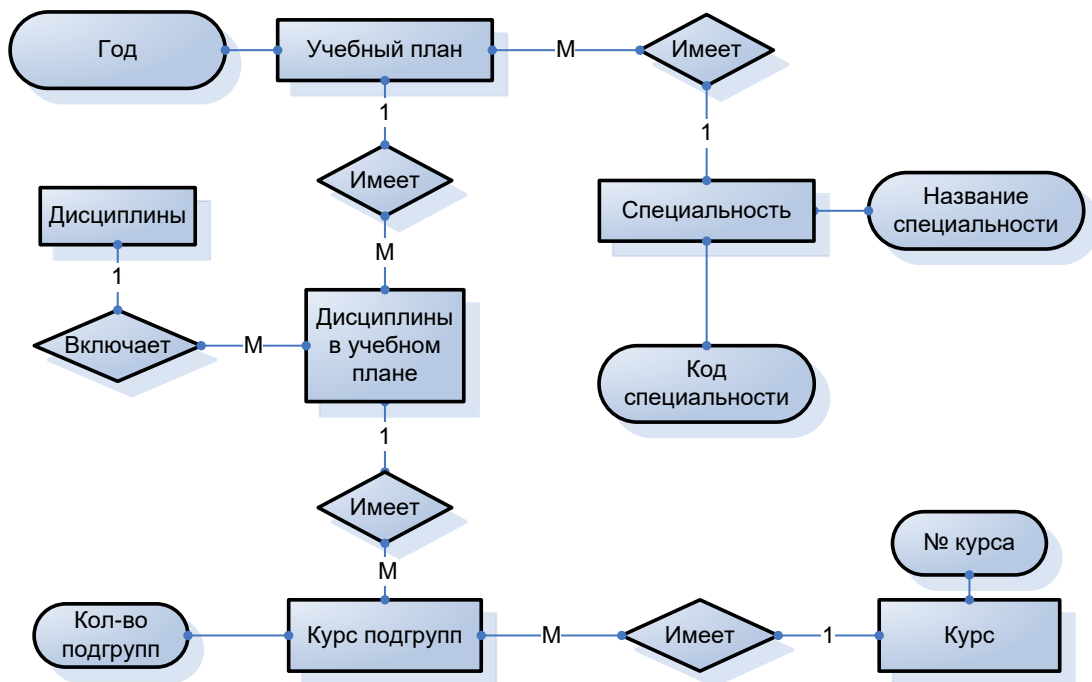


Рисунок 6.8 - Диаграмма ERD «Подгруппы»

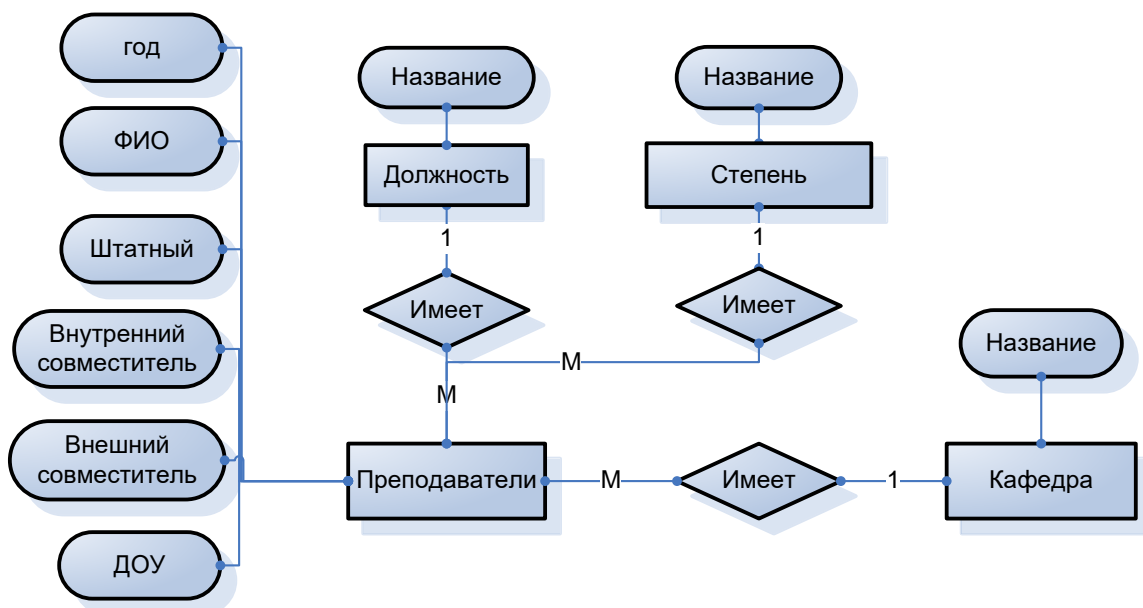


Рисунок 6.9 Диаграмма ERD «Преподаватели»

Определение логической структуры базы данных.

Для проектирования логической структуры базы данных используется методология IDEF1x. Логическое проектирование заключается в преобразовании модели сущностей ERD в логическую модель IDEF1x и преобразова-

нии локальных схем данных в глобальную логическую модель данных, представленных на рисунке 6.10.

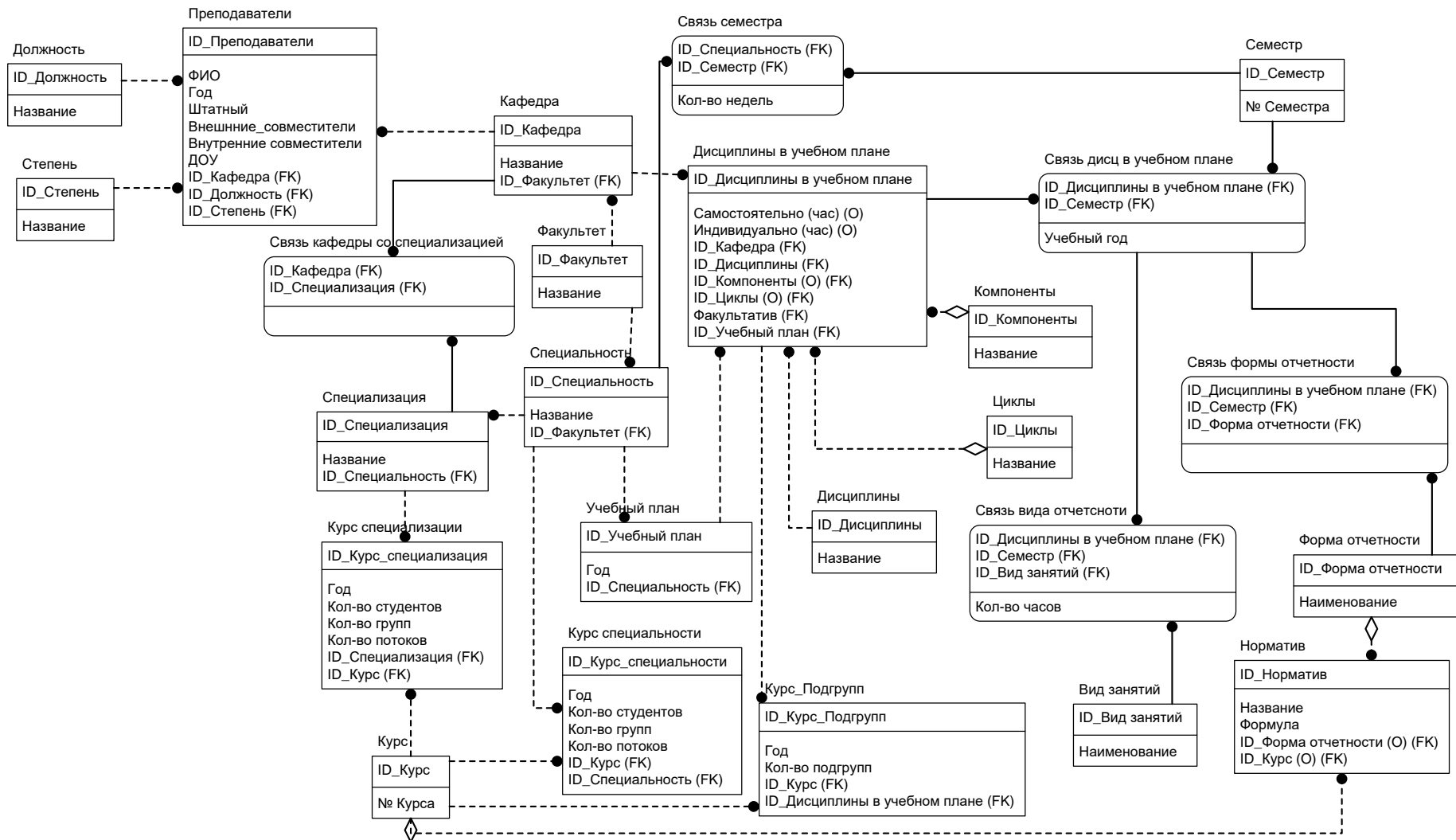


Рисунок 6.10 – глобальная логическая модель данных

В таблице 6.1 представлен пример описание физических таблиц базы данных созданных в MSSQL 2000

Таблица 6.1 –Пример описания состава таблиц физической схемы данных.

Specialnost	Таблица специальностей	ID	Уникальный номер специальности	Integer
		Name	Название специальности	Varchar (255)
		ID_Fakultet	Идентификатор факультета, к которому относится данная специальность	Integer

Доступ в Delphi к SQL Server осуществляется через ADO (ActiveX Data Objects)-компоненты, которые инкапсулирует основную функциональность OLE DB в виде тонкой IDispatch-обертки. Такая технология дает возможность средствам разработки использовать Automation (OLE DB), представляющие COM-сервера с реализацией предопределенного набора COM-интерфейсов.

Руководство пользователя.

Работа приложения начинается с окна Учебный план (рисунок 6.11), на котором отображается информация об учебном плане.

Для отображения учебного плана следует выбрать Специальность (кнопка «...» справа от поля специальность), чей учебный план собираются просматривать. Далее нужно выбрать либо ввести год учебного плана. Также нужно выбрать на закладках интересующий цикл и выбрать из списка необходимый компонент. Если было все сделано правильно, и по выбранным параметрам существуют данные, то они отобразятся в таблице.

Учебный план

Учебный план Контингент Норматив Преподаватели Каталог Отчеты

Учебный план на
 Специальность ... Год

Дисциплины специализации в области применения ...

Компонент

Наименование дисциплины	Индивидуально	Самостоятельно	Кафедра
▶ Иностранный язык	35	25	Ин. языков ЕФ
Отечественная история	7	53	Отеч. истории
Физическая культура	44	14	Физвоспитания
Экономика	7	53	Политэкономии и РЭ

... ...

Семестр

Форма отчетности

Зачет Коллоквиум
 Контрольная работа Курсовая работа
 Экзамен

Вид занятий

Лабораторные
 Лекции
 Практика
 Недель

Рис. 6.11 - Учебный план

При нажатии в главном меню на «Контингент» отобразится форма «Специальности и специализации» (рисунок 6.12) на которой можно добавлять, изменять и удалять специальности и специализации, а также редактировать контингент по выбранным специальностям.

Специальности и специализации

Учебный план Контингент Норматив Преподаватели Каталог Отчеты

Специальности

Все Факультет Математики и компьютерных наук ... Год 2007 Контингент

Код	Название	№ Курса	Студ	Групп	Потоков
10100	Математика	1	50	2	1
351500	МОиАИС	2	37	2	1
351400	ПИВЭ	3	98	4	1
		4	118	5	2
		5	92	4	1

351400 ПИВЭ

Добавить Изменить Удалить

Специализации по специальности ПИВЭ Контингент

Код	Название специализации	Название кафедры	№ Курса	Студ	Групп	Потоков
351413	Информационные системы в бизнес - реинжини...	Информационных систем	3	34	2	1
351413	Информационные системы в бизнес - реинжини...	Программного обеспечения	4	45	2	1
351414	Корпоративные экономические информации	Информационных систем	5	0	0	1
351414	Корпоративные экономические информации	Программного обеспечения				

351413 ионные системы в бизнес - реинжиниринге Информационных систем ...

Добавить Изменить Удалить Удалить связь с кафедрой

Рис. 6.12 - Специальности и специализации

Далее можно перейти к форме норматив (рисунок 6.13) нажав в главном меню «Норматив». На этой форме можно производить добавление, изменение и удаление нормативов. Для добавления в первом пустом поле (слева от знака «=») следует записать название норматива, после выстроить метод расчета норматива с помощью выпадающих списков, что справа от знака «=». Для изменения следует просто изменить значение выпадающего списка. Для удаления норматива следует нажать правой кнопкой мыши на поле, где содержится название норматива и нажать «Удалить».

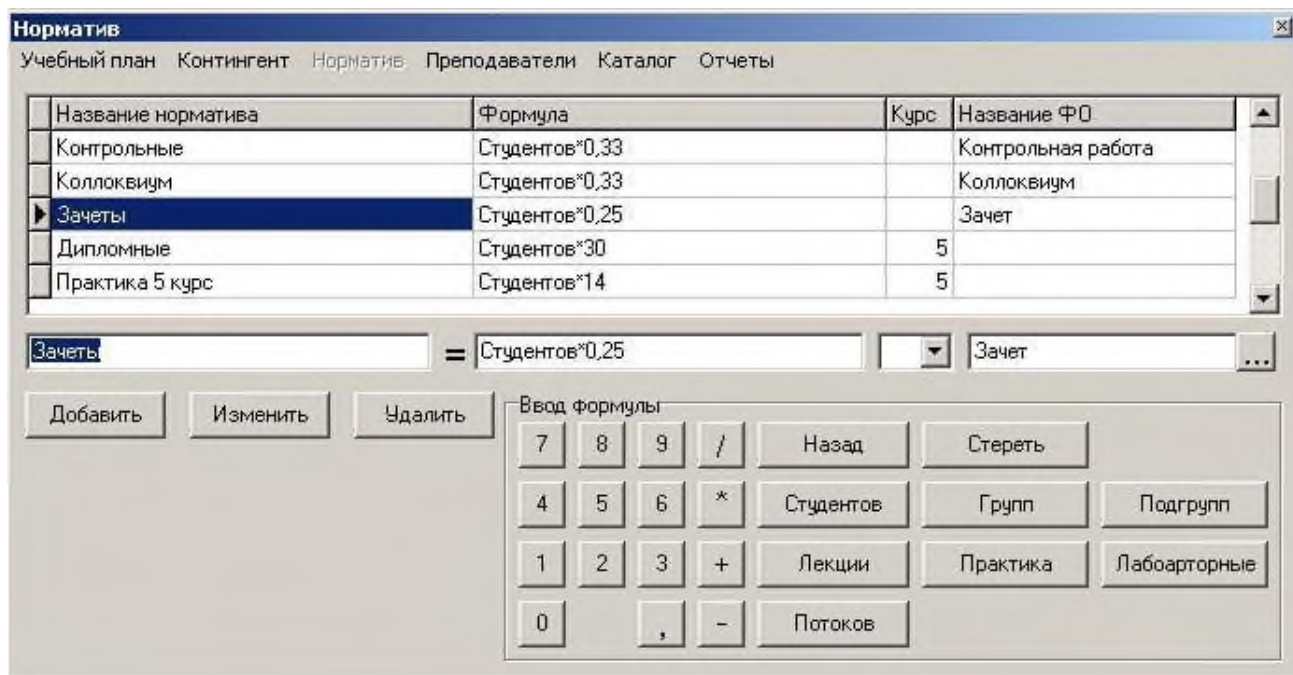


Рис. 6.13 - Норматив

Для перехода к форме «Преподаватели» (рисунок 6.14) следует нажать на главном меню «Преподаватели». На форме отображаются агрегированные данные по всем преподавателям.

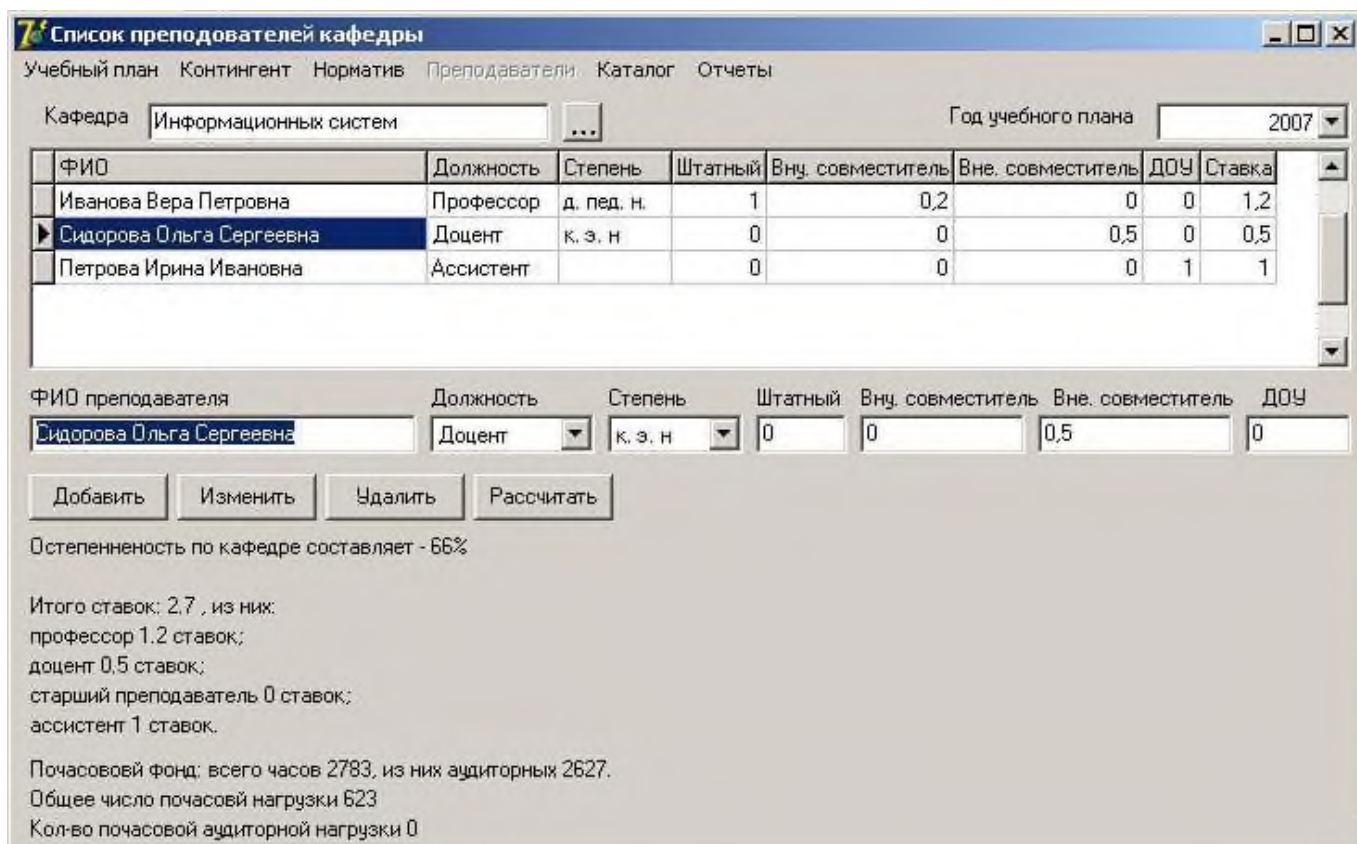


Рис. 6.14 - Преподаватели.

Остальное описание дипломной работы не приводится.

ТЕМА 2 «Информационная система поддержки дистанционного проведения практических и индивидуальных занятий»

Данная тема выполнялась в Тюменском государственном университете в рамках дипломной работы по специальности «Прикладная информатика в экономике» студентом группы 335-1 А. Л. Шарикова (дипломная работа признана лучшей дипломной работы ТЮмГУ за 2008г. по данной специальности). В этой главе приводятся фрагменты этой дипломной работы, иллюстрирующие применение объектно-ориентированного подхода.

Описание задачи

В настоящее время не существует бизнес – процесс «дистанционное проведение практических и индивидуальных занятий». Есть некоторый опыт проведения занятий с передачей файлов по E-mail, а также, опыт работы Института дистанционного образования (ИДО).

Прием контрольных работ и лабораторных работ по E-mail.

Преподаватель проводит очные практические занятия, где студенты не успевают сдать выполненное задание. В этом случае, он принимает отчеты на свой электронный адрес, рецензирует их и отправляет обратно студенту. Когда выполненное задание соответствует предъявляемым требованиям, он отмечает его выполнение. Возможна окончательная защита практической работы при проведении очных занятий (используется для студентов филиалов). Такая же последовательность применяется при сдаче контрольных работ.

Прием контрольных работ в ИДО.

В приложение Е приводится описание бизнес – процесса и используемых документов. Преподаватель выдает задание для выполнения курсовых и контрольных работ специалистам. Специалист размещает задание на сайте ИДО. После того, как студент выберет задание и выполнит его, отчет отправляется специалисту. Отчет сдается как в печатном, так и в электронном виде. Специалист рассылает отчеты пре-

подавателям кафедр. Преподаватели проверяют работу, пишут рецензию и отправляют ее специалисту. Специалист ИДО ведет учет работ, полученных от этого студента, и определяет стадию проверки (на проверке у преподавателя; отчет проверен, но есть замечания; замечания по рецензии исправлены, но преподавателем не проверены; и т. д.). Специалист так же ведет учет проверки работ преподавателем. Эта информация учитывается при оформлении акта приема - передачи услуг, который отправляется в бухгалтерию. Учет по каждому студенту и по преподавателям проводится в Excel (данные не сопоставляются).

Описание организационной структуры

Организационная структура объекта автоматизации

Кафедра информационных систем (кафедра ИС) является основным учебно-научным структурным подразделением университета, осуществляющим учебную, методическую, научную деятельность, воспитательную работу со студентами, а также подготовку научно-исследовательских кадров и повышение их квалификации [3].

Преподаватели кафедры ИС проводят занятия по дисциплинам "Информатика", "Математика и информатика", "Базы данных" и другим специальным дисциплинам в области информационных технологий практически на всех факультетах университета.

В рамках текущего проекта в качестве объекта автоматизации рассматривается учебный процесс, затрагивающий взаимодействие преподавателей кафедры со студентами университета, обучающимися в территориально удаленных филиалах (Нижневартовск, Урай, Ишим, Нягань, Муравленко, Новый Уренгой, Ноябрьск).

Рассмотрим элементы учебного процесса и отразим их особенности, проявляющиеся в исследуемом процессе:

- 1) Взаимодействие студентов с преподавателем – учет и контроль заданий, выполняемых студентами, а также проведение консультаций по назначенным заданиям.

- 2) Предоставление учебных материалов обучающимся – передача студентам заданий, определенных для выполнения.
- 3) Контроль знаний обучающихся – учет назначенных заданий и их оценок, подготовка отчетов по задолженностям студентов, по состоянию выполнения заданий.
- 4) Генерирование сводных статистических контрольных точек успеваемости – подготовка сводных отчетов по филиалам, по дисциплинам, по успеваемости студентов определенной группы.
- 5) Администрирование групп обучающихся – формирование списков групп студентов, определение преподавателя на дисциплину, читаемую в филиале.

Нормативно-справочная информация

Кафедра информационных систем в своей непосредственной деятельности руководствуется решениями Ученого совета университета, Ученого совета института математики и компьютерных наук, Правилами внутреннего распорядка, приказами ректора и Положением о кафедре, разработанным в соответствии с Законом Российской Федерации «Об образовании» от 13.01.96 №12-ФЗ, Федеральным законом «О высшем и послевузовском профессиональном образовании» от 22.08.96 №125-ФЗ, Типовым положением об образовательном учреждении высшего профессионального образования Российской Федерации (приказ Минобрнауки РФ №1870 от 26.06.2001), Уставом университета и другими законодательными и правовыми актами, регламентирующими учебную деятельность высших учебных заведений. [3]

Изменения в организационной структуре объекта автоматизации

Учитывая особенности исследуемого процесса, была разработана модель взаимодействия преподавателя со студентами, обучающимися в территориально удаленных филиалах. Обобщенная схема модели взаимодействия представлена на рисунке 1.

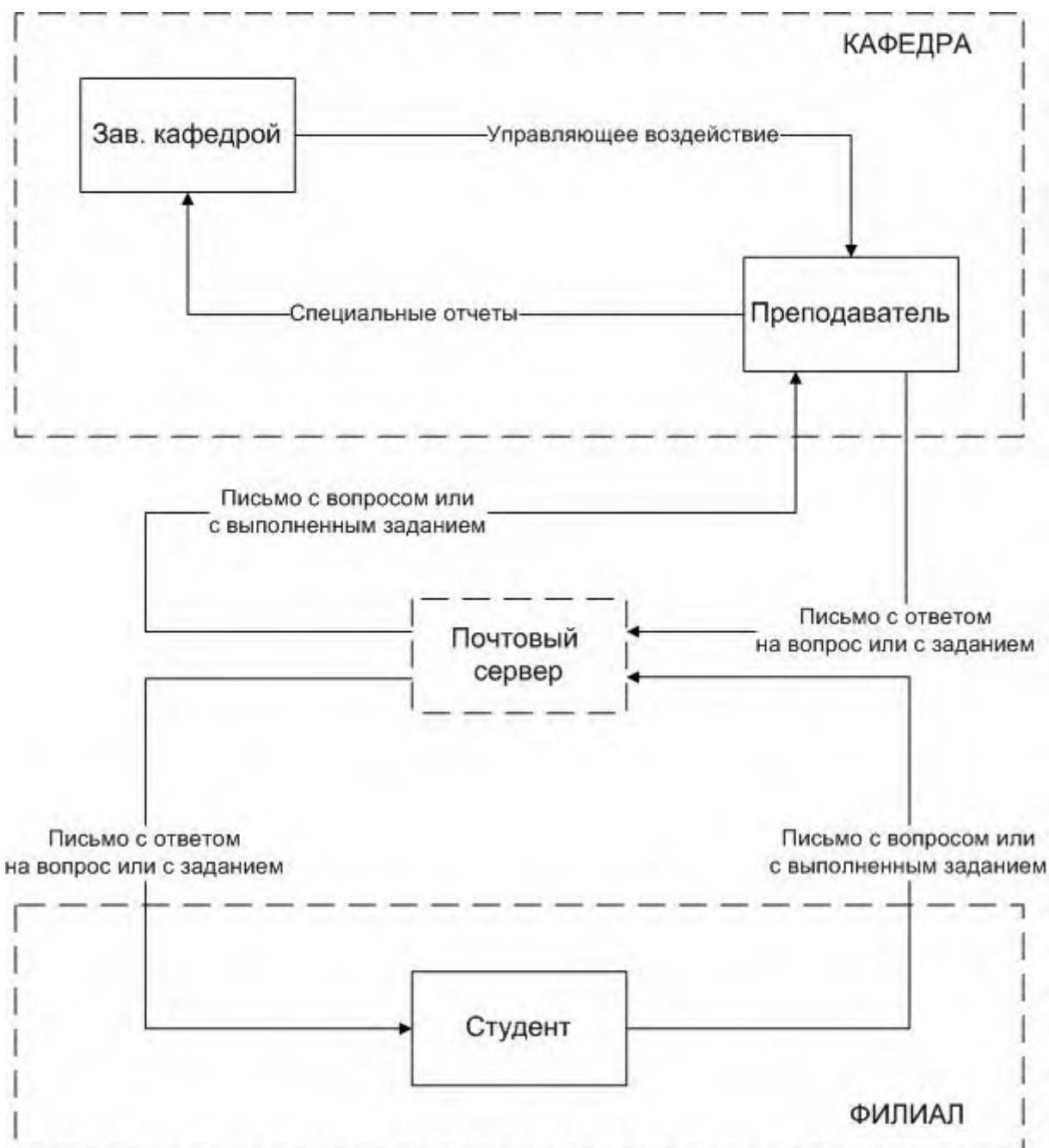


Рисунок 1 – Обобщенная модель взаимодействия преподавателя со студентами филиалов

Разработанная модель обладает следующими преимуществами:

- обеспечивает непрерывность учебного процесса;
- дает более равномерное распределение нагрузки как для преподавателя, так и для студента;
- имеет менее выраженную зависимость от ограничений доступа к сети Интернет.

Постановка задачи

Описание постановки задачи выполнено согласно стандарту ГОСТ 36.602-89 “Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы” [4].

Общие сведения

Наименование системы

Полное наименование системы: “Автоматизированная система “Offline-система учета и контроля выполнения заданий в дистанционном образовании”.

Сокращенное наименование системы: “АС “Offline-система учета и контроля выполнения заданий в дистанционном образовании”.

Наименование предприятий разработчика и заказчика

Предприятие-разработчик:

ГОУ ВПО «Тюменский государственный университет»

Институт математики и компьютерных наук

Ответственное лицо: студент Шарикова Анастасия Леонидовна;

Предприятие-заказчик:

ГОУ ВПО «Тюменский государственный университет»

Институт математики и компьютерных наук

Кафедра информационных систем

Ответственное лицо: заведующий кафедрой ИС Ивашко Александр

Григорьевич.

Плановые сроки выполнения работ

Начало выполнения работ: декабрь 2007 года.

Окончание выполнения работ: июнь 2008 года.

Источник финансирования

ГОУ ВПО «Тюменский государственный университет».

Назначение и цели создания системы

Назначение системы

“Offline-система учета и контроля выполнения заданий в дистанционном образовании” является автоматизированной системой управления и предназначена для использования на кафедре информационных систем Института математики и компьютерных наук ГОУ ВПО «Тюменский государственный университет» в процессе работы преподавателей кафедры со студентами, обучающимися в филиалах университета.

Перечень автоматизируемых пунктов:

1. Деятельность заведующего кафедрой, направленная:

- на определение количества студентов в группе, имеющих задолженности по различным дисциплинам;
- на определение количества студентов в филиале, имеющих задолженности по различным дисциплинам;
- на определение количества студентов, имеющих задолженности по определенной дисциплине;
- на определение состояния выполнения лабораторных работ студентами.

2. Деятельность преподавателя кафедры, направленная:

- на определение количества студентов в группе, имеющих задолженности по конкретной дисциплине;
- на определение состояния выполнения лабораторных работ студентами определенной группы;
- на предоставление студентам формулировок лабораторных работ.

Цели создания системы

- Обеспечить взаимодействие преподавателя со студентами, обучающимися в филиалах, в течение всего семестра.
- Снизить затраты времени преподавателя на сортировку и распределение почты.
- Снизить затраты времени преподавателя на подготовку отчетов по задолженностям студентов, по успеваемости студентов.
- Снизить затраты времени заведующего кафедрой на подготовку сводных отчетов.

Требования к системе

Требования к системе в целом

В процессе разработки системы должны быть реализованы все поставленные цели.

Программное средство “Offline-система учета и контроля выполнения заданий в дистанционном образовании” должно представлять собой инструмент взаимодействия преподавателя со студентами, обучающимися в филиалах университета, в режиме offline. Кроме того, система должна иметь встроенный почтовый клиент, по функциональным возможностям соответствующий стандартным почтовым клиентам и реализующий сортировку и распределение писем по филиалам, по специальностям, по курсам и по группам студентов.

С целью упрощения подготовки отчетов, предусмотренных учебным процессом, необходимо иметь инструмент для формирования и назначения заданий на выполнение студентам, а также для учета результатов выполнения заданий.

В целях безопасности необходимо предусмотреть разграничение прав доступа в систему.

Программное средство должно иметь стандартный пользовательский интерфейс, выполненный в соответствии с общепринятыми нормами, подчиняться общим

настройкам среды пользователя, должна содержать проверку ввода информации и справочную систему.

Требования к функциям (задачам), выполняемым системой

Цель текущего проекта – разработка программного обеспечения, которое упрощает для преподавателей кафедры проведение учебного процесса, направленного на обучение студентов территориально удаленных филиалов. При реализации проекта предлагается использовать модель взаимодействия преподавателя со студентами, представленную на рисунке 1.

Следовательно, для достижения поставленной цели проектируемая ИС должна предусматривать наличие комплексов задач, подробная характеристика которых представлена в таблице 6.2.

Таблица 6.2 – Характеристика комплекса задач³⁰

Название комплекса задач	Описание комплекса
Назначение заданий группе	<p><u>Назначение:</u> для дальнейшего учета выполнения заданий студентами.</p> <p><u>Распределение действий между пользователями и системой:</u> пользователь с правами доступа “Преподаватель” указывает дисциплину, задание и группу, которой необходимо назначить данное задание. Далее системой производится автоматическая рассылка формулировки задания на электронные почтовые адреса студентов.</p> <p><u>Условия прекращения работы:</u></p> <ul style="list-style-type: none"> • не внесены задания данным пользователем для данной дисциплины; • не заданы электронные почтовые адреса студентов.
Хранение истории переписки	<p><u>Назначение:</u> для отражения истории выполнения задания.</p> <p><u>Распределение действий между пользователями и системой:</u></p>

³⁰ Характеристика комплекса задач выполнена согласно РД 50-36.698-90 “Автоматизированные системы. Требования к содержанию документов” /22/.

Название комплекса задач	Описание комплекса
	<p>система автоматически выполняет распределение полученной почты в БД.</p> <p><u>Условия прекращения работы:</u></p> <ul style="list-style-type: none"> • неправильно сформирован заголовок входящего сообщения.
Учет успеваемости студентов	<p><u>Назначение:</u> для автоматизации подготовки отчетности по успеваемости студентов и по выполнению заданий.</p> <p><u>Распределение действий между пользователями и системой:</u> после назначения преподавателем задания группе в журнале успеваемости по умолчанию стоит статус выполнения задания “нет полученных писем”. При получении от студента первого письма по данному заданию, система автоматически проставляет статус выполнения "задание не принято или проверяется". После проверки окончательного варианта задания пользователь с правами доступа “Преподаватель” может поменять статус выполнения на “задание выполнено”.</p>
Распределение нагрузки преподавателей	<p><u>Назначение:</u> для автоматизации подготовки отчетности по успеваемости студентов и по выполнению заданий.</p> <p><u>Распределение действий между пользователями и системой:</u> на основе информации, полученной от заведующего</p>
	<p>кафедрой, пользователь с правами доступа “Администратор” вносит в систему данные по нагрузке преподавателей.</p> <p><u>Условия прекращения работы:</u></p> <ul style="list-style-type: none"> • не заполнены справочники “Филиалы”, “Группы”, “Дисциплины”; • не зарегистрирован пользователь (преподаватель).
Подготовка отчетов (Список должников, Состояние выполнения заданий, Сводные отчеты)	<p><u>Назначение:</u> для оперативного контроля деятельности объекта автоматизации в рамках автоматизируемого процесса.</p> <p><u>Распределение действий между пользователями и системой:</u> Пользователи с правами доступа “Преподаватель”, “Зав. кафедрой” имеют возможность сформировать отчет по выбранным параметрам (преподаватель, группа, дисциплина).</p>

Название комплекса задач	Описание комплекса
	<u>Условия прекращения работы:</u> <ul style="list-style-type: none"> • отсутствие программного обеспечения, используемого при формировании отчетов

Выходная информация

В рассматриваемой системе можно выделить 4 типа выходных потоков – Ведомость учета успеваемости студентов, Отчет по задолженностям, Отчет по состоянию выполнения заданий, Сводный отчет. Таблица 6.3 содержит информацию, характеризующую каждый тип.

Таблица 6.3 – Список выходных сообщений

№	Идентификатор	Форма	Периодичность	Получатель
1	Ведомость учета успеваемости	Документ по форме, представленной на рисунке 2	ежемесячно	Пользователи с правами доступа “Преподаватель” и “Зав. кафедрой”
2	Отчет по задолженностям	Документ по форме, представленной на рисунке 3	ежемесячно	Пользователи с правами доступа “Преподаватель” и “Зав. кафедрой”
3	Отчет по состоянию выполнения заданий	Документ по форме, представленной на рисунке 4	ежемесячно	Пользователи с правами доступа “Преподаватель” и “Зав. кафедрой”
4	Сводный отчет	Документ	ежемесячно	Пользователь с правами доступа “Зав. кафедрой”

Ниже представлены формы входящих документов.

Дата составления															
Преподаватель															
Дисциплина															
Филиал					Специальность										
Курс					Группа										
Задания	Пометка о выполнении														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ФИО студента															

Рисунок 6.14 – Ведомость учета успеваемости студентов

Ведомость учета успеваемости группы студентов представляет собой список записей по успеваемости студентов. Каждому студенту группы по каждой дисциплине курса сопоставлена одна запись в соответствующем журнале успеваемости. Количество полей определяется количеством назначенных преподавателем заданий. В полях “Пометка о выполнении” фиксируется статус выполнения задания студентом.

СПИСОК ДОЛЖНИКОВ		
Дата составления		
Преподаватель		
Дисциплина		
Филиал		Специальность
Курс		Группа
Список студентов		
ФИО студента	Перечень дисциплин	
Статистика		
всего студентов в группе	должников в группе	процент должников

Рисунок 6.16– Форма отчета "Список должников"

СОСТОЯНИЕ ВЫПОЛНЕНИЯ ЗАДАНИЙ					
				Дата составления	
Преподаватель					
Дисциплина					
Филиал		Специальность			
Курс		Группа			
Список студентов					
ФИО студента	Сдано заданий на проверку		Зачтено заданий		
	шт.	%	шт.	%	
Статистика					
всего студентов в группе		должников в группе		процент должников	

Рисунок 6.17 – Форма отчета, отражающего состояние выполнения заданий

Список должников – документ, содержащий фамилии тех студентов группы, у которых на момент формирования отчета существуют задолженности по каким-либо предметам. При этом напротив каждой фамилии приводится список дисциплин с задолженностями. В разделе “Статистика” документа отражается процент студентов-должников от общей численности группы. Документ формируется на основе ведомости учета успеваемости.

Отчет по состоянию выполнения заданий формируется на основе ведомости учета успеваемости студентов и содержит данные о количестве выполненных студентом заданий и количестве заданий, отправленных им на проверку. В заключительном разделе документа приводятся статистические данные по количеству студентов группы, имеющих задолженности.

Входная информация

Входная информация для проектируемой системы представлена в таблице 6.6.

Таблица 6.4 – Список входных документов

№	Идентификатор	Срок и частота поступления	Источник
1	Список студентов	начало учебного года (1 раз в год)	учебная часть
2	Список преподавателей	начало учебного года	учебная часть

		(1 раз в год)	
3	Учебный план	начало учебного года (1 раз в год)	учебная часть
4	Задания для оценки знаний студентов	согласно учебному плану	преподаватель
5	Выполненные задания, присланные студентами	по мере выполнения	студент
6	Письма с вопросом	по мере возникновения	Студент

Состав и содержание работ

Разработка “Offline-системы учета и контроля заданий в дистанционном образовании” производится в период с декабря 2007 года по июнь 2008 года.

Предполагаемая среда разработки: CodeGear Delphi 2007 for Win32.

Предполагаемый язык программирования: Object Pascal.

Предполагаемая СУБД: Microsoft SQL Server 2000.

Детальный план проведения работ представлен в виде диаграммы Ганта (Gantt chart) (рисунок 6.18).



Рисунок 18 – План-график проекта

Порядок сдачи/приемки

После завершения разработки заказчику предоставляются распечатка всех документов и электронный носитель информации. Электронная копия включает в себя следующие элементы:

- полный набор текстовых материалов (руководство пользователя, руководство программиста), представляется в электронном и в бумажном видах;
- программное средство, с полным набором программных модулей для установки и запуска продукта, с полным набором базы данных, где должна быть либо реальная информация, либо данные контрольного примера.

Электронный носитель (диск) должен быть подписан автором. Если носителей несколько, то они должны быть пронумерованы.

Проектирование информационного обеспечения

Выбор методологии проектирования

Разработка программного средства “Offline-система учета и контроля выполнения заданий в дистанционном образовании” производится в рамках процесса ICONIX, который представляет собой нечто среднее между рациональным унифицированным процессом (Rational Unified Process – RUP) и методологией программирования eXtreme (XP) /62/.

В процессе ICONIX применяется унифицированный язык моделирования UML (Unified Modeling Language), причем выявляется минимальное подмножество UML, достаточное для работы над программным проектом.

На рисунке 19 представлена обобщенная модель процесса ICONIX: сверху изображена динамическая модель, описывающая поведение, а снизу – статическая, описывающая структуру.

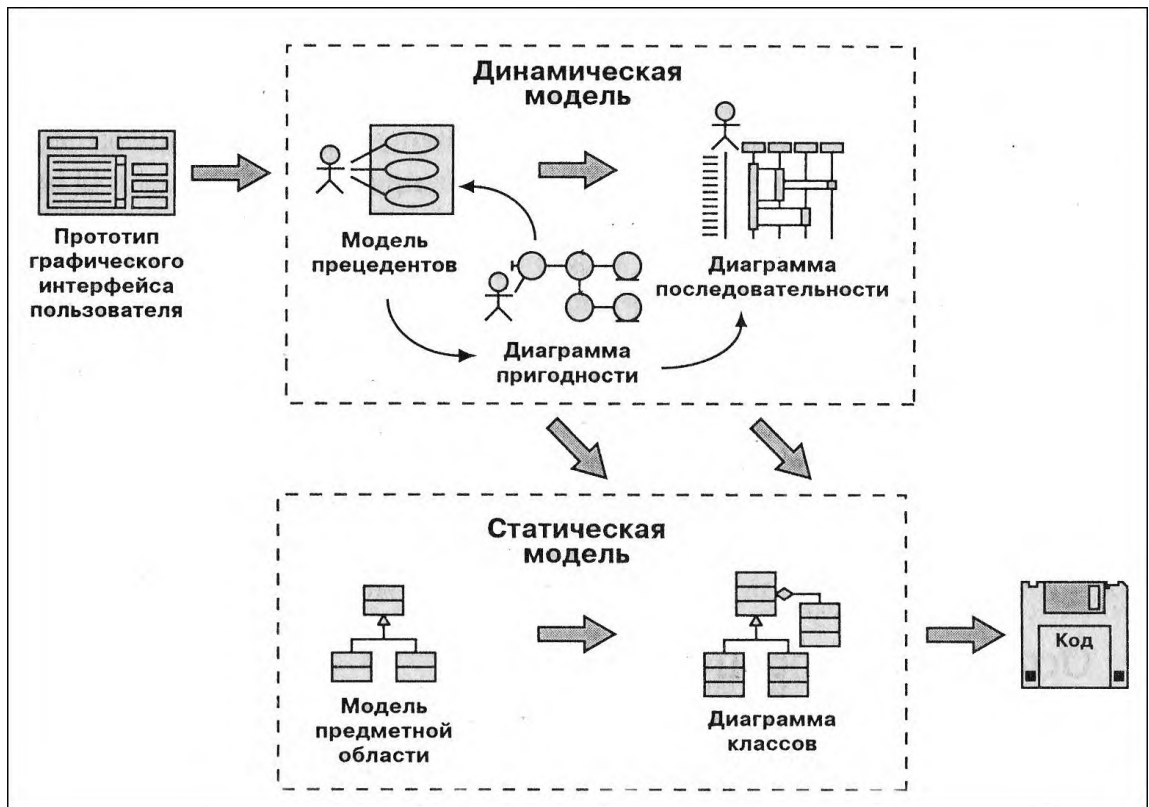


Рисунок 19 – Схема процесса ICONIX

Моделирование предметной области

При построении модели предметной области необходимо выявить основные объекты и отношения между ними. Полученная в результате диаграмма бизнес-объектов описывает статические аспекты разрабатываемой системы.

На рисунке 20 изображена модель предметной области, представленная в виде диаграммы классов уровня анализа.

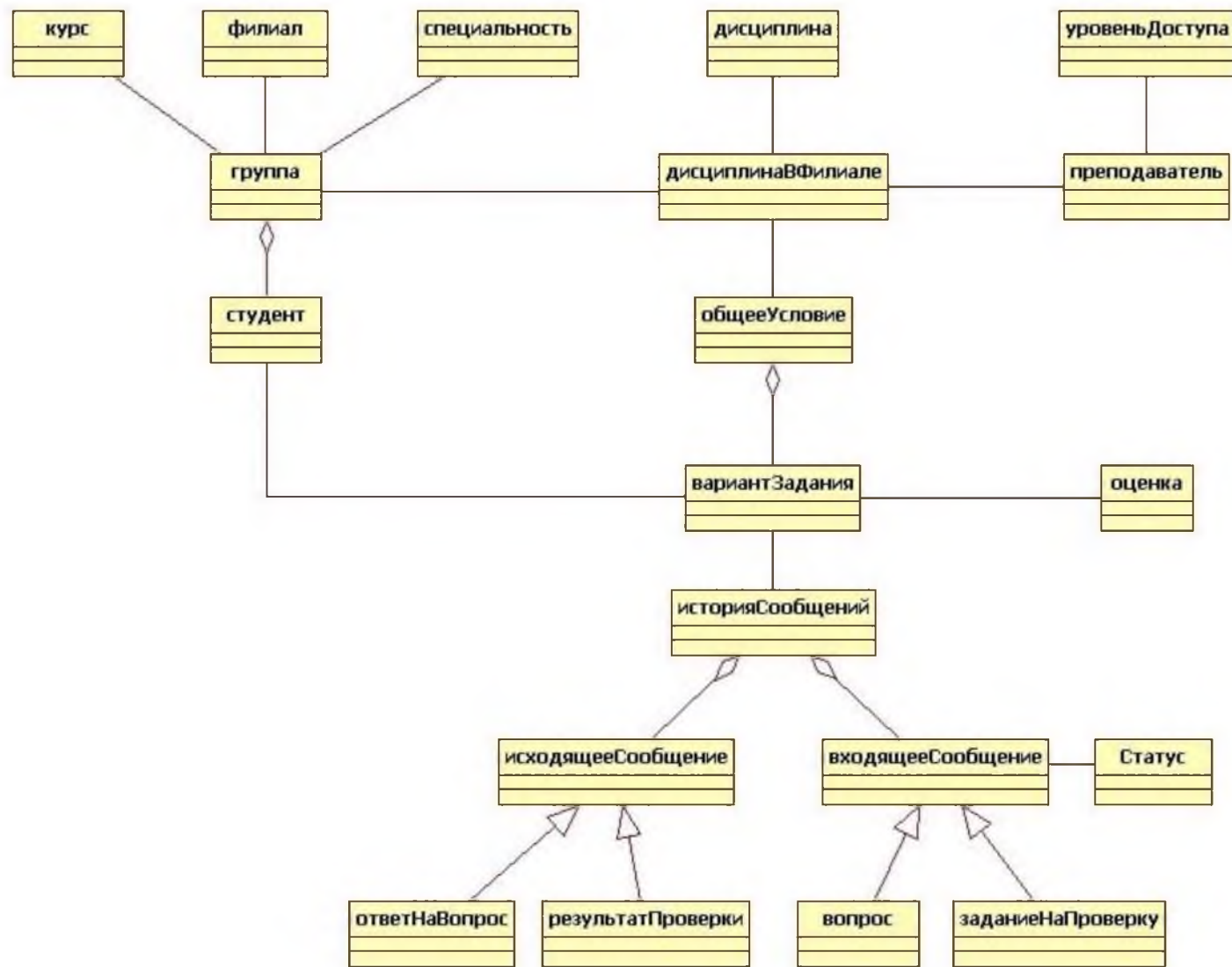


Рисунок 20 – Модель предметной области

Диаграмма предметной области представляет собой словарь терминов, которыми разработчики могут пользоваться на более поздних этапах. Кроме того данная модель является основой для разработки диаграммы классов уровня проектирования.

В процессе работы над проектом модель бизнес-объектов будет уточняться, в частности, анализ пригодности даст возможность дополнить модель атрибутами и позволит выявить ранее не включенные объекты, необходимые для каждого из прецедентов.

Прототип пользовательского интерфейса

На начальном этапе разработки были проведены работы по созданию приблизительного пользовательского интерфейса. Ниже представлены формы-прототипы будущей системы.

“Offline-система учета и контроля выполнения заданий в дистанционном образовании” имеет четыре справочника – “Филиалы”, “Специальности”, “Группы”, “Дисциплины”. Рисунки 21 и 22 иллюстрируют возможные операции со справочными данными – добавление, удаление, редактирование, фильтрацию.

Каждому пользователю должен быть определен уровень доступа к системе. Функция разграничения прав доступа пользователей также возлагается на администратора.

Пользователь, наделенный правами администратора, имеет доступ к разделу “Пользователи”. Посредством данного интерфейса обеспечивается возможность создания и изменения учетных записей пользователей системы.

Каждый пользователь с правами доступа преподавателя может работать через систему только со студентами, у которых он преподает. Распределение преподавателей по группам в филиалах, осуществляется администратором с помощью соответствующих форм³¹.

³¹ В тексте пособия не приводится 15 разработанных и описанных пользовательских интерфейсов.

Off-Line консультации в дистанционном образовании - Администратор: справочник «Группы»

Приложение: Данные

Добавить Удалить Редактировать Очистить фильтр

Пользователь: ДМИТРИЕВ АЛЕКСЕЙ НИКОЛАЕВИЧ (администратор)

Группы

редактирование справочных таблиц

Номер группы	Курс	Шифр	Специальность	Фильтр по специальности	Филиал
Г01	1	ГМУ	Государственное и муниципальное управление		Новый Уренгой
Г01	1	ГМУ	Государственное и муниципальное управление		Муравленко
П01	1	ПИвЭ	Прикладная информатика в экономике		Муравленко
Г01	1	ГМУ	Государственное и муниципальное управление		Урай
П01	1	ПИвЭ	Прикладная информатика в экономике		Урай
Г01	1	ГМУ	Государственное и муниципальное управление		Нижневартовск
Г02	2	ГМУ	Государственное и муниципальное управление		Муравленко
Ф02	2	Фик	Финансы и кредит		Урай
Г02	2	ГМУ	Государственное и муниципальное управление		Нижневартовск
П02	2	ПИвЭ	Прикладная информатика в экономике		Нижневартовск
Ф02_1	2	Фик	Финансы и кредит		Урай
Ф02	2	Фик	Финансы и кредит		Нижневартовск
Г02	2	ГМУ	Государственное и муниципальное управление		Урай
М02	2	МЭ	Мировая экономика		Нижневартовск
П02	2	ПИвЭ	Прикладная информатика в экономике		Новый Уренгой
П02	2	ПИвЭ	Прикладная информатика в экономике		Муравленко
Ф02	2	Фик	Финансы и кредит		Муравленко
П02	2	ПИвЭ	Прикладная информатика в экономике		Ишим
Ф02_2	2	Фик	Финансы и кредит		Урай
Г02_1	2	ГМУ	Государственное и муниципальное управление		Урай
М02	2	МЭ	Мировая экономика		Ишим
М02	2	МЭ	Мировая экономика		Урай
Ф03	3	Фик	Финансы и кредит		Муравленко
М03	3	МЭ	Мировая экономика		Нижневартовск
П03	3	ПИвЭ	Прикладная информатика в экономике		Новый Уренгой
Г03	3	ГМУ	Государственное и муниципальное управление		Нижневартовск
М03	3	МЭ	Мировая экономика		Муравленко
Ф03	3	Фик	Финансы и кредит		Ишим

Рисунок 21 – Форма «Режим администратора: справочник Группы»

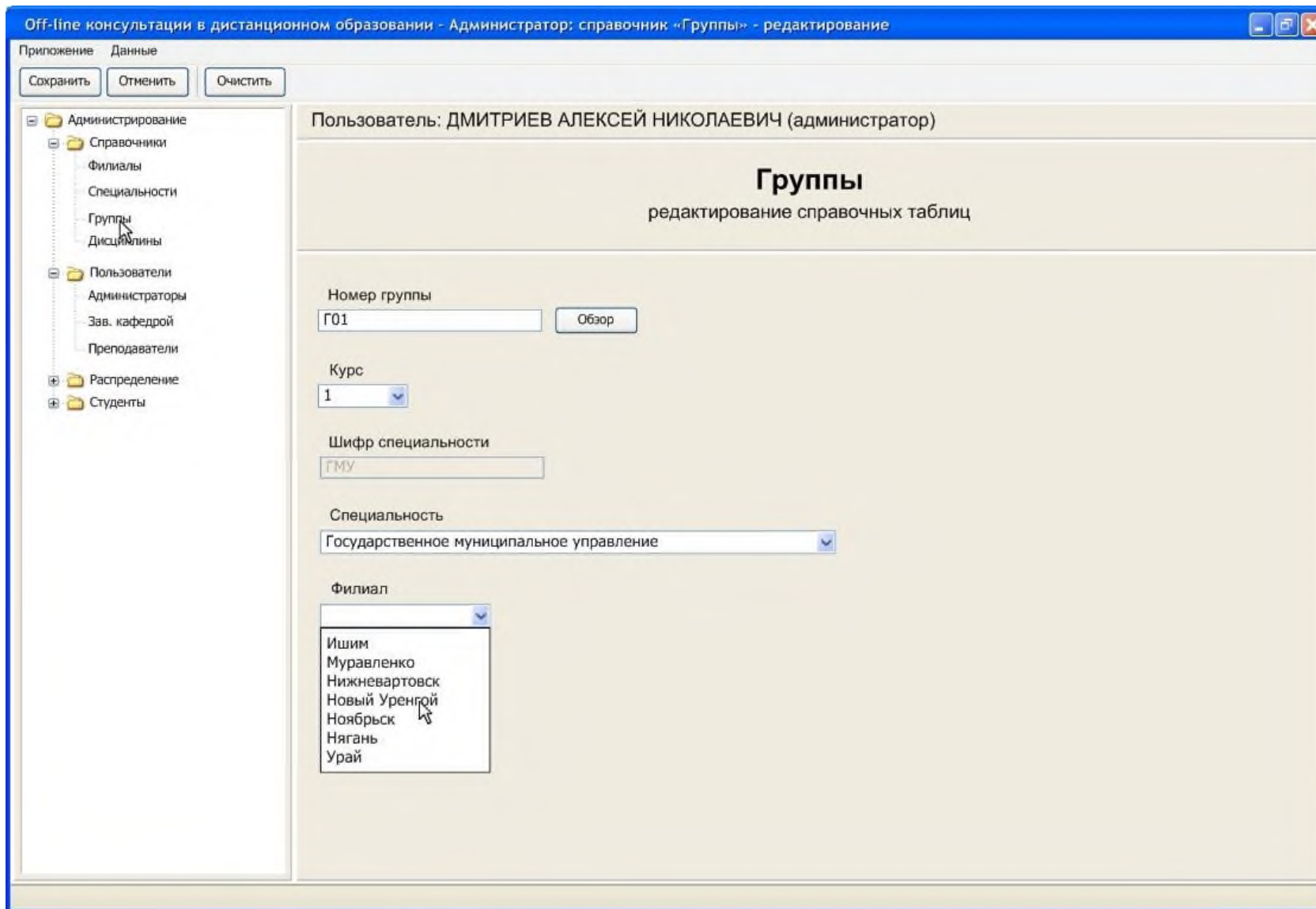


Рисунок 22 – Форма “Режим администратора: справочник Группы – редактирование”

Интерфейс заведующего кафедрой предоставляет возможность просмотра журналов успеваемости всех преподавателей (рисунок 23). Подготовка сводных отчетов производится вызовом команды “Отчеты” из главного меню.

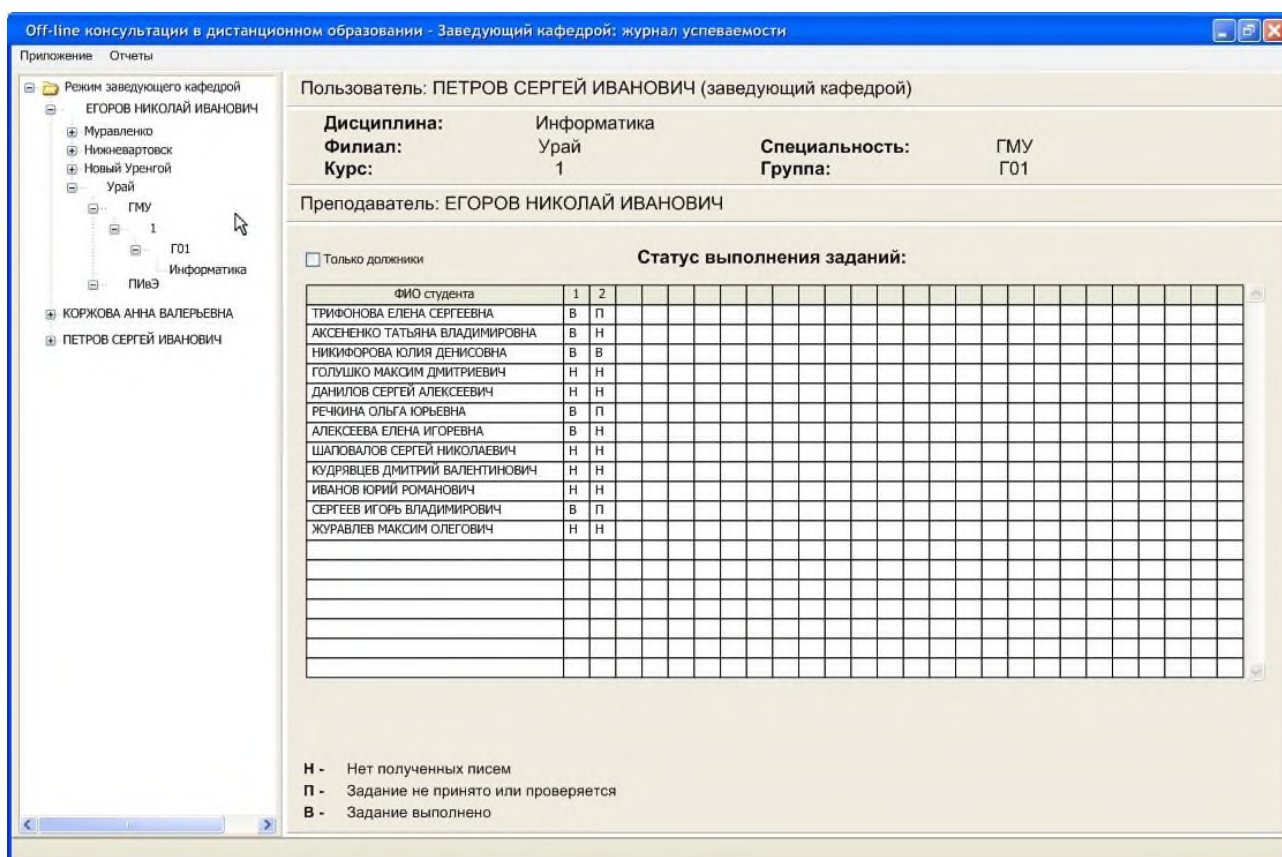


Рисунок 23 – Форма “Режим заведующего кафедрой: журнал успеваемости”

На прототипах пользовательского интерфейса была отработана технология взаимодействия преподавателя со студентами.

Моделирование прецедентов

Диаграмма вариантов использования – это исходное концептуальное представление системы в процессе ее проектирования и разработки /64/. Создание диаграммы вариантов использования имеет следующие цели:

- определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;

- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

На рисунке 24 отражены прецеденты, определенные с помощью прототипа пользовательского интерфейса. Изображенный пакет включает требования к операциям с заданиями.



Рисунок 24 – Диаграмма вариантов использования

Полная модель прецедентов, а также ее спецификации, описывающие действия пользователя и реакцию системы на эти действия, приведены в приложении³².

Создание моделей вариантов использования позволило формализовать основные требования к проектируемой системе и специфицировать ее поведение.

Анализ пригодности

В процессе ICONIX анализ пригодности выполняет несколько задач /62/:

- тексты прецедентов уточняются, проверяется их полнота и правильность;
- уточняется статическая модель, выявляются новые объекты.

Ниже приведены диаграммы пригодности, соответствующие прецедентам “Сформировать задание”, “Загрузить задание в систему”, “Назначить задание студентам” (рисунки 25, 26, 27).

³² В учебном пособии не приводится

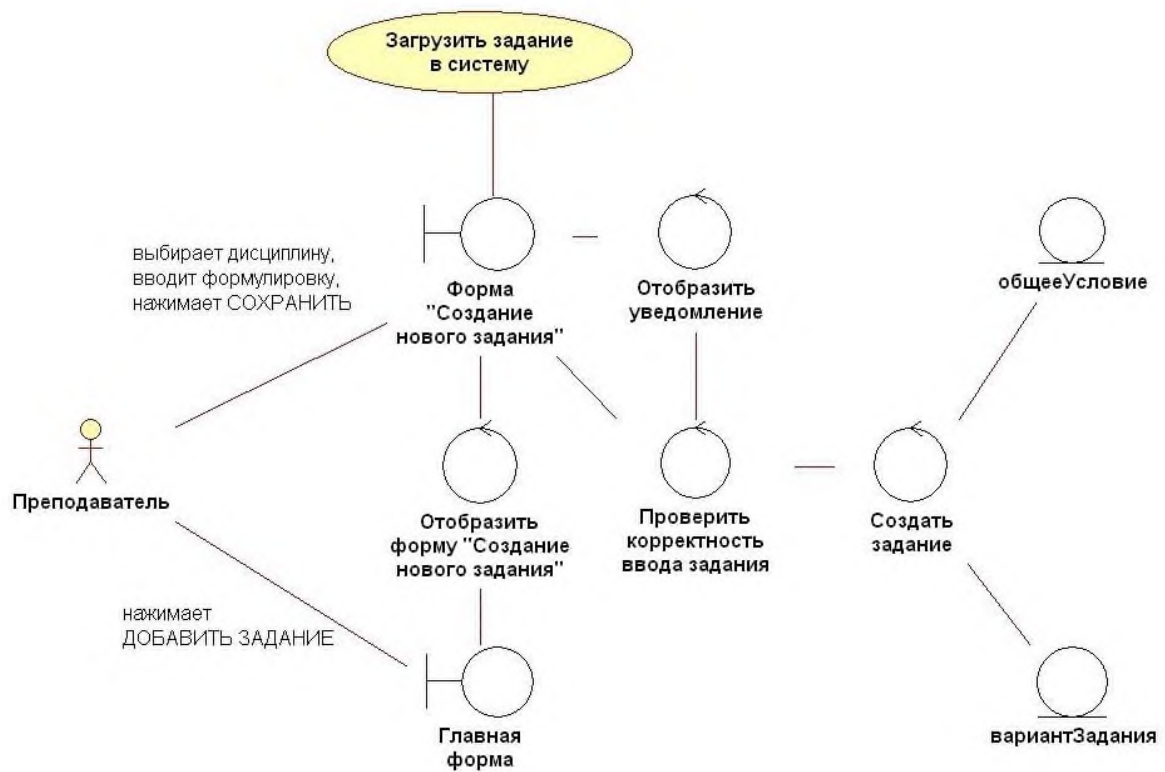


Рисунок 25 – Диаграмма пригодности на прецедент “Сформировать задание”

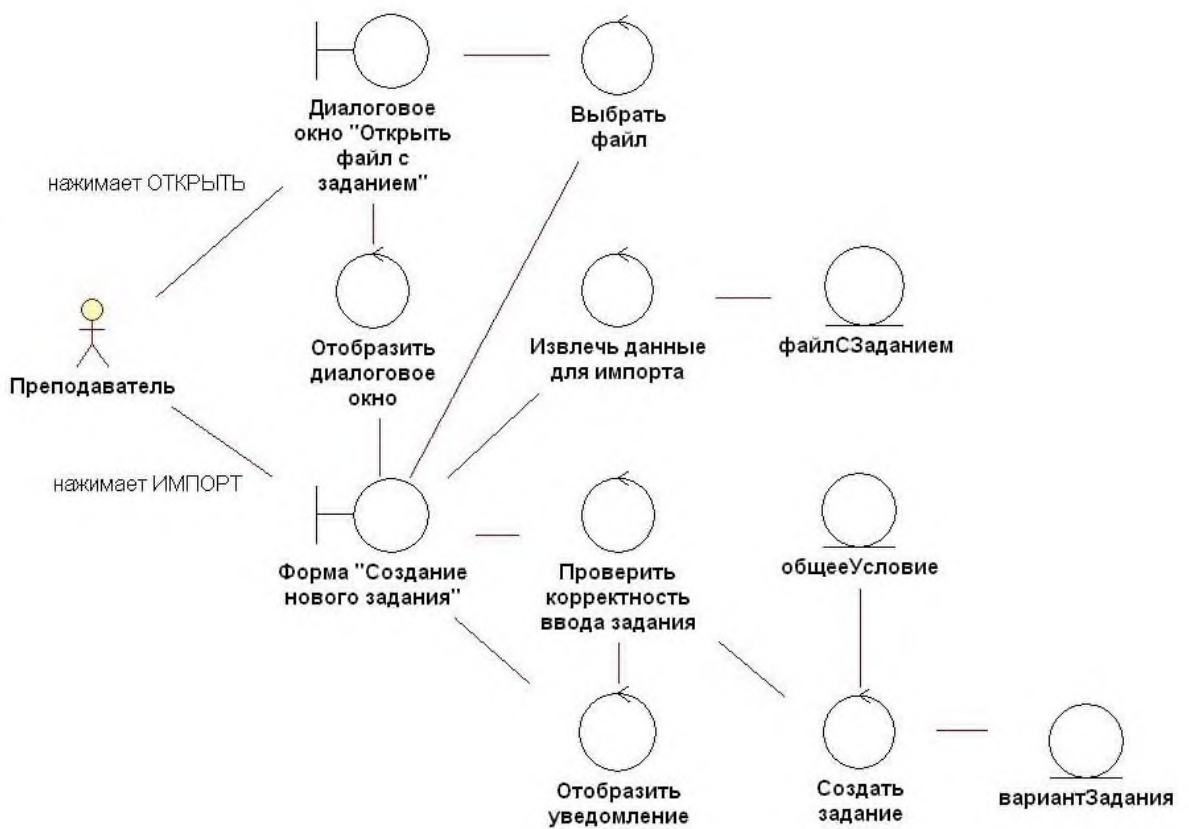


Рисунок 26 – Диаграмма пригодности на прецедент “Загрузить задание в систему”



Рисунок 27 – Диаграмма пригодности на прецедент “Назначить задание студентам”

В результате анализа пригодности модель предметной области была дополнена новыми объектами. Кроме того, были уточнены атрибуты классов, проверены на полноту и правильность тексты спецификаций прецедентов.

Полный набор диаграмм пригодности приведен в приложении³³.

Моделирование динамического представления системы

Диаграммы последовательностей сфокусированы на временной последовательности сообщений, что дает возможность показать взаимодействие объектов во времени.

Ниже представлены диаграммы последовательностей для прецедентов “Сформировать задание”, “Загрузить задание в систему”, “Назначить задание студентам” (рисунки 28, 29, 30).

³³ В учебно пособии не приводится

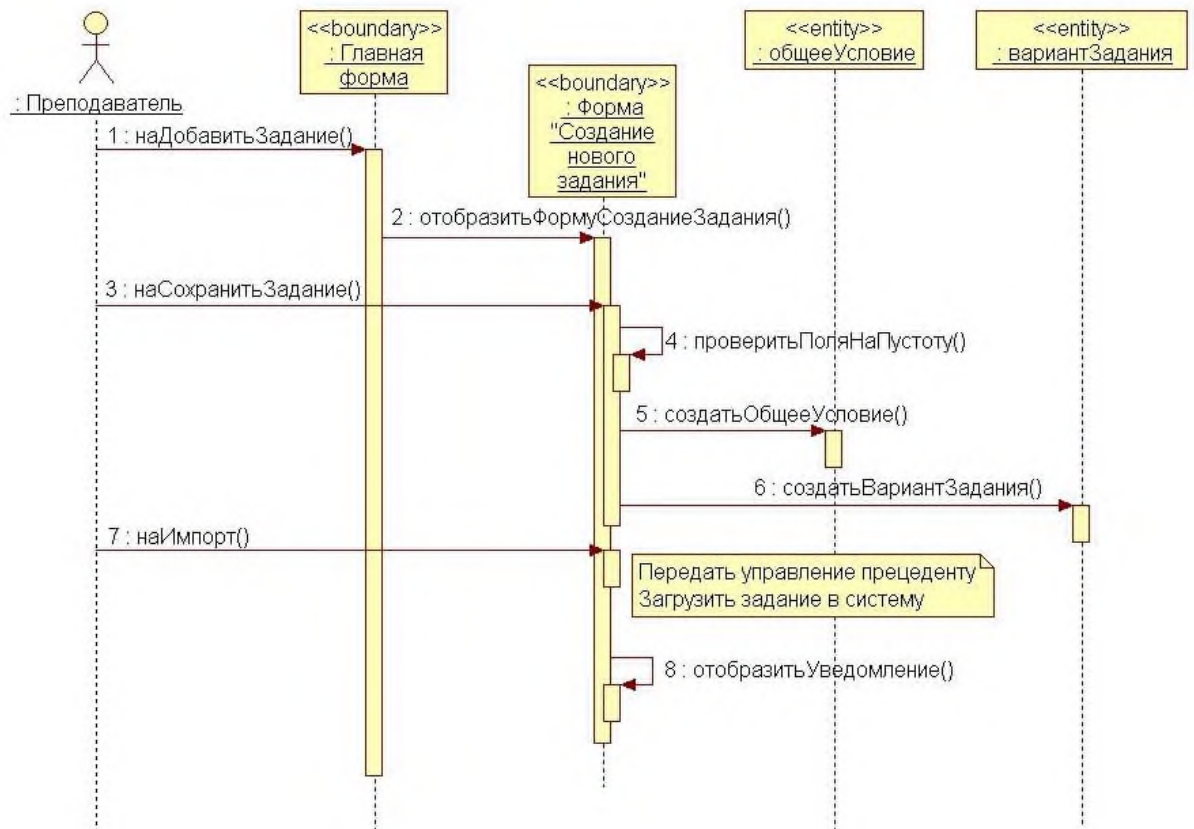


Рисунок 28 – Диаграмма последовательностей для прецедента “Сформировать задание”

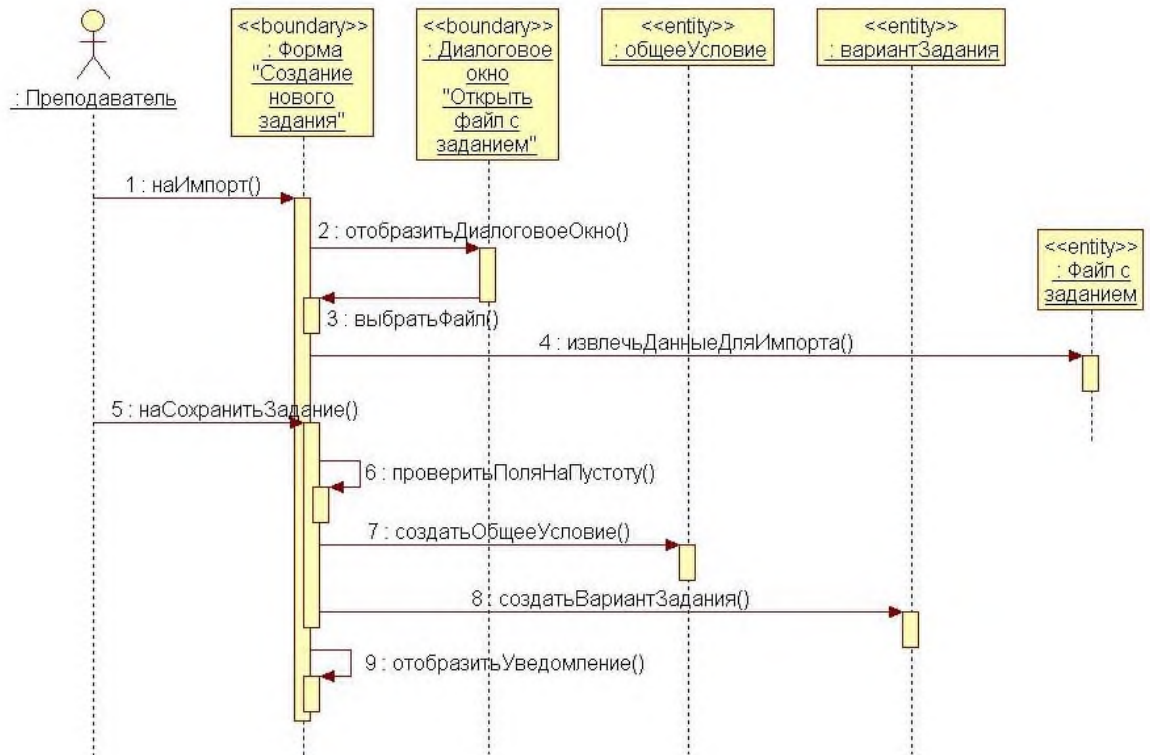


Рисунок 29 – Диаграмма последовательностей для прецедента “Загрузить задание в систему”

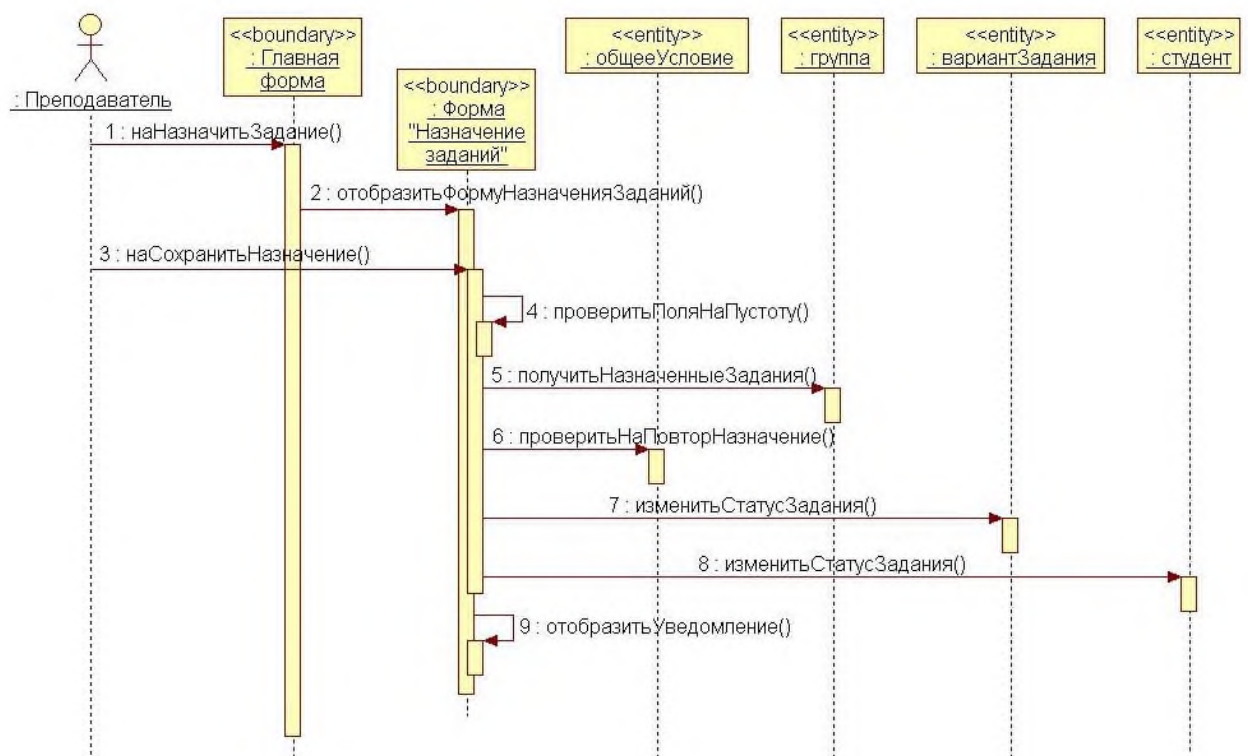


Рисунок 30 – Диаграмма последовательностей для прецедента “Назначить задание студентам”

Применение диаграмм последовательностей позволило распределить поведение между граничными, сущностными и управляющими объектами; детально показать взаимодействия между объектами, участвующими в каждом прецеденте; распределить операции по классам. Полный набор диаграмм последовательностей содержится в приложении³⁴.

Диаграммы классов

Диаграммы последовательностей помогли распределить поведение между классами, в результате чего была получена модель классов уровня проектирования (рисунок 31). На основе полученной диаграммы классов была сформирована статическая модель проектируемой системы, представленная в виде диаграммы классов уровня реализации (рисунок 32).

³⁴ В учебном пособии не приводится

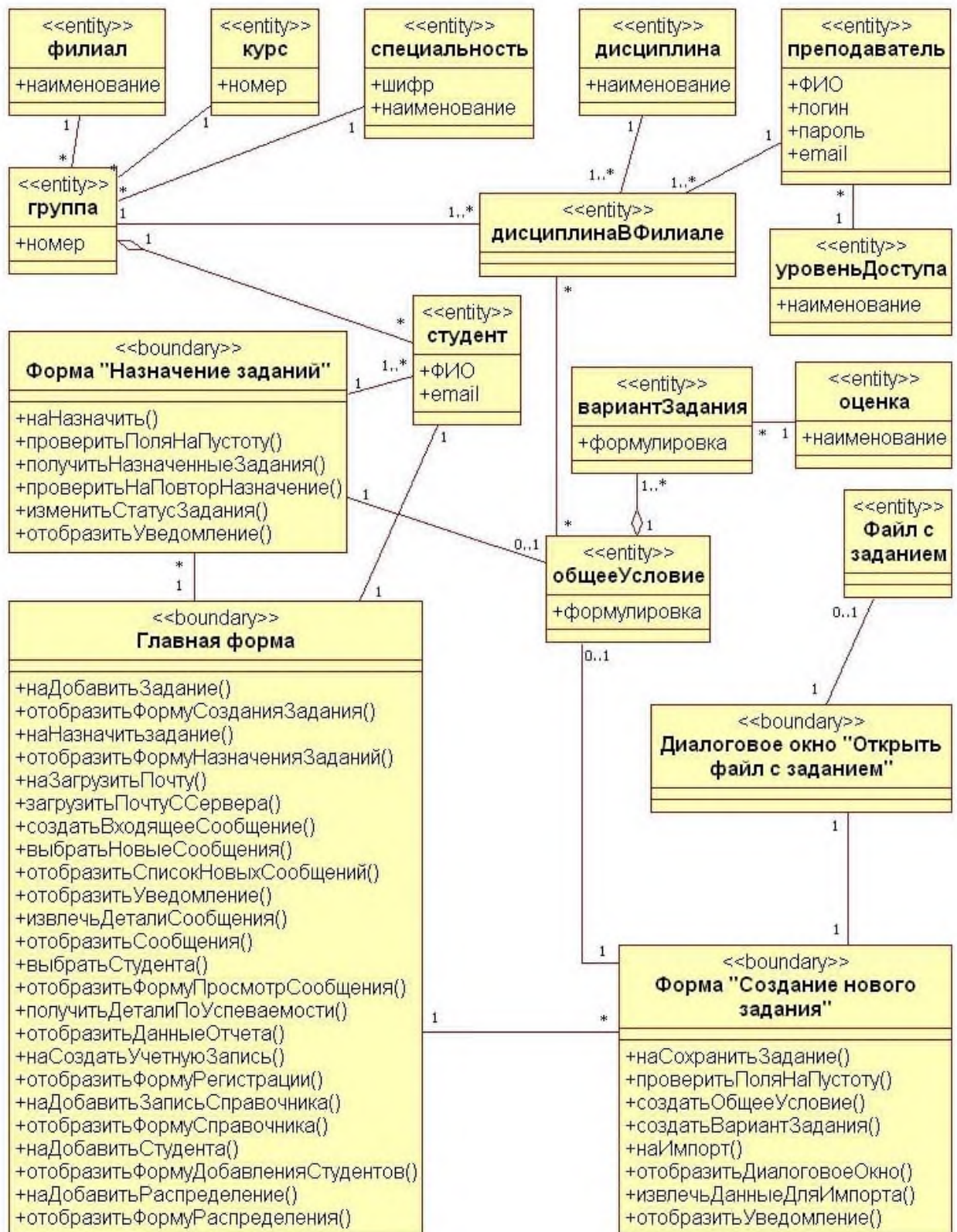


Рисунок 31 – Диаграмма классов уровня проектирования

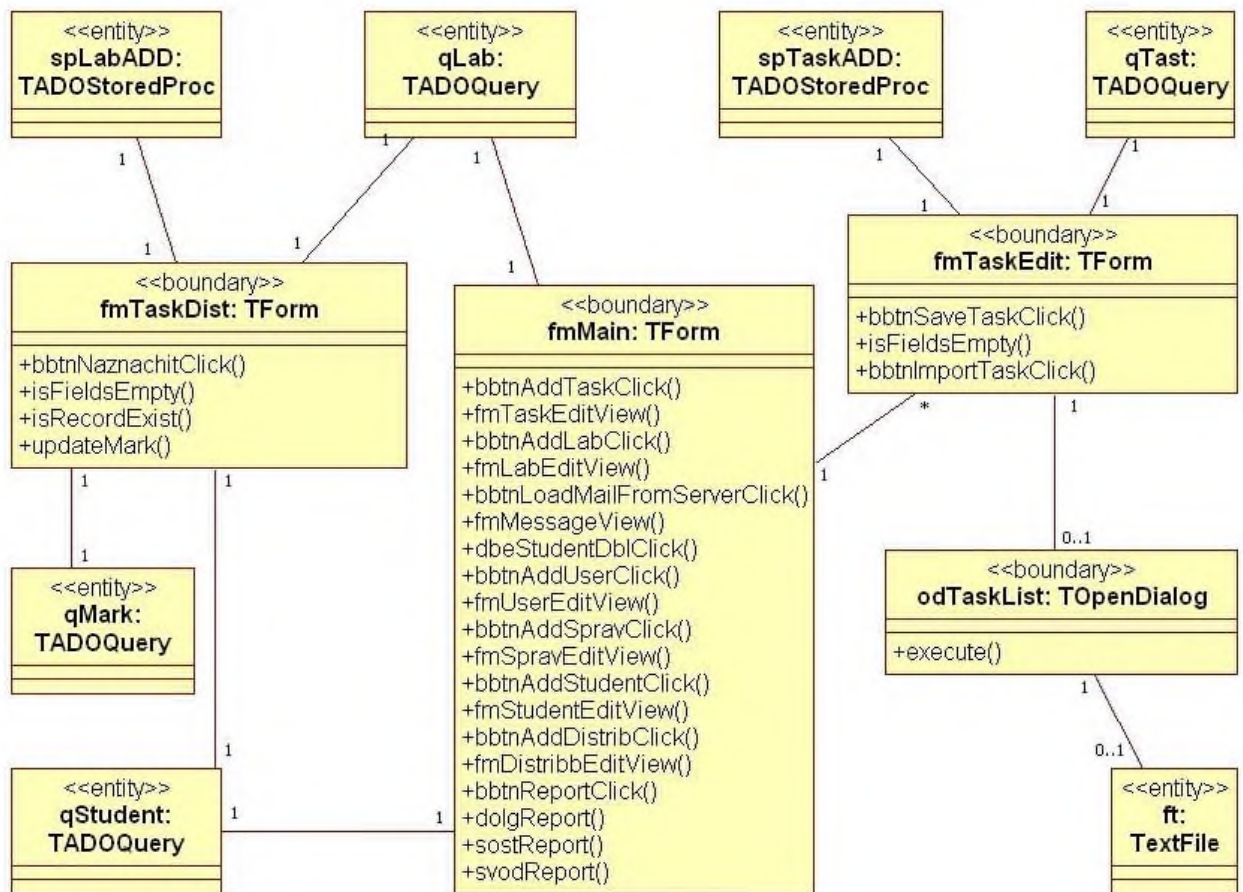


Рисунок 32 – Диаграмма классов уровня реализации

Разработанная диаграмма классов дает возможность легко перейти к программному коду – выявлены необходимые классы, их атрибуты и операции.

Полная модель классов представлена в приложении³⁵.

Схема данных

На основе модели предметной области и диаграмм пригодности, была разработана модель данных, представленная на рисунке 33.

³⁵ В учебном пособии не приводится.

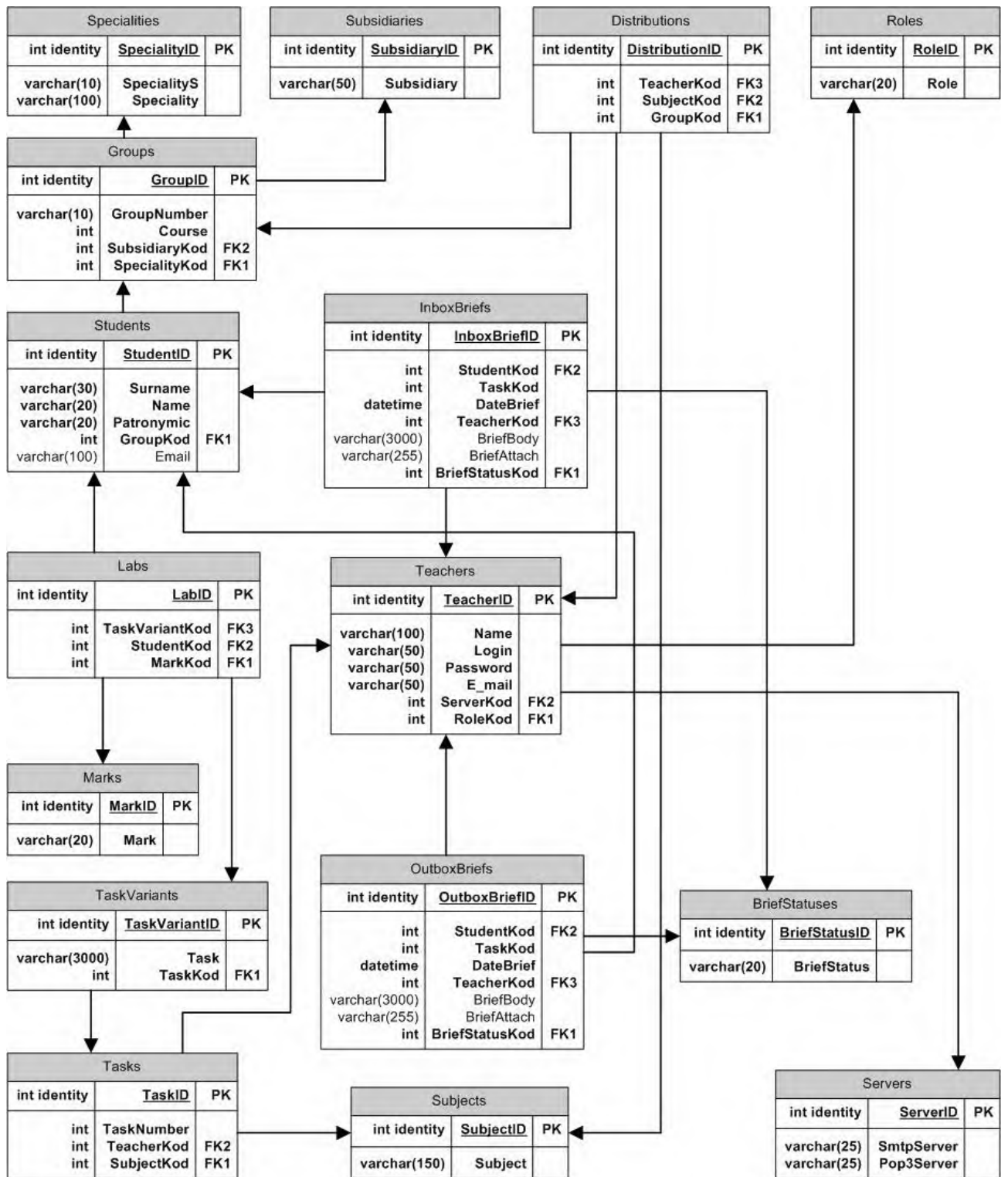


Рисунок 33 – Физическая схема данных

Проектирование программного обеспечения

Описание программного обеспечения

“Offline-система учета и контроля выполнения заданий в дистанционном образовании” представляет собой прикладное программное обеспечение, разработанное для кафедры ИС Института математики и компьютерных наук университета. Система предназначена для решения следующих задач:

- обеспечение взаимодействие преподавателя со студентами, обучающимися в филиалах, в течение всего семестра;
- снижение затрат времени преподавателя на сортировку и распределение почты;
- снижение затрат времени преподавателя на подготовку отчетов по задолженностям студентов, по успеваемости студентов;
- снижение затрат времени заведующего кафедрой на подготовку сводных отчетов.

Для функционирования программы необходима установка приложений Microsoft SQL Server 2000 – для полноценной работы с БД и Microsoft Excel – для подготовки отчетов.

Требования к прикладному программному обеспечению

Выше перечисленные задачи вызывают необходимость реализации ряда требований к проектированию прикладного программного обеспечения.

Система должна соответствовать следующим требованиям:

- обеспечить связь с почтовым сервером для получения и отправки сообщений;
- обеспечить распределение входящих данных в БД для удобства дальнейшей обработки информации;
- обеспечить подготовку необходимых отчетов.

Структура и состав программного обеспечения

Определить архитектуру разрабатываемой системы, установив зависимости между компонентами, позволяет диаграмма компонентов (Component Diagram). На рисунке 34 представлена модель размещения, совмещенная с диаграммой компонентов проектируемой системы.

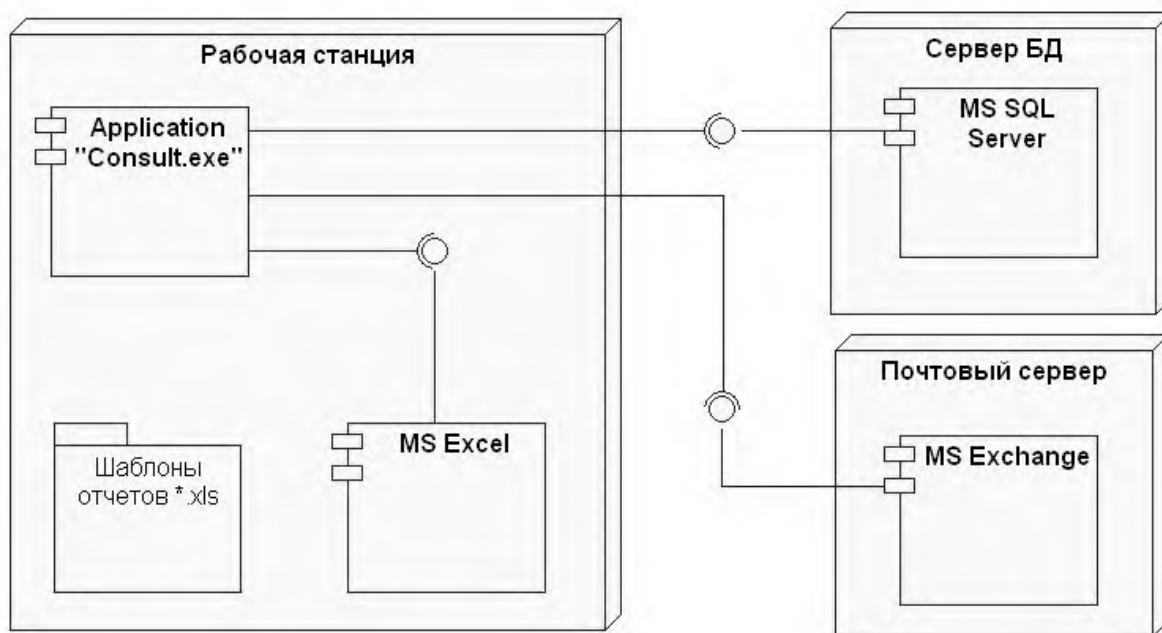


Рисунок 34 – Диаграмма размещения

Таким образом, “Offline-система учета и контроля выполнения заданий в дистанционном образовании” включает следующие структурные единицы:

- файл приложения, включающий в себя реализацию основных функциональных требований;
- файл БД, реализующий разработанную схему данных и обеспечивающий хранение входящей и справочной информации;
- пакет шаблонов MS Excel, используемый при подготовке отчетов.

Средства разработки программного обеспечения

Для реализации спроектированной схемы базы данных была выбрана СУБД Microsoft SQL Server 2000. В первую очередь, данное решение обосновано тем, что в предполагаемом месте внедрения проекта уже используется данная СУБД. Кроме того, MS SQL Server 2000 обладает следующими преимуществами:

- для обработки хранимых в таблицах сервера данных используется модификация стандартного языка SQL – Transact SQL. Это довольно удобный язык программирования, позволяющий легко реализовывать алгоритмы обработки данных путем написания хранимых процедур и табличных триггеров;
- утилита SQL Server Enterprise Manager имеет удобный графический интерфейс и является мощным инструментом управления БД;
- MS SQL Server, работающий только под управлением ОС Windows, использует в своей работе сервисы безопасности Windows NT;
- существует возможность реализации сетевой версии проекта.

При написании программного обеспечения использована среда CodeGear Delphi 2007 for Win32, особенностями которой считаются объектно-ориентированный подход, поддержка баз данных (в том числе, MS SQL Server), тесная интеграция с Windows-программированием и компонентная технология.

Руководство пользователя

“Offline система учета и контроля выполнения заданий в дистанционном образовании” имеет стандартный пользовательский интерфейс. В верхней части основной формы находится главное меню (рисунок 35).

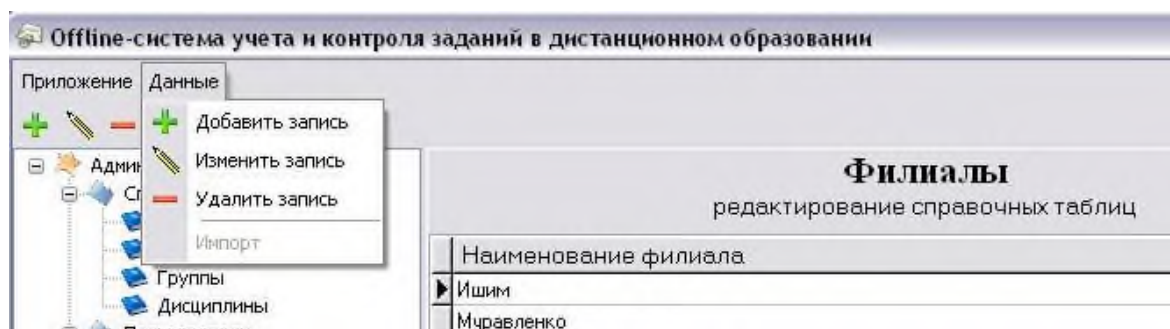


Рисунок 35 – Главное меню приложения

Пункт меню “Приложение” дает доступ к опциям смены пользователя и выхода из системы. Пункт меню “Данные” (режим “Администратор”) содержит команды для работы со справочными таблицами: “Добавить запись”, “Изменить запись”, “Удалить запись”, “Импорт”.

Для удобства пользователей часто используемые команды меню вынесены на панель инструментов, которая расположена под главным меню приложения (рисунок 35).

В левой части основного окна приложения расположено дерево навигации, посредством которого осуществляется переход от одного раздела к другому (рисунок 36).

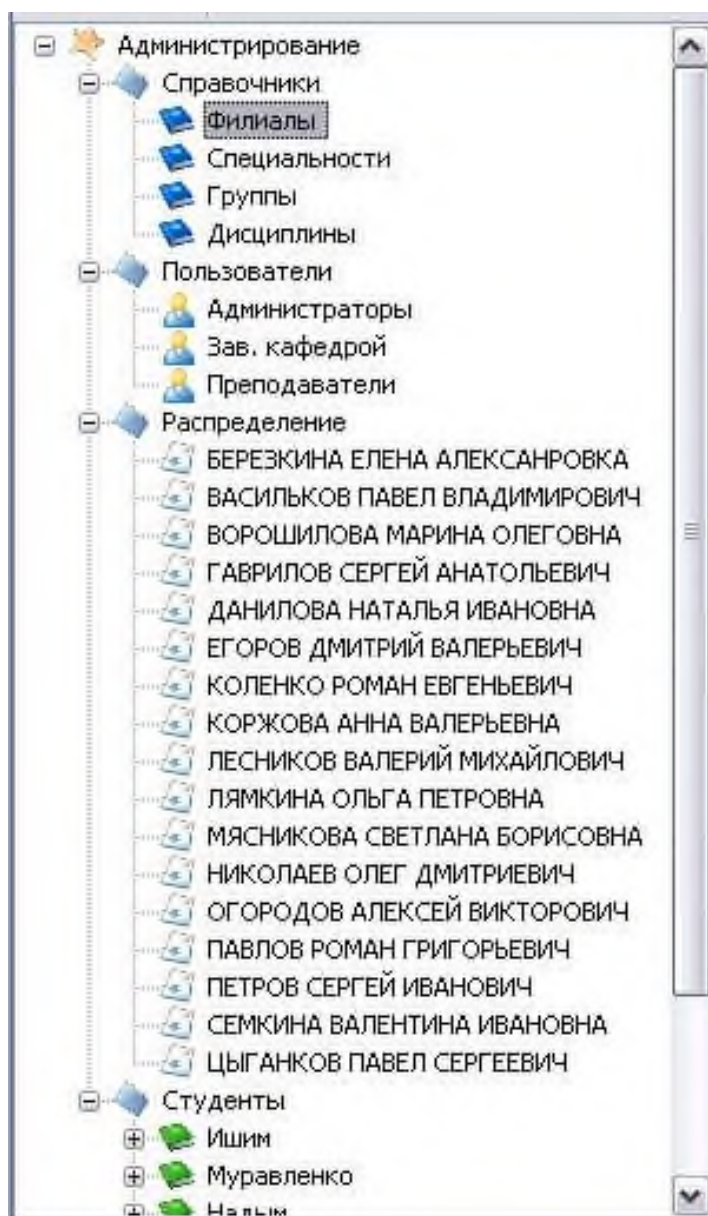


Рисунок 36 – Навигационное дерево

Оставшаяся часть окна приложения предназначена для отображения данных.

Режим «Администратор»

В данном режиме представлено 4 раздела: “Справочники”, “Пользователи”, “Нагрузка” и “Студенты”.

В разделе “Справочники” пользователь имеет возможность просматривать, редактировать и удалять записи справочных таблиц – “Филиалы”, “Специальности”, “Группы”, “Дисциплины”, – а также добавлять новые (рисунки 37,38).

№ гр.	Курс	Шифр	Специальность
361-1	3	ПМ	Прикладная математика
361-2	3	ПМ	Прикладная математика
380-И	1	ПМ	Прикладная математика
381-И	1	ГМУ	Государственное и муниципальное управление
382-И	1	ФиК	Финансы и кредит
422	1	МОАИС	Математическое обеспечение и администрирование информационных сист
432	2	МОАИС	Математическое обеспечение и администрирование информационных сист
Г377Н	2	ГМУ	Государственное и муниципальное управление
Г873	1	ГМУ	Государственное и муниципальное управление
Г874	1	ГМУ	Государственное и муниципальное управление
ИС271	1	ИСТ	Информационные системы и технологии
М364	3	МОАИС	Математическое обеспечение и администрирование информационных сист
М367	1	МОАИС	Математическое обеспечение и администрирование информационных сист
П07-1	2	ПИВЭ	Прикладная информатика в экномике
П33-1	2	ПМ	Прикладная математика
П376Н	2	ПИВЭ	Прикладная информатика в экномике
П376НУ	1	ПИВЭ	Прикладная информатика в экномике
С322	1	СЦ	Социология
С43-07	1	СЦ	Социология
С924	1	СЦ	Социология
Ф 279	1	ФиК	Финансы и кредит
Ф07-1	2	ФиК	Финансы и кредит
Ф07-2	2	ФиК	Финансы и кредит

Рисунок 37– Справочник “Группы”

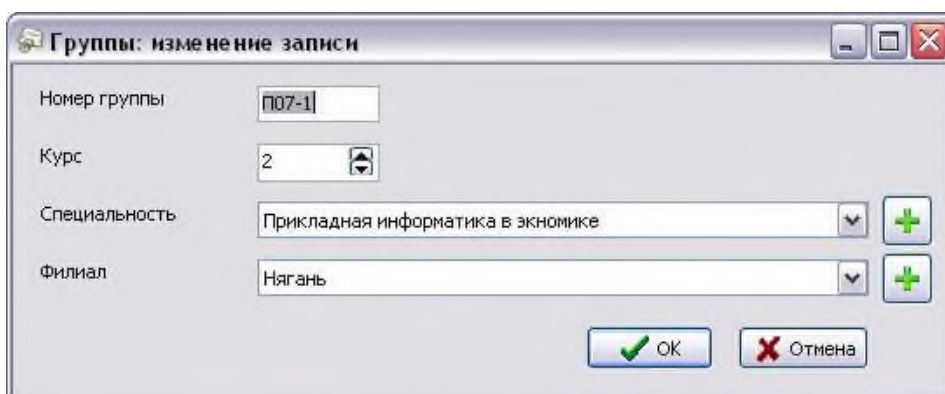


Рисунок 38 – Справочник “Группы”: просмотр и изменение записи

Раздел “Пользователи” позволяет просматривать, редактировать, добавлять/удалять пользователей системы (рисунок 39).

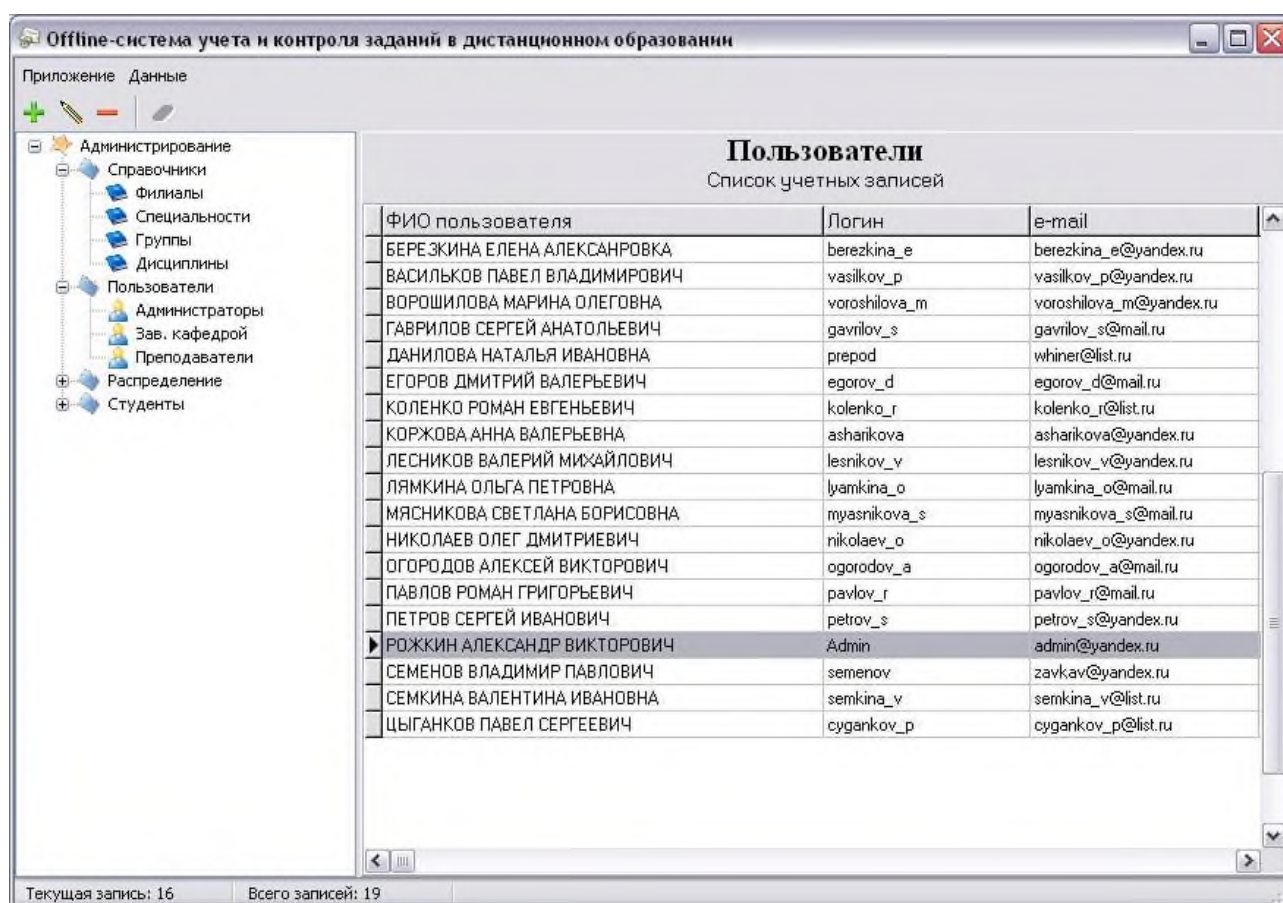


Рисунок 39 – Раздел “Пользователи”

Для отображения пользователей одной группы достаточно выделить одноименный узел в навигационном дереве.

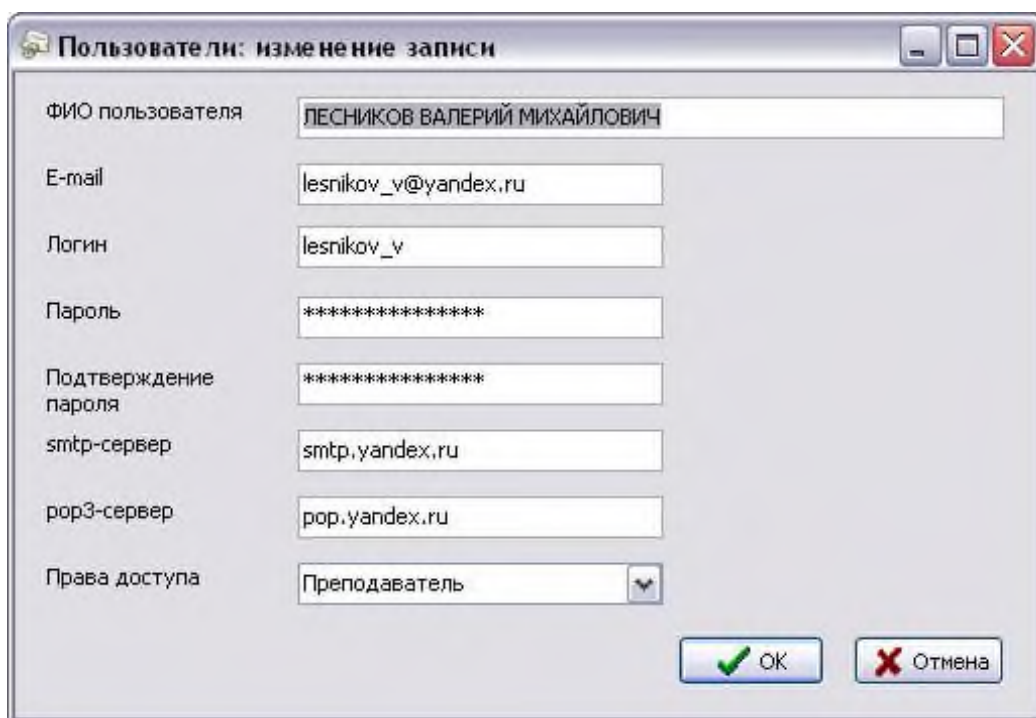


Рисунок 40 – Раздел “Пользователи”:
просмотр и изменение учетной записи пользователя

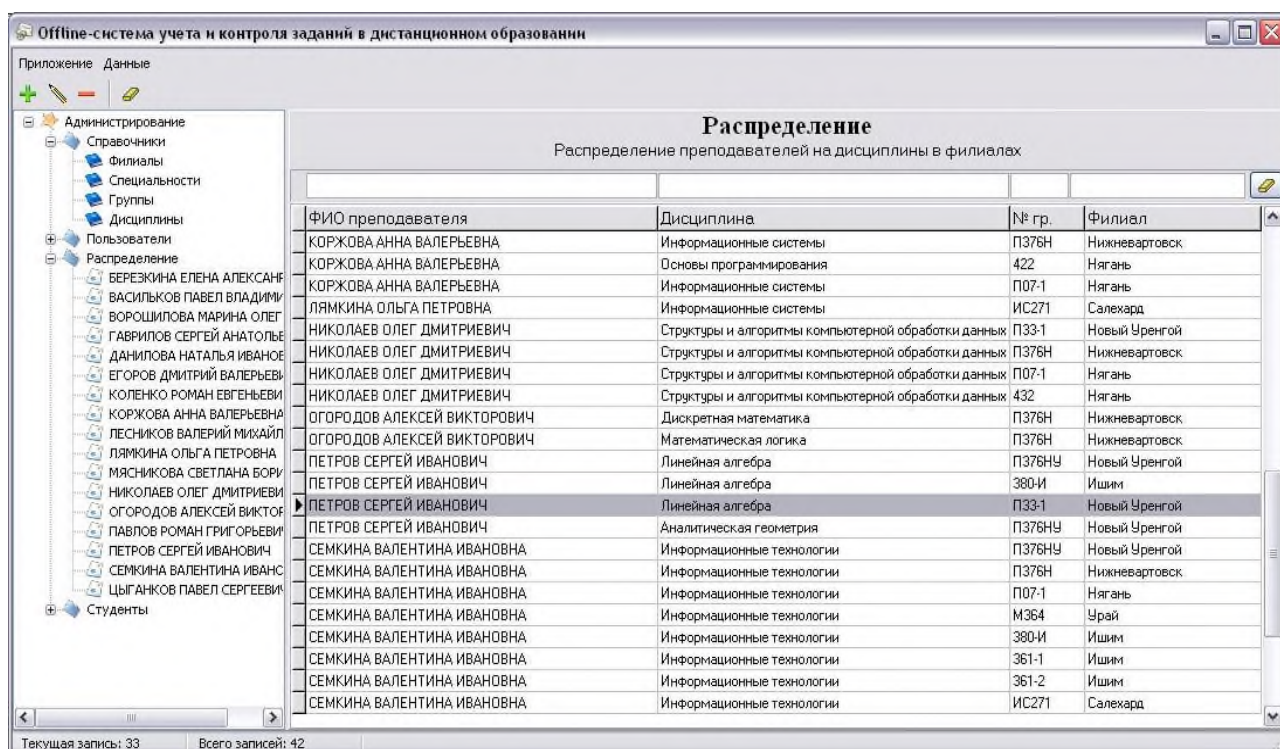


Рисунок 41– Раздел “Распределение”

В разделе “Распределение” осуществляется распределение преподавателей по филиалам (рисунок 41). Под распределением понимается назначение преподавателя

на дисциплину, читаемую для определенной группы в одном из филиалов университета.

Для просмотра списка распределений определенного преподавателя, необходимо в дереве навигации выбрать узел, соответствующий имени данного преподавателя, либо воспользоваться панелью фильтрации, расположенной над таблицей пользователей.

При редактировании или добавлении новой записи (рисунок 42) в список распределений ФИО преподавателя и наименование дисциплины выбираются из соответствующего выпадающего списка. Подстановка номера группы осуществляется выбором необходимой группы на форме, вызванной нажатием клавиши “...” (“Подбор”). На случай отсутствия в списке существующих нужной группы предусмотрена кнопка “Добавить группу”, которая вызывает форму добавления новой записи в справочник “Группы”³⁶.

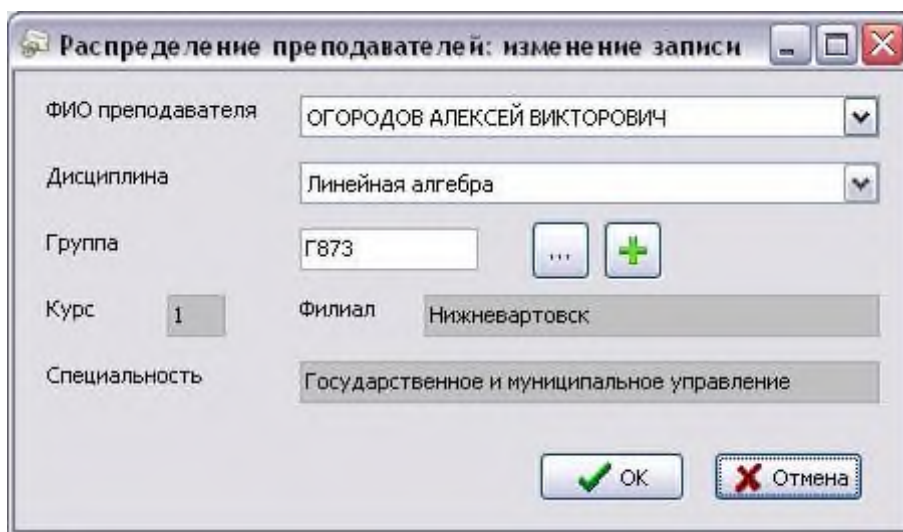


Рисунок 2 – Раздел “Распределение”: просмотр и изменение записи.

Режим «Преподаватель»

Режим “Преподаватель” предназначен непосредственно для преподавателей. При входе на главной форме отображается имя вошедшего пользователя.

В данном режиме доступно 3 раздела: “Студенты”, “Задания” и “Почта”³⁷.

³⁶ Раздел «Администратор» в учебном пособии приводится не полностью

³⁷ Раздел «Преподаватель» в учебном пособии представлен не полностью. Разделы “Студенты”, “Задания” и “Почта” не представлены в учебном пособии не представлены.

Чтобы просмотреть детализацию переписки со студентом по определенному заданию, необходимо щелкнуть по соответствующей ячейке. В результате на экране появится новая форма “Список заданий студента” (рисунок 43) со списком всех заданий, назначенных преподавателем студенту.

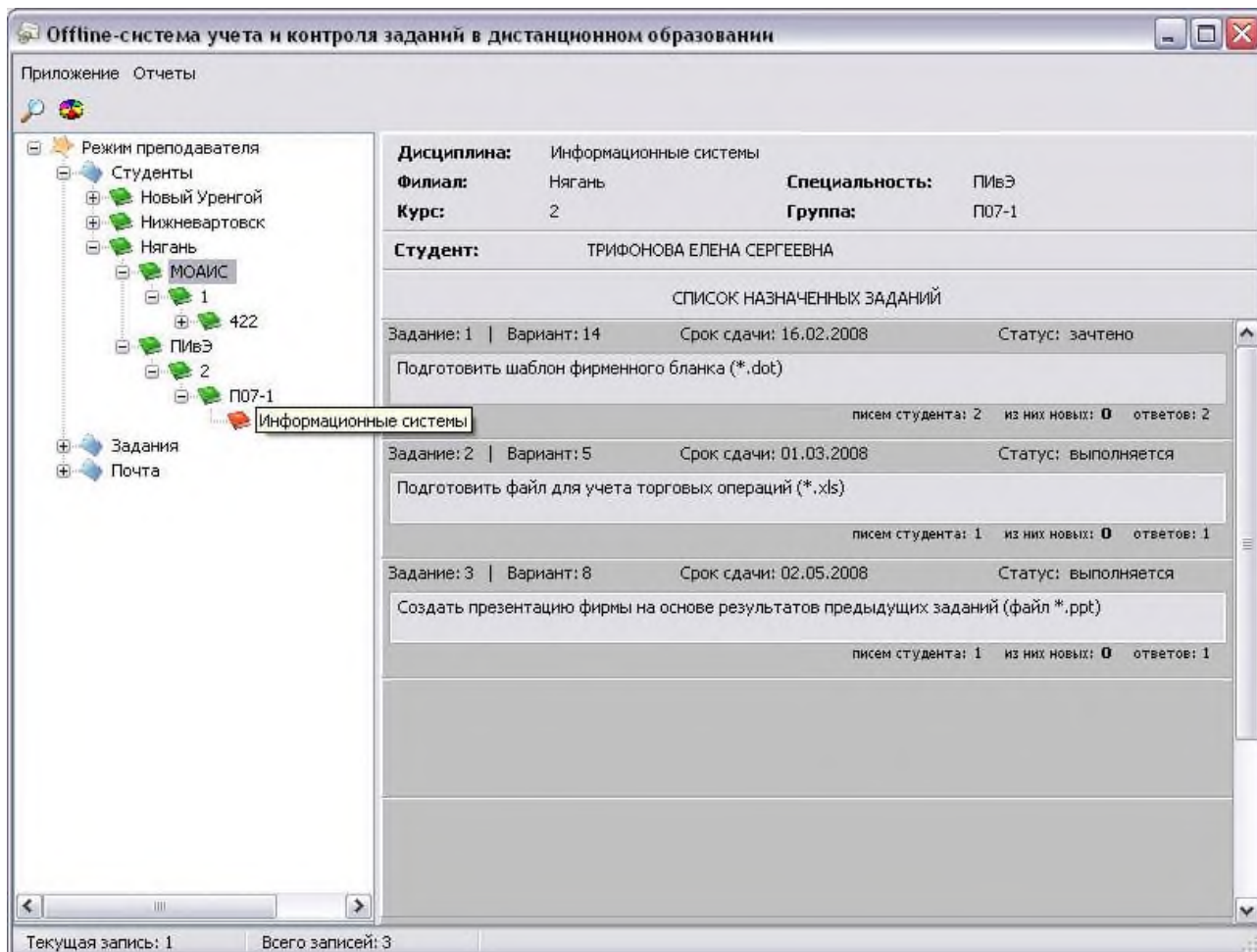


Рисунок 43 – Форма “Список назначенных заданий”

При щелчке по заданию вызывается форма просмотра истории выполнения задания. Двойным щелчком по письму осуществляется просмотр детализации сообщения. Просмотр вложенного файла осуществляется двойным щелчком по его названию. Кнопка “Ответить” помогает сформировать ответное письмо. При этом существует возможность вложения файлов. Для отправки сообщения щелкнуть “Отправить”.

При необходимости преподаватель может распечатать список должников (рисунок 44). Помимо отчета “Список должников” команды пункта меню “Отчеты” дают возможность распечатать ведомость успеваемости студентов группы и подго-

товить отчет по состоянию выполнения заданий. Подготовка отчетов осуществляется через отправку данных в приложение Microsoft Excel.

Режим «Заведующий кафедрой»³⁸

Для пользователя с правами доступа “Зав. кафедрой” навигационное дерево приобретает следующий вид: узлы, отходящие от корня дерева, представляют собой имена преподавателей, имеющих распределение по филиалам. Каждому преподавателю соответствует список дисциплин, читаемых группам студентов, представленный в виде иерархии: Филиал → Специальность → Курс → Группа → Дисциплина. Помимо отчетов, доступных пользователю в режиме “Преподаватель”, заведующий кафедрой может подготовить сводные отчеты, выбрав пункт меню Отчеты → Сводный отчет или нажав соответствующую клавишу на панели инструментов.

³⁸ Режим заведующего кафедрой в учебном пособии описан не полностью.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Положения об итоговой государственной аттестации выпускников высших учебных заведений Российской Федерации. Утверждено приказом Минобрзаования России от 25.03.03 N 1155. http://www.edu.ru/db-mo/mo/Data/d_03/1155.html
2. Государственный образовательный стандарт высшего профессионального образования специальность 351400 «Прикладная информатика (по областям)». Утвержден 14 марта 2000 г. Номер государственной регистрации:52 мжд / сп
3. Государственный образовательный стандарт высшего профессионального образования специальность по направлению подготовки 230200 - Информационные системы. Утвержден 23 декабря 2005 г. Номер государственной регистрации: 761 тех / сп
4. Большая советская энциклопедия. <http://www.rubricon.com/>
5. ГОСТ 34.201-90. Виды, комплектность и обозначение документов при создании автоматизированных систем.
6. Руководство к Своду знаний по управлению проектами. Третье издание. Американский национальный стандарт ANSI/PMI 99-001-2004
7. Кролл П, Кратчхен Ф. Rational Unified Process – это легко. Руководство по RUP. Пер. с англ. – М.: КУДИЦ-ОБРАЗ, 2004.-432с.
8. Арлоу Д., Нештадт А. UML2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2007. – 624 с.
9. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
10. А+ сертификация. Учебный курс. Официальное пособие Microsoft для самостоятельной подготовки- Издательство : Русская Редакция, 2000 – 512с. - ISBN : 5-7502-0055-8

11. Что за зверь научный руководитель?
http://www.aspirantura.spb.ru/rukvo/nauch_ruk.html
12. Положение о научном руководителе (МЭСИ)
<http://www.kfmesi.ru/rukovod.php>
13. Инструкция по подготовке, оформлению и представлению к защите дипломных проектов (работ) в высших учебных заведениях. Утверждено приказом Министерства образования Республики Беларусь от 27 июня 1997 г. № 356.
14. Мильчин А. Э. Издательский словарь-справочник. — Изд. 3-е, испр. и доп., Электронное — М.: ОЛМА-Пресс, 2006
15. Закон РФ «ОБ АВТОРСКОМ ПРАВЕ И СМЕЖНЫХ ПРАВАХ» от 9 июля 1993 года N 5351-1 (в ред. Федеральных законов от 19.07.95 N 110-ФЗ, от 20.07.2004 N 72-ФЗ).
16. Яценко Н. Е. Толковый словарь обществоведческих терминов. СПб. "Лань", 1999. 750с.
17. Системы менеджмента качества. Основные положения и словарь. ГОСТ Р ИСО 9000-2001 (Дата введения 2001-08-31) – 45с.
18. Большой юридический словарь. 3-е изд., доп. и перераб. / Под ред. проф. А. Я. Сухарева. — М.: ИНФРА-М, 2007. — VI, 858 с. — (Б-ка словарей "ИНФРА-М").
19. Буч Г., Рамбо Д., Якобсон И. Ящик UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н.- М.: ДМК Пресс, 2007.- 496с.
20. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
21. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. (Взамен ГОСТ 24.601-86, ГОСТ 24.602-86)
22. РД 50 - 34.698 – 90 Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов (Дата введения 01.01.92).

- 23.Руководство к Своду знаний по управлению проектами. Третье издание. Американский национальный стандарт ANSI/PMI 99-001-2004
- 24.ISO/TR 10006: 1997 (E). Quality Management - Guidelines to quality in project management - p. 1.
- 25.ГОСТ Р ИСО/МЭК 12207-99 "Информационная технология. Процессы жизненного цикла программных средств"
- 26.ГОСТ 7.32-2001 Отчет о научно-исследовательской работе. Структура и правила оформления (Дата введения 2002-07-01).
- 27.ГОСТ 2.105-95 Единая система конструкторской документации. Общие требования к текстовым документам (Дата введения 1996-07-01).
- 28.Смогунов В.В., Киселева Е.А., Филиппов Б.А. Подготовка, оформление и защита диссертационных работ. Учебное пособие. Пенза 2006- 160с.
http://window.edu.ru/window_catalog/files/r36932/stup158.pdf
- 29.ГОСТ 7.12-93 Система стандартов по информации, библиотечному и издательскому делу. Сокращение слов на русском языке. Общие требования и правила
- 30.ГОСТ 7.1-2003 Система стандартов по информации, библиотечному и издательскому делу. Библиографическое описание документа. Общие требования и правила оформления.
- 31.Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002.- 448с.
- 32.ГОСТ Р ИСО/МЭК ТО 15271-2002 "Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств)"
- 33.ГОСТ Р ИСО/МЭК 15288—2005. Информационная технология. Системная инженерия. Процессы жизненного цикла систем.
- 34.Руководство к Своду знаний по управлению проектами. Третье издание. Американский национальный стандарт ANSI/PMI 99-001-2004.

35. Software & Systems Process Engineering Meta-Model Specification. Version 2.0
<http://www.omg.org/spec/SPEM/2.0/PDF>
36. Кролл П., Крачтен Ф. Rational Unified Process – это легко. Руководство по RUP. Пер. с англ.-М.: КУДИЗ-ОБРАЗ, 2004.- 432с.
37. Курс PRJ270: Essentials of Rational Unified Process
38. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002.-496с.
39. Мцяшек, Лешек, А. Анализ требований и проектирование систем. Разработка систем с использованием UML.: Пер.с англ– М. : Издательский дом «Вильямс», 2002.- 432с.
40. Википедии — свободная энциклопедия. [http:// ru.wikipedia.org/](http://ru.wikipedia.org/)
41. SWOT-анализ как инструмент стратегического менеджмента (подготовлена компанией «БИГ-Петербург» в рамках проекта «7 нот менеджмента»)
http://bigc.ru/publications/swot_analise.php
42. Wikipedia the free encyclopedia. http://en.wikipedia.org/wiki/Main_Page
43. Реинжиниринг: не автоматизируйте – уничтожайте.: Пер с англ. Reengineering Work: Don't Automate, Obliterate by Michael Hammer; Harvard Business Review, July--August 1990. <http://www.cfin.ru/chuvakhin/bpr.shtml>
44. Хаммер М. Реинжиниринг корпораций. Манифест революции в бизнесе/ Майкл Хаммер, Джеймс Чампи; пер. с англ. Ю.Е.Корнилович - М.: Манн, Иванов и Фербер, 2006.-287с.
45. Value Chain Framework
http://www.12manage.com/methods_porter_value_chain_ru.html
46. Тельнов Ю.Ф. Реинжиниринг бизнес-процессов Компонентная методология.- 2-е изд., перераб. и доп. М.: Финансы и статистика, 2004.-320с.
47. Michael Porter's Value Chain // Электронный ресурс:
<http://markus.spb.ru/navalochnaya/porter.shtml>; с ссылкой на: (с) Harvard Business School Publishing , журнал «Искусство управление». – N4. – 2001.
48. TOGAF- The Open Group Architecture Framework. <http://www.opengroup.org/>
49. Zachman Framework www.zifa.com

50. John Zachman/ Concepts of The Framework For Enterprise Architecture/
http://apps.adcom.uci.edu/EnterpriseArch/Zachman/zachman3_files/zachman3.htm
51. Description of Row to Row differences by David Hays.
<http://apps.adcom.uci.edu/EnterpriseArch/Zachman/>
52. Институт типовых решений производства (ИТРП).
<http://www.itrp.ru/content/theory/>
53. Ed Yourdon. Just Enough Structured Analysis
<http://www.yourdon.com/jesa/jesa.php>
54. P. Chen The Entity-Relationship Model-Toward a Unified View of Data. ACM Transactions on Database Systems, Vol. 1, No. 1. March 1976, Pages 9-36.
55. Т. Конноли, К.Брег, А.Страчан, Базы данных: проектирования реализация и сопровождения. Теория и практика, 2-е изд.- М.: Издательский «Вильямс», 2001.-1120с.
56. Коннолли, Томас, Бегг, Карелии. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2003. — 1440 с. : ил. — Парал. тит. англ.
57. Боггс .У., Боггс М. UMLи Rational Rose 2002. – М.: «ЛОРИ», 2004.- 509с.
58. Арлой, Д., Нейштадт, А. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2 –е издание.-Пер.с англ. – СПб: Символ-Плюс, 2007.-624с.
59. Миллер, Р. Базы данных и UML. – М.: «ЛОРИ», 2004.- 420с.
60. Терри Квантрани, Джим Палистрант Визуальное моделирование с помощью IBM Software Architect и UML. Пер с англ.- Б.: КУДИЦ-ПРЕСС.-2007.-192с.
61. Дж. Рамбо, М.Блаха UML2.0. Объектно-ориентированное модельрование и разработка. 2-е изд. СПб.: Питер. 2007- 544с.
62. Розенберг, Д., Скотт, К. Применение объектного моделирования с использованием UML и анализа прецедентов. Пер. с англ. – М.: ДМК Пресс, 2002.- 160 с.

- 63.Ивашко, А.Г., Григорьев, М.В., Коломиец, И.И Проектирование информационных систем. Учебно-методическое пособие. Тюмень: Издательство Тюменского государственного университета, 2007. – 328с.
- 64.Учебный курс “Нотация и семантика языка UML” Интернет Университета информационных технологий (INTUIT). <http://www.intuit.ru>

Примерный план-график подготовки студентом-выпускником квалификационной (дипломной) работы

№ п/п	Этап подготовки	Контрольная точка	Рекомендуемые сроки
1.	Выбор научного руководителя	Заявление, подписанное руководителем или распоряжение по кафедре	до 1 октября
2.	Выбор темы дипломной работы	Задание на выполнение квалификационной работы.	до 1 ноября
3.	Выбор места прохождения преддипломной практики	Договор нахождение производственной практикой	до 15 ноября
4.	Составление первоначального плана дипломной работы	План выполнения преддипломной практикой и квалификационной работы	до 1 декабря
5.	Прохождение преддипломной практики и выполнение задач, поставленных в квалификационной работе	Отчет по преддипломной практике (защита отчета проводится в первую неделю после завершения практики). Реализованные модели и программные компоненты.	С середины декабря 14 недель
6.	Написание текста дипломной работы, представление первоначального варианта дипломной работы научному руководителю	Первоначальный вариант текста квалификационной работы и разработанные компоненты информационной системы или программного обеспечения	до 1 мая
7.	Доработка дипломной работы в соответствии с замечаниями руководителя	Допуск руководителя к предварительной защите, текст квалификационной работы, доклад и демонстрационные материалы	до 20 мая
8.	Предзащита дипломной работы на заседании выпускающей кафедры	Допуск к защите, назначение рецензента квалификационной работы	до 27 июня
9.	Доработка дипломной работы в соответствии с замечаниями, высказанными на предзащите, окончательное оформление	Окончательный вариант дипломной работы	до 5 июня
10.	Получение отзыва научного руководителя, передача диплома на рецензирование	Отзыв научного руководителя	до 10 июня

11.	Получение рецензии, передача завершённой работы на рецензирование	Рецензия на квалификационную работу (в том случае если рецензент не является работником университета, то его подпись на отзыве должна быть заверена печатью организации)	до 15 июня
12.	Подготовка к защите (подготовка доклад и раздаточного материала, разработка презентации в MS Power-Point)	Доклад, раздаточный материал и презентация	до 20 июня
13.	Защита дипломной работы		вторая половина июня
	<i>Всего: 32-36 недель</i>		

Приложение Б
Шаблон задания на выполнение дипломной работы

ГОУ ВПО ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКА И КОМПЬЮТЕРНЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

Направление (специальность) _____

Группа _____

УТВЕРЖДАЮ
Заведующий кафедрой

(ФИО, должность, степень)

(Подпись, дата)

Дата выдачи задания

ЗАДАНИЕ НА ДИПЛОМНУЮ РАБОТУ

Студенту _____
(Фамилия, И., О.)

Научный руководитель _____
(Фамилия, И., О., место работы, должность)

1. Тема: _____

2. Назначение и цель системы

Подпись научного руководителя _____

3. Требования к объекту исследования или разработки

4. Перечень разрабатываемых моделей и алгоритмов

5. Исходные материалы и источники разработки

6. Консультанты с указанием относящихся к ним разделов работы

Раздел	Консультант	Подпись, дата	
		Задание выдал	Задание принял

7. Календарный план

№	Наименование этапа работы	Срок выполнения этапа	Примечание
	Сдача отчета по преддипломной практике		
	Предзащита		
	Сдача законченной дипломной работы		

Задание принял к исполнению _____
(подпись студента)

Подпись научного руководителя _____

Основные понятия жизненного цикла разработки по RUP

(Примеры показаны в контексте RUP 2003)

Унифицированный процесс не регламентирует использование объектно-ориентированного подхода. В тоже время он является специализацией Software Process Engineering Metamodel (SPEM¹), спецификация которого, в свою очередь, описана на основе языка UML. Объектно-ориентированная концепция SPEM побуждает использовать объектно-ориентированный подход при проектировании системы. В дальнейшем будут встречаться формулировка целей и задач с использованием терминологии языка UML.

Общая архитектура организации жизненного цикла RUP представлена на рисунке В.1 /35, 36/.

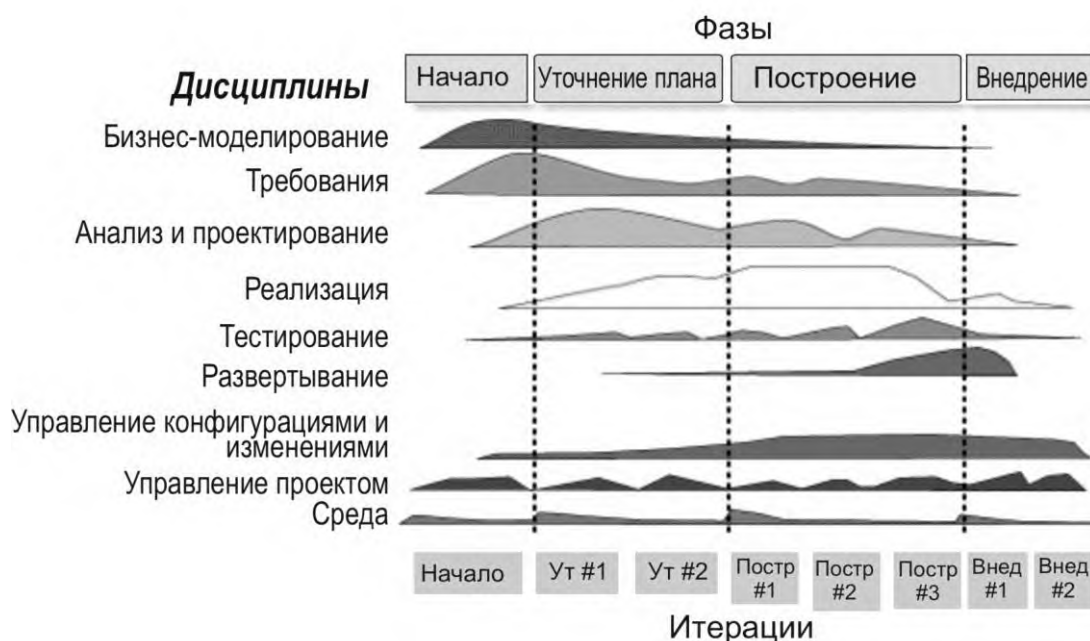


Рисунок В.1 – Архитектура жизненного цикла RUP/

У процессной организации можно выделить две ортогональные структуры или два измерения: динамическая структура и статическая

¹ Software Process Engineering Meta-model (SPEM) — метамодель, предложенная в 2002 г. группой по объектному управлению OMG и первоначально нацеленная на процессы разработки ПО.

структура. Горизонтальное измерение представляет собой динамическую структуру, или временное измерение процесса. Оно показывает, как процесс реализован во времени и включает понятия: фазы, итерации, вехи. Вертикальное измерение представляет собой статическую структуру процесса. Оно описывает такие понятия как дисциплины, артефакты, роли, дисциплины процесса и рабочие процессы.

Динамическая структура RUP.

Временное измерение проекта разделяется на четыре фазы.

Во время выполнения фазы «Начало» определяются границы проекта. Определяются акторы и прецеденты системы. Акторы организуют окружающую среду, а прецеденты определяют границы проекта. Разрабатывается бизнес-план для того, чтобы определить, какие ресурсы и в каком количестве необходимо вложить в проект. По окончании этой фазы необходимо определить основные прецеденты.

В течение фазы «Уточнение» уделяется внимание двум вопросам: окончательно определиться с требованиями и установить базовый план архитектуры. Определение требований и архитектуры дает возможность избежать технических рисков, а также определить план работы на будущее. К концу фазы нужно более точно оценить стоимость и ресурсы проекта.

В конце фазы «Уточнение» должна быть подготовлена рабочая версия системы, которая реализует ключевые сценарии прецедентов и делает акцент на наиболее критические части архитектуры.

Во время фазы «Построение» продукт доводится до бета-версии. На этой стадии итерации реализуют дополнительные сценарии прецедентов.

Во время фазы «Внедрение» продукт передается конечному пользователю, при этом проводятся мероприятия по их обучению, а также по развертыванию и поддержке продукта.

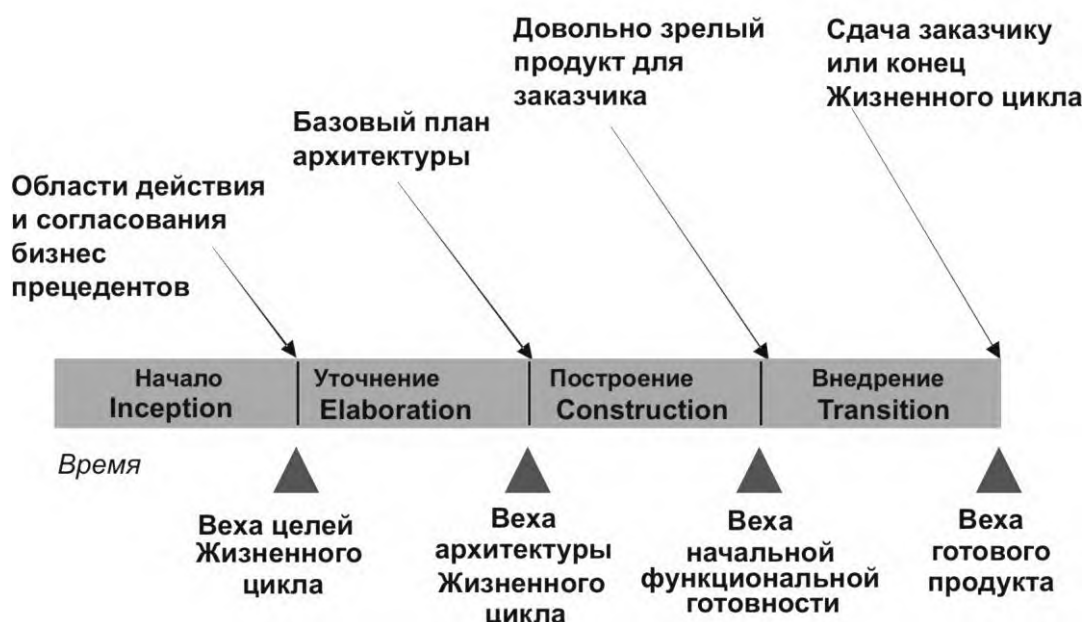
Итераций² на каждой фазе проводится столько, сколько надо для достижения бизнес-целей. Если на запланированной итерации не удастся достигнуть поставленных целей, то необходимо добавить еще одну итерацию.

Для каждой фазы определяются цели и критерии оценки. По завершению фазы (рисунок В.2) на каждой основной вехе пересматривается проект и принимается решение, либо продолжить проект, либо свернуть его, либо внести в него исправление. Критерии, влияющие на это решение, отличаются на каждой фазе.

ЛСО (Веха целей жизненного цикла): согласованность границ проекта, риски поняты и обоснованы.

ЛСА (Веха архитектуры жизненного цикла): самые высокие риски устранены и достигнута устойчивость архитектуры.

ЮС (Веха начальной функциональной готовности): продукт завершен, а его качество приемлемо для пользователей.



² Итерация – это четкая последовательность действий, основанная на утвержденном плане и критериях оценки, приводящая к конечному результату в виде версии (для внутреннего или внешнего использования) исполняемого продукта.

Рисунок В.2 – Жизненный цикл RUP.

Критерии оценки для фазы «Начало» (LCO) включают:

- достигнуто согласие с заказчиком по поводу границ системы, стоимости и графика работ;
- понимание требований подтверждено согласованием основных прецедентов;
- проведена достоверная оценка стоимости и графика работ, определены приоритеты и основные риски процесса разработки;
- определены глубина и объем всех возможных прототипов архитектуры;
- проведено сопоставление фактических затрат с запланированными.

Критерии оценки для фазы «Уточнение» (LCA) включают:

- достижение устойчивости архитектуры и видение³ продукта;
- анализ элементов основных рисков;
- обоснованность планов и оценок ресурсов необходимых для завершения проекта;
- одобрение заказчиком видения продукта (макета продукта) и плана проекта;
- приемлемый уровень затрат.

Критерии оценки для фазы «Построение» (IOC) включают:

- стабильность, устойчивость и зрелость выпускаемого продукта;
- готовность заказчика к развертыванию продукта (внедрению);
- приемлемый уровень затрат.

³ Видение или концепция (Vision) – Представление пользователей или заказчика о продукте, который должен быть разработан, суммарно определяемое на уровне ключевых потребностей заинтересованных лиц и главных свойств системы. /36/

По окончании фазы «Развертывание» на основе уровня удовлетворенности заказчика, достигаемого на протяжении этой фазы, принимается решение по выпуску продукта. Часто эта веха совпадает с началом следующего цикла разработки, связанного с улучшением и доработкой продукта.

В пределах каждой фазы имеется ряд итераций. Число итераций в каждой фазе различно. Каждая итерация – это завершённый этап, в результате которого выпускается версия (для внутреннего или внешнего использования) артефактов проекта, реализующих запланированные действия в данной итерации.

Артефакт, предназначенный для внутреннего использования (внутренний), остается в среде разработки и (необязательно) демонстрируется заказчику. Подготовка артефакта для внешнего представления более дорогостоящая процедура, т.к. требуется подготовка документации для пользователя и для технической поддержки. Внешние артефакты готовятся обычно на стадии «Внедрение».

Окончание итерации обозначают как вторичные вехи. В этой точке оценивают технические результаты и по мере надобности пересматривают планы на будущее.

Статическая структура RUP.

Статическая структура определяет организацию таких элементов процессов как задачи (или действия), роли, артефакты и дисциплины. Унифицированный процесс делится на логические группы – «Дисциплины», которые объединяют роли, задачи, артефакты и другие средства управления процессом. Руководство RUP предупреждает, что нельзя рассматривать дисциплины как стадии выполнения каскадной модели. Действия дисциплин выполняются многократно по всему жизненному циклу процесса создания ПО.

Содержание дисциплины организовано в следующие секции.

Представление: цель дисциплины и отношение с другими дисциплинами.

Основные принципы: важны для понимания ключевые определения и концепции дисциплины.

Рабочий процесс: поток действий (задач) для достижения измеримого результата, выраженный в терминах деталей рабочего процесса. Рабочий процесс представлен деталями рабочего процесса в виде диаграммы деятельности (рисунок В.3).

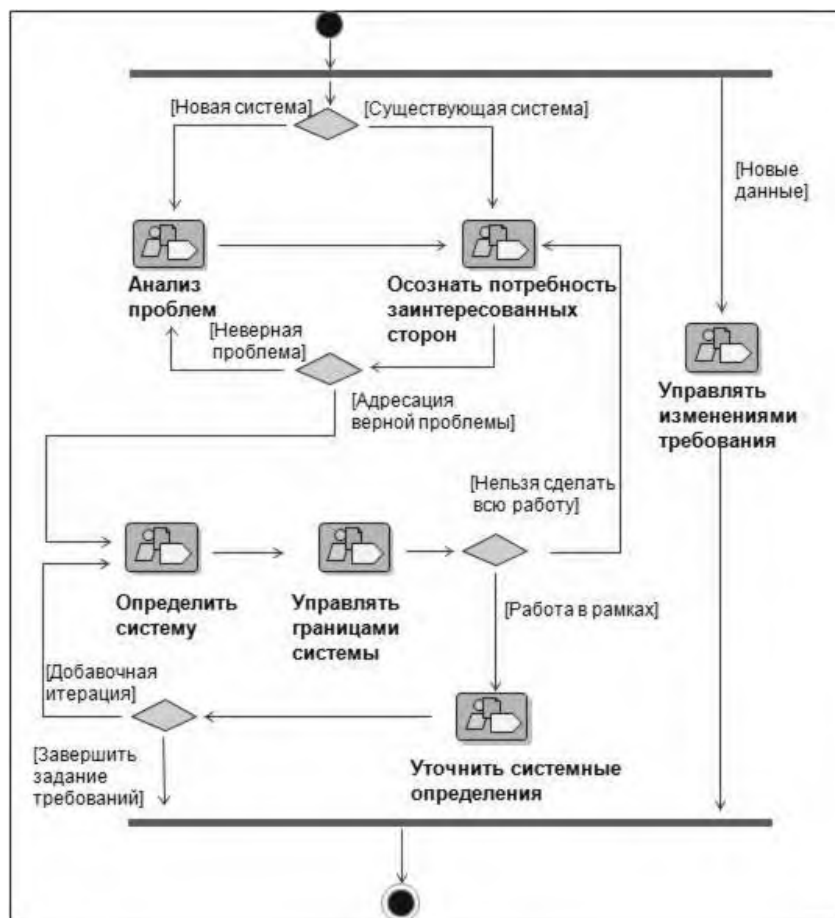


Рисунок В.3 – Пример рабочего процесса управления требованиями (RUP 2003).

Детали рабочего процесса являются в свою очередь контейнерами для действий (задач), сгруппированных по ролям, входных и выходных артефактов и заинтересованных сторон. Выполнение этих действий тесно связано и выполняются параллельно и итерационно при этом достигается определенный результат. Унифицированный процесс вводит понятие де-

тали рабочего процесса для того, что бы представить процесс на более высоком уровне абстракции, а также для того, что бы упростить понимание последовательности выполнения задач.

Краткий обзор детали рабочего процесса: графическое представление действий, которые выполнены в этой дисциплине, и ролей, которые исполняют их.

Краткий обзор артефактов: графическое представление артефактов, которые произведены в этой дисциплине, и ролей, которые являются ответственными.

Руководство: обзор правил, рекомендаций и эвристических методов для выполнения задач, последовательности действий и работы с артефактами.

Инструкции по работе с инструментальными средствами: устанавливает связи с представлением практического руководства по применению инструментальных средств разработки.

В RUP роль определяет поведение и ответственность индивидуума или ряда индивидуумов, работающих совместно в команде. Действия или задачи это единица работы выполняемая конкретной ролью. Для действия определяется цель, которая может формулироваться как создание одного артефакта (выходного) на основании информации других (входные) артефактов. Обычно задача используется как элемент планирования или оценки прогресса процесса. Действие может быть выполнено многократно от итерации к итерации.

Действия подразделяются на шаги, которые подразделяются на шаги: продумывания, исполняющие и проверочные. Не все шаги могут выполняться при выполнении действия.

Артефакты это материальные элементы проекта: модели, элементы моделей, документы, элементы документов, исходный код, исполняемые файлы. Артефакты могут иметь формальную структуру (модель прецедентов) или не иметь (интервью).

Шесть дисциплин RUP выполняются в каждой итерации в каскадной модели (рисунок В.4 – не показана дисциплина бизнес моделирование). От итерации к итерации в каждой дисциплине могут меняться детали рабочего процесса, действия, шаги в действиях, артефакты. Более того в различных итерациях могут различные люди выполнять одну и ту же роль.

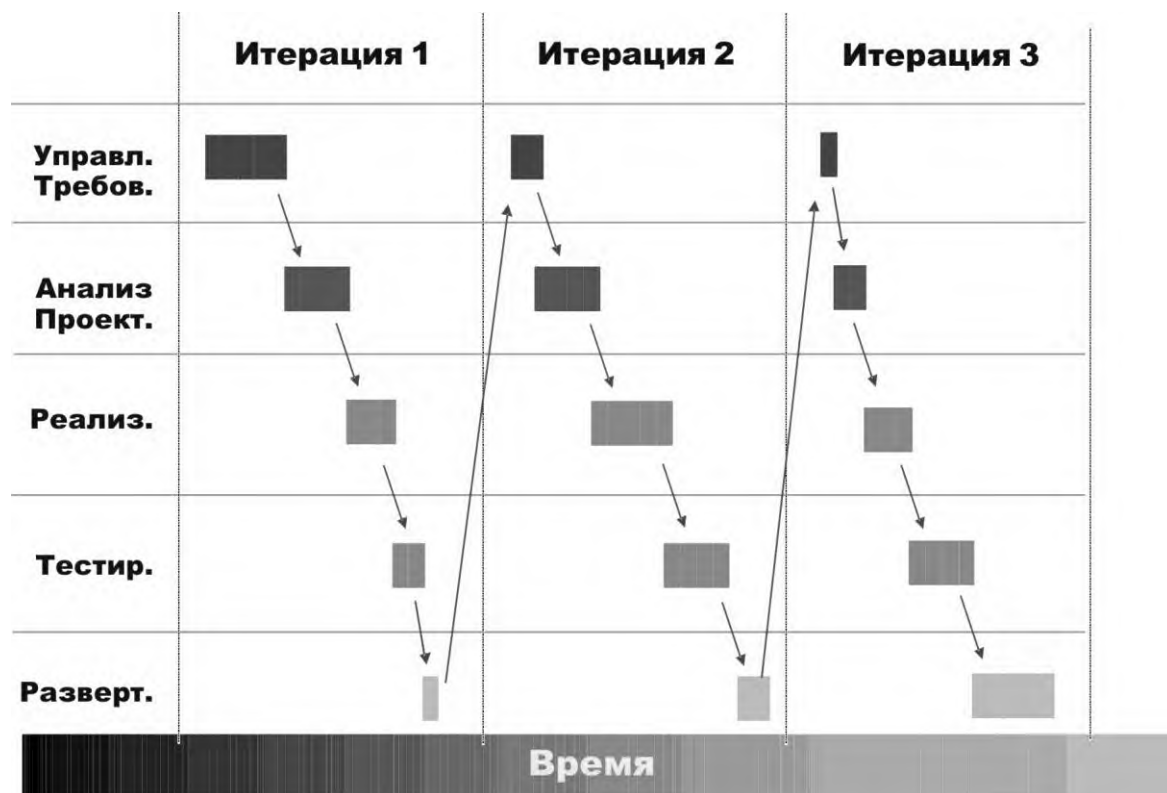


Рисунок В.4 - Изменение фокуса в итерациях со временем.

Артефакты это материальные элементы проекта: модели, элементы моделей, документы, элементы документов, исходный код, исполняемые файлы. Артефакты могут иметь формальную структуру (модель прецедентов) или не иметь (интервью).

RUP определяет шаблоны каждого артефакта. Ниже приводятся шаблоны документов: видение, описания прецедента, дополнительная спецификация,

<Название проекта>

Видение

Примечание: Настоящее руководство в шаблоне отформатировано с помощью стиля InfoBlue. В этом стиле шрифт имеет атрибут «скрытый», что позволяет вам управлять видимостью текста руководства в шаблоне. Для этого используйте меню Сервис → Параметры → Вид → флажок «Скрытый текст». Подобная опция существует и для управления видимостью текста при печати: Сервис → Параметры → Печать.

Введение

[Целью настоящего документа является сбор, анализ и определение в общем виде потребностей пользователей <Название системы> и её возможностей. Здесь рассматриваются потребности всех заинтересованных сторон и конечных пользователей, а также причины появления этих потребностей. Более подробная информация о том, как <Название системы> удовлетворяет эти потребности, приводится в модели прецедентов и в дополнительных спецификациях.]

[Введение предоставляет собой обзор содержимого документа видения. Оно включает в себя:

- цели создания документа;*
- область применения;*
- используемые в тексте определения, акронимы и аббревиатуры;*
- ссылки на другие документы или иные источники информации;*
- сведения об организации настоящего документа.]*

Позиционирование продукта

Формулировка проблемы

[Здесь приводится обобщённая формулировка проблемы, на решение которой будет направлен проект. Для этого может быть использован следующий формат:]

Проблема	<i>[описание проблемы]</i>
Область интересов	<i>[стороны, которых затрагивает проблема]</i>
Основное проявление	<i>[в чём состоит основное проявление проблемы?]</i>
Успешное решение позволит	<i>[список ключевых достижений, которые должны быть получены в случае успешного решения проблемы]</i>

Определение позиции продукта

[Здесь приводится полное (на высшем уровне абстракции) определение той позиции, которую, как предполагается, продукт займёт на рынке. Для этого может быть использован следующий формат:]

Целевая аудитория	<i>[целевая аудитория пользователей]</i>
Потребности	<i>[формулировка потребности или возможности]</i>
Категория продукта	<i>[категория продукта]</i>
Предложение	<i>[формулировка ключевых характеристик продукта, которые станут основной причиной его покупки]</i>
Основные конкуренты	<i>[главная альтернатива продукта]</i>
Преимущества	<i>[формулировка главного отличия предлагаемого решения]</i>

[Позиция продукта определяет цель создания приложения и значение проекта для участвующего в нём персонала]

Заинтересованные стороны

Общие сведения о заинтересованных сторонах

Тип	Описание	Обязанности
<i>[Название типа заинтересованной стороны.]</i>	<i>[Краткая характеристика заинтересованной стороны.]</i>	<i>[Ключевые обязанности заинтересованной стороны в процессе разработки продукта, сформулированные в общем виде. Обязанности могут быть, например, такими:</i> <ul style="list-style-type: none"><i>• обеспечить обслуживание системы;</i><i>• обеспечить рыночный спрос на возможности продукта;</i><i>• наблюдать за ходом выполнения проекта;</i><i>• осуществлять финансирование;</i> <i>и т.д.]</i>

Условия использования продукта

[Здесь приводятся подробная информация о среде, в которой находится целевая аудитория пользователей продукта. Основные аспекты, которые необходимо рассмотреть:]

- Каково количество людей, вовлечённых в выполнение задачи? Изменяется ли это число?*
- Какова продолжительность цикла решения задачи? Какова продолжительность времени, затрачиваемого на каждый вид деятельности? Изменяется ли это значение?*
- Какие существуют специфические ограничения, обусловленные влиянием окружающей среды: требования к мобильности, работе в условиях улицы, в полёте и т.д.?*
- Какие системные платформы используются в данный момент? Какие платформы будут использованы в будущем?*
- Какие другие приложения используются в данный момент или будут использоваться? Существует ли необходимость интегрировать ваше приложение с ними?*

Вот куда могут быть включены элементы Бизнес-модели, с целью обозначить задачу, роли и т.д.]

Общее представление о продукте

Окружение продукта

[В этом подразделе Видения продукт рассматривается во взаимосвязи с другими продуктами и пользовательским окружением. Если продукт является независимым и полностью автономным, это необходимо указать именно в данном подразделе. Если продукт является компонентом более крупной системы, тогда в данном подразделе должен быть определён механизм взаимодействия этих систем и набор необходимых интерфейсов. Одним из наиболее простых способов демонстрации основных компонентов крупной системы, взаимосвязей между ними, а также внешних интерфейсов системы является блок-диаграмма.]

Допущения и зависимости

[Здесь необходимо перечислить все факторы, которые влияют на возможности продукта, приведенные в Видении. Также необходимо перечислить допущения, изменение которых приведёт к изменению Видения. Например, допущение может предполагать наличие специальной операционной системы на аппаратном обеспечении, отведенном для программного продукта. Если операционная система отсутствует, Видение необходимо изменить.]

Потребности и возможности

[В данном разделе не следует раскрывать ваши проектные решения. Описание возможностей системы нужно приводить в общем виде, уделяя основное внимание определению требуемых возможностей продукта и причинам (но не способам) их разработки.]

Потребность	Приоритет	Возможность	Плановый момент реализации

Альтернативы и их оценка

[Здесь необходимо определить альтернативные варианты, доступные для заинтересованных сторон. Сюда можно включить:

- *покупку продукта у конкурентов;*
- *разработку собственных решений;*
- *сохранение текущей ситуации.*

Необходимо перечислить известные конкурирующие варианты, которые уже существуют или могут быть достигнуты. По каждому варианту необходимо отметить его основные сильные и слабые стороны, воспринимаемые заинтересованными сторонами или конечным пользователем.]

Прочие требования к продукту

[В этом разделе необходимо перечислить применяемые стандарты, требования к аппаратному обеспечению или платформе, требования к производительности и состоянию окружающей среды.

Необходимо привести требования к качеству выполнения, надежности, отказоустойчивости, практичности (удобству и простоте использования) и прочие подобные характеристики, не отмеченные в списке возможностей.

Также необходимо отметить ограничения разработки, внешние ограничения, или другие зависимости.

Необходимо привести специфические требования к документации, включая руководство пользователя, электронную справочную систему, инструкцию по установке, маркировку (нанесение маркировочных знаков) и упаковку.

Необходимо определить приоритет всех перечисленных в данном разделе требований. При этом, если необходимо, используйте такие характеристики требований, как стабильность (устойчивость), польза (выгода), требуемый объем работ и риск.]

<Название проекта> Дополнительные спецификации

Версия <Версия>

[Примечание: Настоящий шаблон предназначен для использования в Rational Unified Process. Текст синего цвета (стиль «InfoBlue»), который заключён в квадратные скобки, содержит указания для автора документа и должен быть удалён перед его публикацией. Абзац, следующий за такими указаниями, будет автоматически отформатирован с помощью стиля основного текста документа (стиль «Body Text».)]

[Для того чтобы настроить автоматические поля в Microsoft Word (при выделении эти поля отображаются с серым фоном), выберите меню Файл>Свойства и заполните атрибуты «Название», «Тема» и «Учреждение» соответствующими значениями. После того, как вы закроете диалог, автоматические поля в документе могут быть обновлены следующим образом: выберите пункт меню Правка>Выделить всё (или Ctrl-A) и нажмите F9 или просто кликните на нужное поле и нажмите F9. Обновление полей для верхнего и нижнего колонтитула должно быть сделано отдельно. Сочетание Alt-F9 позволяет переключаться между режимом отображения кодов автоматических полей и режимом отображения их значений. В справочной системе Word можно найти более подробную информацию о работе с полями.]

История изменений

Дата	Версия	Описание	Автор
<дд.ММ.гггг>	<х.х>	<описание>	<Ф.И.О.>

Содержание

1.	Введение	246
1.1	Цели	246
1.2	Область применения	246
1.3	Определения, сокращения и аббревиатуры	246
1.4	Ссылки	246
1.5	Общие сведения	246
2.	Функциональность	246
2.1	<Требование к функциональности 1>	247
3.	Практичность	247
3.1	<Требование к практичности 1>	247
4.	Надежность	247
4.1	<Требование к надежности 1>	247
5.	Производительность	248
5.1	<Требование к производительности 1>	248
6.	Поддержка	248
6.1	<Требование к поддержке 1>	248
7.	Ограничения проектирования	248
7.1	<Ограничение проектирования 1>	248
8.	Требования к электронной справочной системе и системе помощи	248
9.	Покупные компоненты	248
10.	Интерфейсы	249
10.1	Пользовательские интерфейсы	249
10.2	Аппаратные интерфейсы	249
10.3	Программные интерфейсы	249
10.4	Интерфейсы связи	249
11.	Требования к лицензированию	249
12.	Уведомления	249
13.	Применяемые стандарты	249

Дополнительные спецификации

Введение

[Введение представляет собой обзор содержимого данного документа. Оно включает в себя:

- *цели создания документа;*
- *область применения;*
- *используемые в тексте определения, сокращений и аббревиатуры;*
- *ссылки на другие документы или иные источники информации;*
- *сведения об организации настоящего документа.*

В настоящем документе представлены системные требования, которые трудно описать в Модели прецедентов. Такие требования включают в себя:

- *Правовые и регулятивные требования и применяемые стандарты.*
- *Характеристики качества разрабатываемой системы, включая практичность, надёжность, производительность и возможность поддержки.*
- *Прочие требования, такие как требования к операционным системам и окружению, требования к совместимости и ограничения проектирования.]*

Цели

[Определите цель создания настоящего документа.]

Область применения

[Краткое определение области применения данного документа; список проектов, связанных с ним, и всё остальное, на что оказывает влияние данный документ.]

Определения, сокращения и аббревиатуры

[Данный подраздел содержит определения всех терминов, сокращений и аббревиатур, необходимых для правильной интерпретации документа. Эта информация может быть представлена в виде ссылок на статьи Глоссария проекта.]

Ссылки

[Данный подраздел содержит полный перечень всех документов, на которые содержится ссылка в документе. Для каждого документа указывается название, номер публикации (если это необходимо), дата и организация-издатель. Укажите источники, из которых были получены данные ссылки. Эта информация может быть представлена ссылкой на приложение или другой документ.]

Общие сведения

[В данном подразделе даётся краткое описание содержимого документа, объясняется его организация.]

Функциональность

[В данном подразделе дается описание тех функциональных требований к системе, которые выражены на естественном языке. Для многих приложений это может составлять большую часть SRS-пакета (пакета требований к программному обеспечению), а потому следует подумать над организацией этого раздела. Обычно данный подраздел организуют относительно возможностей, но подходят также и другие методы его организации, например, относительно пользователей или

подсистем. В функциональные требования можно включить возможности системы (разного уровня абстракции), а также требования к безопасности.

В тех местах раздела, где для фиксирования функциональности применяются специальные инструменты (средства управления требованиями, средства моделирования и т.д.), необходимо дать описание того как и с помощью каких средств (название и местоположение инструмента) можно получить доступ к этим данным.]

<Требование к функциональности 1>

[Описание требования.]

Практичность

[В данный раздел должны быть включены все требования, которые влияют на практичность. Например:

- Специфицируйте время, необходимое обычному или продвинутому пользователю, чтобы научиться эффективно выполнять отдельные операции.
- Определите измеримое время выполнения типичных задач.
- Определите основные требования к практичности новой системы на примере других систем, которые пользователь знает и любит.
- Специфицируйте требования по соответствию общим стандартам практичности – например, стандартам IBM CUA или GUI-стандартам, опубликованным Microsoft для Windows 95.]

<Требование к практичности 1>

[Описание требования.]

Надежность

[Здесь должны быть приведены требования к надежности системы. Советуем рассмотреть следующие аспекты:

- Доступность – время доступности системы (в процентах от общего времени её работы), допустимое время работы системы (в часах), доступ для технического обслуживания, допустимые операции в режиме сниженной функциональности и т.д.
- Средняя наработка на отказ (MTBF) – обычно измеряется в часах, но можно использовать также дни, месяцы или годы.
- Среднее время восстановления (MTTR) – как долго система может быть неисправной в случае отказа.
- Точность – используя известные стандарты, определите точность выходных параметров системы.
- Максимально допустимое число ошибок или интенсивность отказов – обычно выражается в количестве ошибок на тысячу строк кода или на одну функцию.
- Виды ошибок – классификация ошибок на незначительные, значительные и критические. Например, требования должны определять, что подразумевается под «критической» ошибкой: полная потеря данных, невозможность использования некоторых функций системы.]

<Требование к надежности 1>

[Описание требования.]

Производительность

[В данном разделе должны быть определены параметры производительности. Включите сюда специальные требования к времени отклика системы. Там, где необходимо, приведите ссылки на соответствующие Прецеденты.]

- *Время отклика для транзакции (среднее, максимальное значения).*
- *Пропускная способность (например, транзакций в секунду).*
- *Емкость (например, количество потребителей или транзакций, на которое рассчитана система).*
- *Режимы сниженной функциональности (какие операции допустимы, когда функциональность системы снижена по каким-то причинам.)*
- *Использование ресурсов: оперативная и дисковая память, средства коммуникации и т.д.]*

<Требование к производительности 1>

[Описание требования.]

Поддержка

[В данном подразделе приводятся все требования, направленные на улучшение поддержки или ремонтпригодности разрабатываемой системы, включая стандарты кодирования, соглашения по именованию, библиотеки классов, доступ для технического обслуживания, утилиты для технического обслуживания.]

<Требование к поддержке 1>

[Описание требования.]

Ограничения проектирования

[В данном подразделе приводятся все ограничения проектирования для разрабатываемой системы. Ограничения проектирования отражают проектные решения, которые predeterminedены и которых следует придерживаться. Это, например, требования к применению конкретных языков, процессу разработки, применяемому инструментарию, а также архитектурные и проектные ограничения, покупные компоненты, библиотеки классов и т.д.]

<Ограничение проектирования 1>

[Описание требования.]

Требования к электронной справочной системе и системе помощи

[Опишите, если необходимо, требования к электронной справочной системе, системам помощи, информации о программе и т.д.]

Покупные компоненты

[В данном разделе приводится описание всех покупных компонентов, которые должны применяться в системе, ограничения относительно их лицензирования и использования, а также все связанные с этим стандарты совместимости/интероперабельности и стандарты интерфейсов.]

Интерфейсы

[В данном разделе приводятся определения интерфейсов, которые должны поддерживаться приложением. Для того чтобы программное обеспечение разрабатывалось и верифицировалось в соответствии с требуемыми интерфейсами, необходимо указать протоколы, порты и логические адреса и т.д.]

Пользовательские интерфейсы

[Опишите пользовательский интерфейс, который должен быть реализован в системе.]

Аппаратные интерфейсы

[В данном разделе определяются все аппаратные интерфейсы, которые должна поддерживать система, включая логическую структуру, физические адреса, ожидаемый режим работы и т.д.]

Программные интерфейсы

[В данном разделе приводится описание программных интерфейсов с другими компонентами системы. Это могут быть покупные компоненты, повторно используемые компоненты другого приложения, или компоненты, разработанные для подсистем, не входящих в рамки данного SRS, но с которыми данное приложение должно взаимодействовать.]

Интерфейсы связи

[Приведите все интерфейсы связи с другими системами или устройствами: локальная вычислительная сеть, удалённые серийные устройства и т.д.]

Требования к лицензированию

[Определите все требования к лицензионным ограничениям или другим ограничениям использования, которые должна иметь система.]

Уведомления

[В данном разделе приводятся все необходимые для данной системы правовые уведомления, уведомления об авторском праве, патентные уведомления, различные предупреждения, а также торговая марка, логотип и т.д.]

Применяемые стандарты

[В данном разделе приводятся ссылки на все используемые стандарты (и их разделы). Например, это могут быть правовые стандарты, стандарты качества, стандарты управления, промышленные стандарты практичности, интероперабельности, интернационализации, соглашения операционной системы и т.д.]

<Название проекта>

Спецификация прецедента: <Название прецедента>

Версия <Версия>

[Примечание: Настоящий шаблон предназначен для использования в Rational Unified Process. Текст синего цвета (стиль «InfoBlue»), который заключён в квадратные скобки, содержит указания для автора документа и должен быть удалён перед его публикацией. Абзац, следующий за такими указаниями, будет автоматически отформатирован с помощью стиля основного текста документа (стиль «Body Text».)]

[Для того чтобы настроить автоматические поля в Microsoft Word (при выделении эти поля отображаются с серым фоном), выберите меню Файл>Свойства и заполните атрибуты «Название», «Тема» и «Учреждение» соответствующими значениями. После того, как вы закроете диалог, автоматические поля в документе могут быть обновлены следующим образом: выберите пункт меню Правка>Выделить всё (или Ctrl-A) и нажмите F9 или просто кликните на нужное поле и нажмите F9. Обновление полей для верхнего и нижнего колонтитула должно быть сделано отдельно. Сочетание Alt-F9 позволяет переключаться между режимом отображения кодов автоматических полей и режимом отображения их значений. В справочной системе Word можно найти более подробную информацию о работе с полями.]

История изменений

Дата	Версия	Описание	Автор
<дд.ММ.гггг>	<х.х>	<описание>	<Ф.И.О.>

Содержание

1.	Название прецедента	253
1.1	Краткое описание	253
2.	Потоки событий	253
2.1	Основной поток	253
2.2	Альтернативные потоки	253
	2.2.1 <Альтернативный поток 1>	253
	2.2.2 <Альтернативный поток 2>	254
3.	Специальные требования	254
3.1	<Специальное требование 1>	254
4.	Предусловия	254
4.1	<Предусловие 1>	254
5.	Постусловия	254
5.1	<Постусловие 1>	254
6.	Точки расширения	254
6.1	<Название точки расширения 1>	255

Спецификация прецедента: <Название прецедента>

Название прецедента

Краткое описание

[Данное описание должно кратко отражать роль и назначение прецедента. Одного абзаца должно быть достаточно.]

Потоки событий

Основной поток

[Прецедент начинается тогда, когда актер начинает производить какие-либо действия. Именно актер всегда инициирует прецеденты. Прецедент должен описывать, что делает актер и что система делает в ответ. Описание должно быть сформулировано в форме диалога между актером и системой.]

*Прецедент должен описывать **что** происходит внутри системы, но не как и не почему это происходит. Если происходит обмен информацией, то должны быть указаны входные и выходные данные. Например, будет недостаточно сказать, что актер вводит информацию о клиенте; лучше сказать, что актер вводит имя и адрес клиента. При этом Глоссарий терминов часто может оказаться полезным для управления сложностью прецедента. Вы можете также определить здесь и информацию о клиенте, но помните, что не стоит перегружать прецедент деталями.*

Простые альтернативные потоки могут быть представлены прямо в тексте описания прецедента. Если описание альтернативного потока занимает всего несколько предложений, приведите его прямо в этом разделе. Если альтернативный поток более сложный, используйте для его описания специальный раздел настоящего документа.

Порой рисунок «стоит тысячи слов», хотя полностью и не заменяет собой четкого, понятного описания. Поэтому если необходимо, вставьте в описание прецедента графические представления пользовательских интерфейсов, технологических маршрутов и т.д. Если блок-схема полезна для представления сложного процесса принятия решений, обязательно используйте её! Подобным же образом диаграмма состояний отражает поведение системы лучше, чем множество страниц текста. Используйте правильный способ представления информации о вашей проблеме, но будьте осторожны при использовании терминологии, графических нотаций и фигур, которые ваша аудитория может не понять. Помните, что ваша цель не затуманить, а прояснить.]

Альтернативные потоки

<Альтернативный поток 1>

[Более сложные альтернативные потоки событий должны быть описаны в отдельном разделе, ссылки на который содержатся в разделе с основным потоком. Каждый альтернативный поток представляет собой сценарий некоторого альтернативного поведения. Таких потоков может быть несколько – по числу исключительных ситуаций в основном потоке. Потоки могут быть длинны настолько, насколько это необходимо для описания событий связанных с альтернативным

поведением. Когда альтернативный поток заканчивается, события основного потока возобновляются до тех пор, пока не произойдет другая исключительная ситуация.]

<Альтернативный подпоток>

[В случае необходимости описание альтернативных потоков может быть разбито на подразделы.]

<Альтернативный поток 2>

[Скорее всего, прецедент будет содержать несколько альтернативных потоков. Храните описание каждого альтернативного потока отдельно. Использование альтернативных потоков повышает читаемость прецедента, а также не допускает его декомпозицию в иерархию прецедентов. Помните, что прецеденты – это всего лишь набор текстовых спецификаций, главная цель которых ясно, кратко и понятно документировать поведение системы.]

Специальные требования

[Специальные требования – это типичные нефункциональные требования, характерные для прецедента, которые не могут быть легко специфицированы в тексте потока событий. Примером таких требований могут быть юридические и регулятивные требования, применяемые стандарты, а также параметры качества создаваемой системы, включая практичность, надежность, производительность и поддержку. Здесь могут быть приведены и прочие требования, например: требования к операционным системам и окружению, требования к совместимости, ограничения проектирования.]

<Специальное требование 1>

Предусловия

[Предусловие (прецедента) – это состояние, в котором должна находиться система перед началом выполнения прецедента.]

<Предусловие 1>

Постусловия

[Постусловие (прецедента) – это список состояний, в котором может оказаться система сразу после завершения прецедента.]

<Постусловие 1>

Точки расширения

[Точки расширения прецедента.]

<Название точки расширения 1>

[Определение позиции точки расширения в потоке событий.]

<Название проекта> Перечень рисков

Версия <Версия>

[Примечание: Настоящий шаблон предназначен для использования в Rational Unified Process. Текст синего цвета (стиль «InfoBlue»), который заключён в квадратные скобки, содержит указания для автора документа и должен быть удалён перед его публикацией. Абзац, следующий за такими указаниями, будет автоматически отформатирован с помощью стиля основного текста документа (стиль «Body Text»).]

[Для того чтобы настроить автоматические поля в Microsoft Word (при выделении эти поля отображаются с серым фоном), выберите меню Файл>Свойства и заполните атрибуты «Название», «Тема» и «Учреждение» соответствующими значениями. После того, как вы закроете диалог, автоматические поля в документе могут быть обновлены следующим образом: выберите пункт меню Правка>Выделить всё (или Ctrl-A) и нажмите F9 или просто кликните на нужное поле и нажмите F9. Обновление полей для верхнего и нижнего колонтитула должно быть сделано отдельно. Сочетание Alt-F9 позволяет переключаться между режимом отображения кодов автоматических полей и режимом отображения их значений. В справочной системе Word можно найти более подробную информацию о работе с полями.]

Версия **Ошибка!** **Неизвестное**
имя свойства до-
кумента.

3

29.11.2022

Ошибка! Неизвестное имя свойства документа.

История изменений

Дата	Версия	Описание	Автор
<дд.ММ.гггг>	<х.х>	<описание>	<Ф.И.О.>

Ошибка! Неизвестное имя свойства документа.

Содержание

1.	Введение	259
1.1	Цели	259
1.2	Область применения	259
1.3	Определения, сокращения и аббревиатуры	259
1.4	Ссылки	259
1.5	Общие сведения	259
2.	Риски	259
2.1	<Идентификатор риска 1 – описательное имя или номер>	259
2.1.1	Величина или ранг риска	259
2.1.2	Описание	259
2.1.3	Влияние	259
2.1.4	Признаки	260
2.1.5	Стратегия снижения риска	260
2.1.6	План на случай реализации риска	260
2.2	<Идентификатор риска 2 – описательное имя или номер>	260

Ошибка! Неизвестное имя свойства документа.

Перечень рисков

Введение

[Введение представляет собой обзор содержимого документа в целом. Оно включает в себя:

- *цели создания документа;*
- *область применения;*
- *используемые в тексте определения, сокращений и аббревиатуры;*
- *ссылки на другие документы или иные источники информации;*
- *сведения об организации настоящего документа.]*

Цели

[Определите цель создания настоящего документа.]

Область применения

[Краткое определение области применения данного документа; список проектов, связанных с ним, и всё остальное, на что оказывает влияние данный документ.]

Определения, сокращения и аббревиатуры

[Данный подраздел содержит определения всех терминов, сокращений и аббревиатур, необходимых для правильной интерпретации документа. Эта информация может быть представлена в виде ссылок на статьи Глоссария проекта.]

Ссылки

[Данный подраздел содержит полный перечень всех документов, на которые содержится ссылка в Перечне рисков. Для каждого документа указывается название, номер публикации (если это необходимо), дата и организация-издатель. Укажите источники, из которых были получены данные ссылки. Эта информация может быть представлена ссылкой на приложение или другой документ.]

Общие сведения

[В данном подразделе даётся краткое описание содержимого Перечня рисков, объясняется его организация.]

Ошибка! Неизвестное имя свойства документа.

Риски

<Идентификатор риска 1 – описательное имя или номер>

Величина или ранг риска

[Показатель величины риска может быть использован для ранжирования рисков от наиболее опасных для проекта до самых незначительных.]

Описание

[Краткое описание риска.]

Влияние

[Перечислите воздействия, которые риск оказывает на проект или на его продукт.]

Признаки

[Опишите методы, с помощью которых можно наблюдать за состоянием риска, выявлять моменты, когда риск реализовался или близок к реализации. Включите в описание такие параметры как метрики и их граничные величины, результаты тестов, особые события и т.д.]

Стратегия снижения риска

[Опишите, что в настоящий момент сделано в проекте для уменьшения влияния риска.]

План на случай реализации риска

[Опишите образ действий, необходимых в случае реализации риска: альтернативные решения, урезание (сокращение) функциональности и т.д.]

<Идентификатор риска 2 – описательное имя или номер>

Нотация моделей взаимодействия

Диаграмма прецедентов

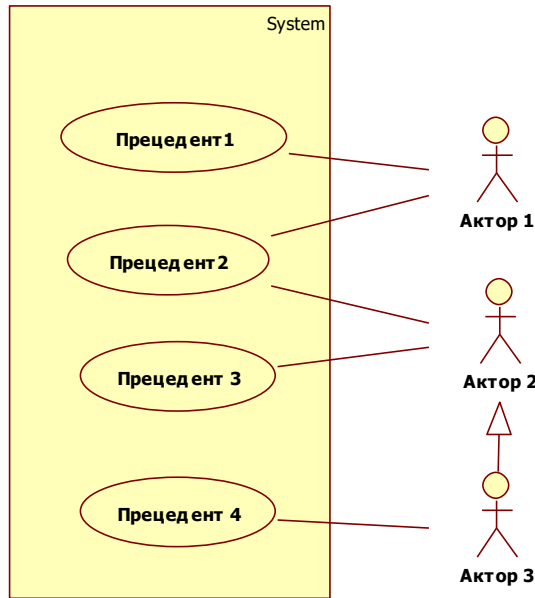
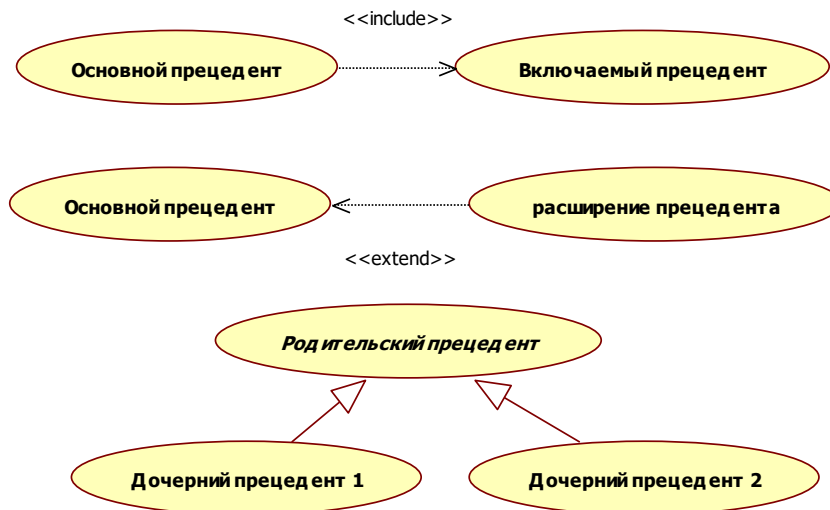


Диаграмма прецедентов (между акторами допускается только отношение генерализации)



Возможные отношения между прецедентами (родительский прецедент рекомендуется делать абстрактным)

Диаграмма последовательности

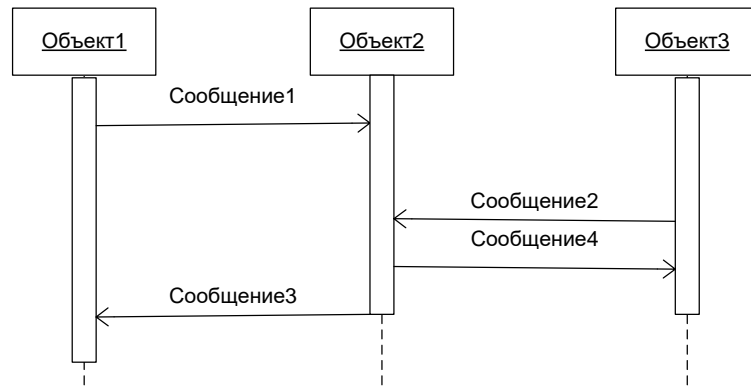


Диаграмма последовательности с асинхронным управлением между активными объектами.

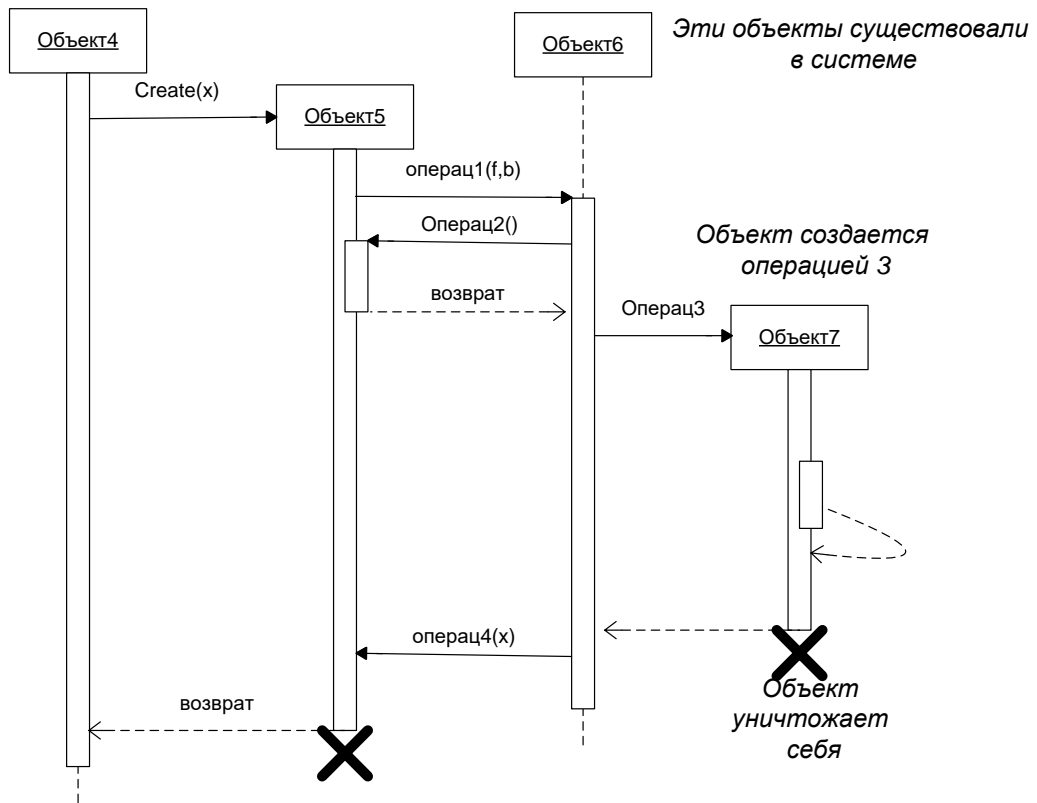


Диаграмма последовательности с процедурными потоками управления

Диаграмма коммуникации

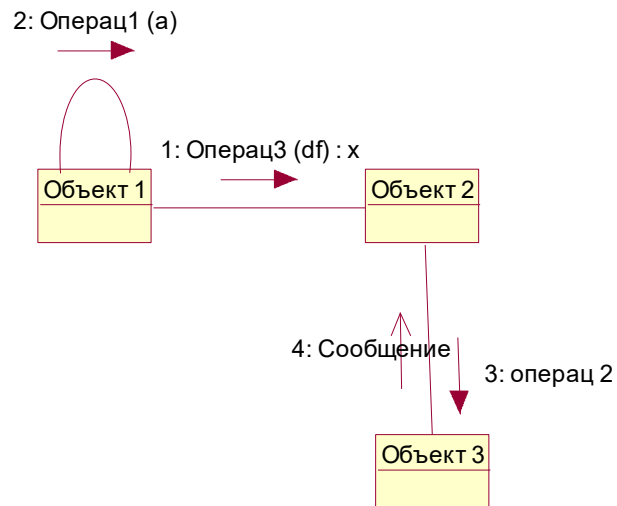


Диаграмма коммуникации.

Диаграмма деятельности

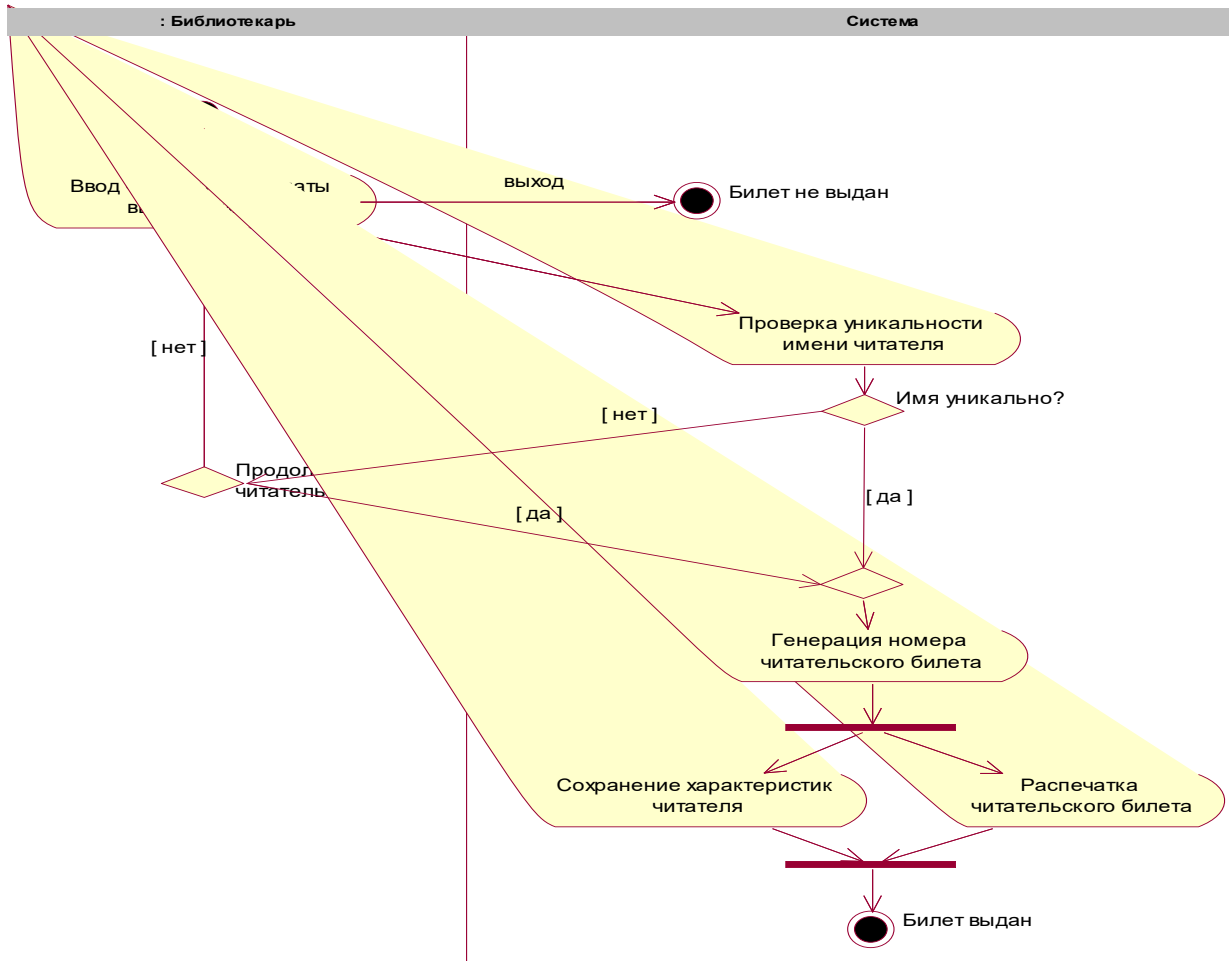
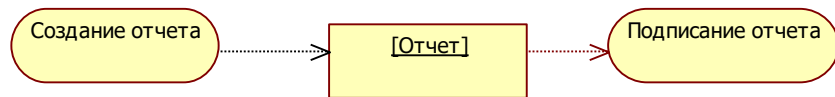
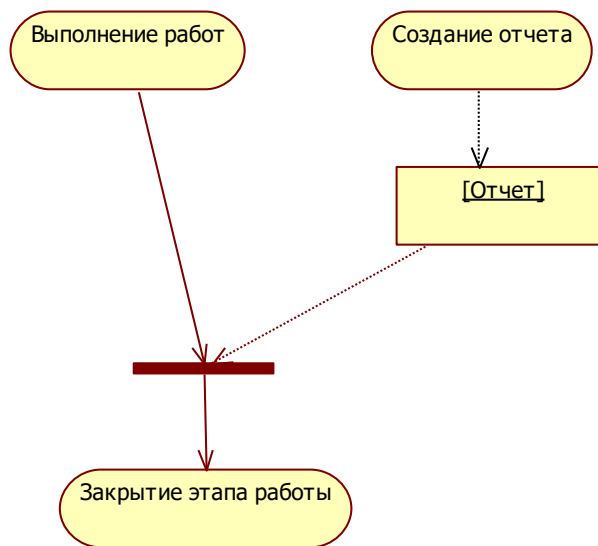


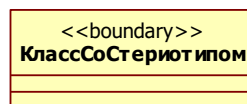
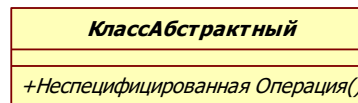
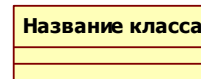
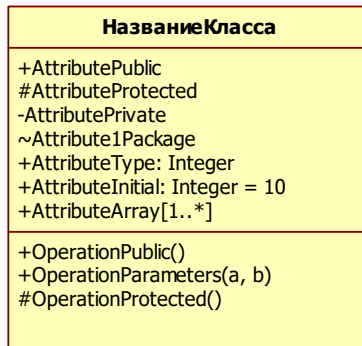
Диаграмма деятельности с разделением ответственности (использование swimlanes- «плавающих дорожек»)





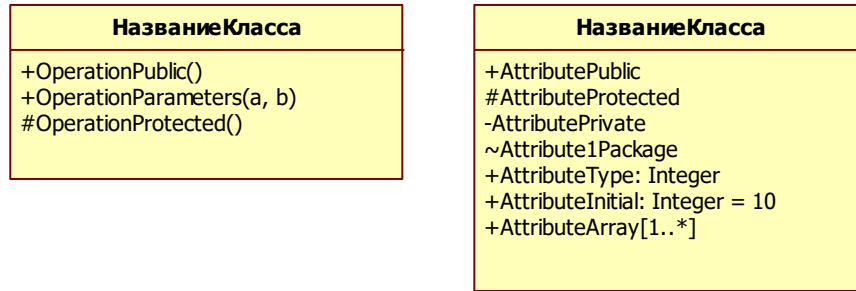
Использование объектных потоков на диаграмме деятельности

Нотация моделей классов

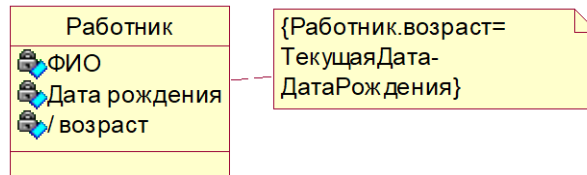


- + видят все классы,
- # видим для класса или его потомка,
- видим только для данного класса,
- ~ видим для любого класса из данного пакета

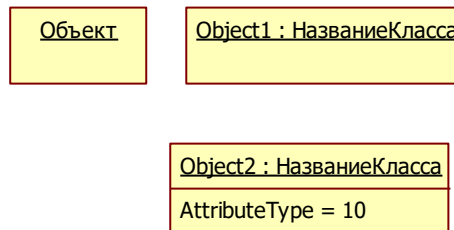
Класс (примеры определения)



Классы, при отображении которых были скрыты атрибуты и операции



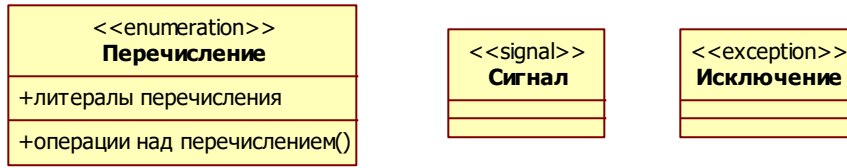
Вычисляемый атрибут (возраст)



Объекты



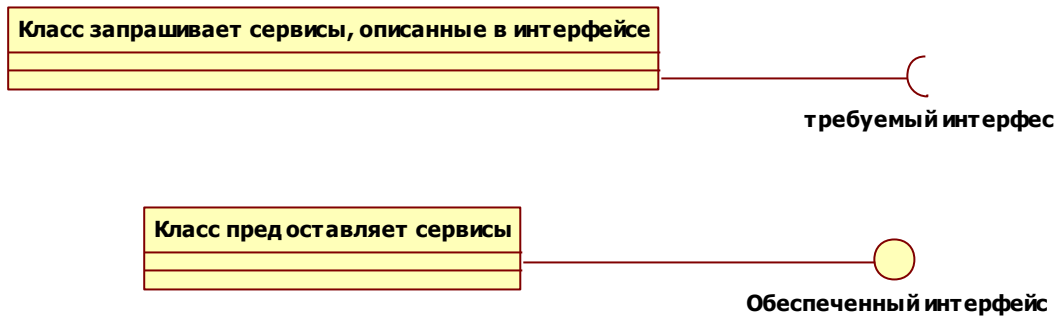
Пакет и подсистема



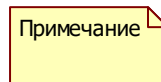
Распространенные стереотипы



Интерфейс

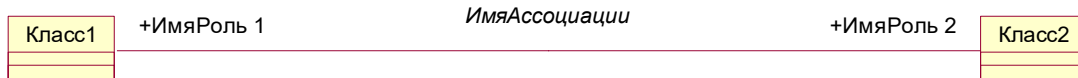


Требуемый и обеспеченный интерфейс

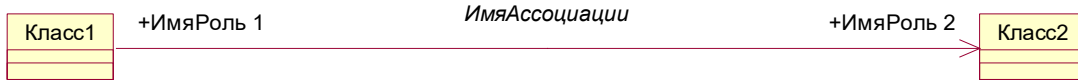


Комментарии

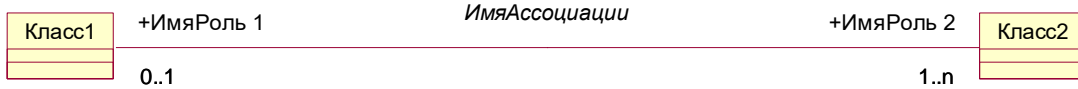
ОТНОШЕНИЯ



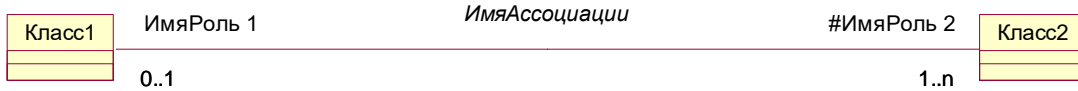
Ассоциация



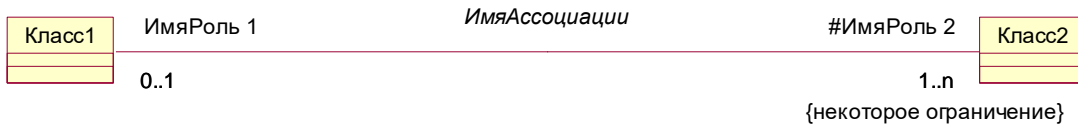
Ассоциация с навигацией



Ассоциация с заданной множественностью



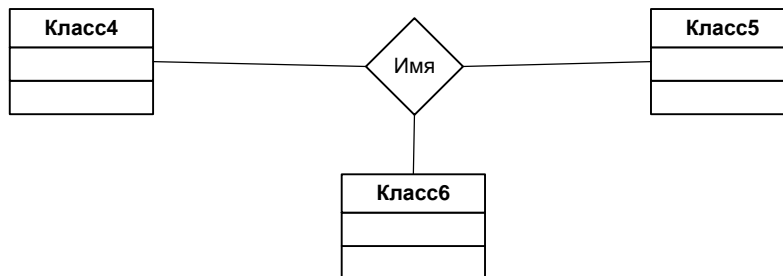
Ассоциация с заданной видимостью полюсов



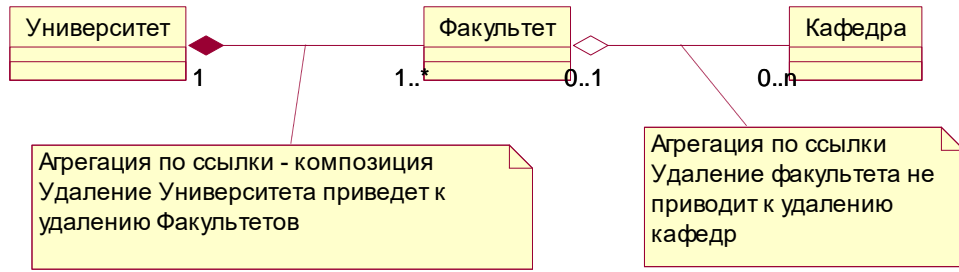
Ассоциация с заданным ограничением



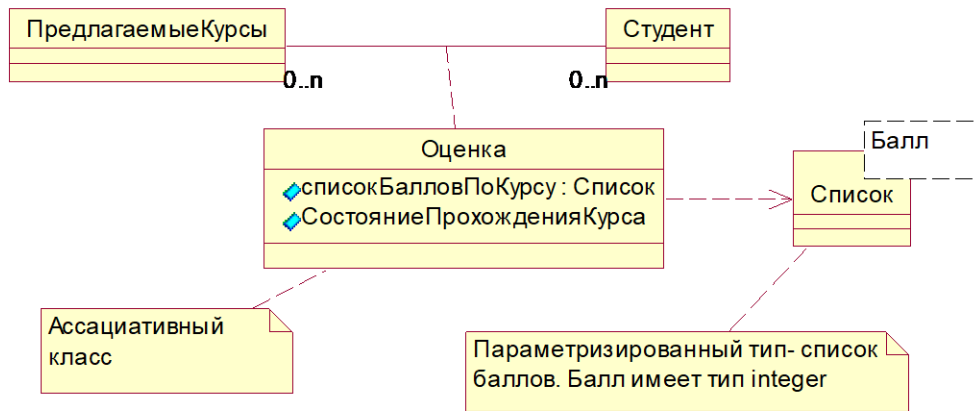
Квалифицированная ассоциация и пример использования



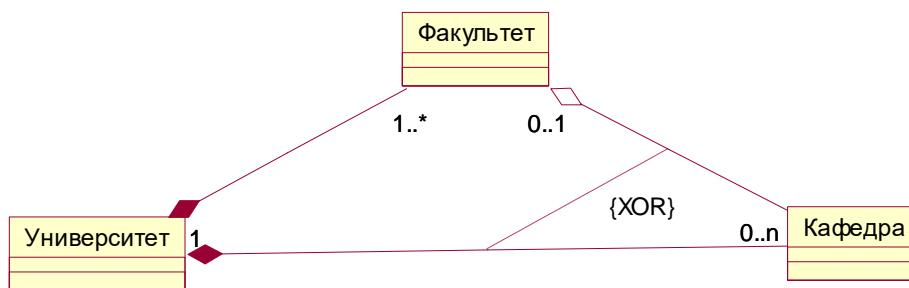
Тернарная ассоциация



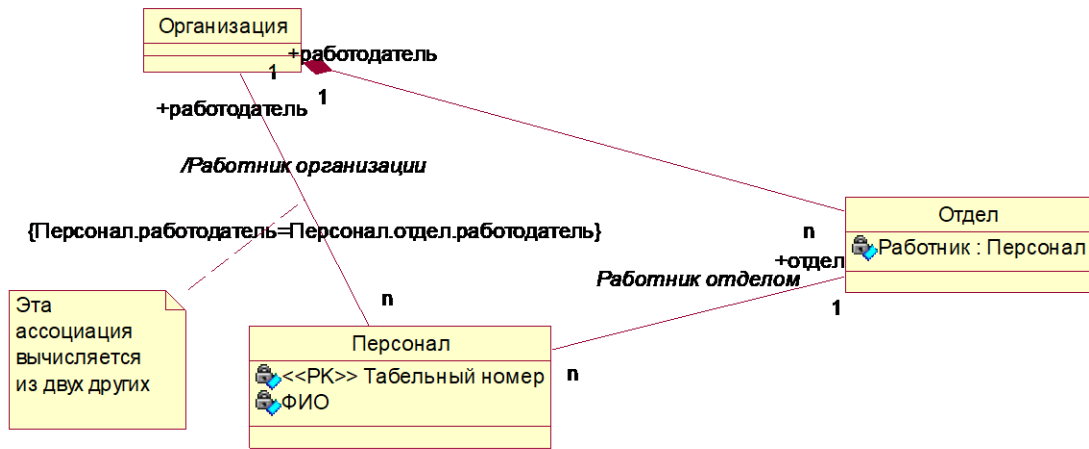
Агрегация и композиция



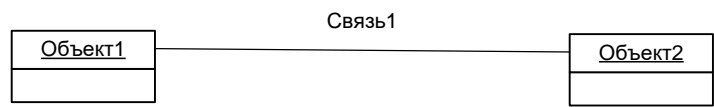
Пример использования ассоциативного класса и параметризованного класса (задан параметризованный тип Список(Балл)).



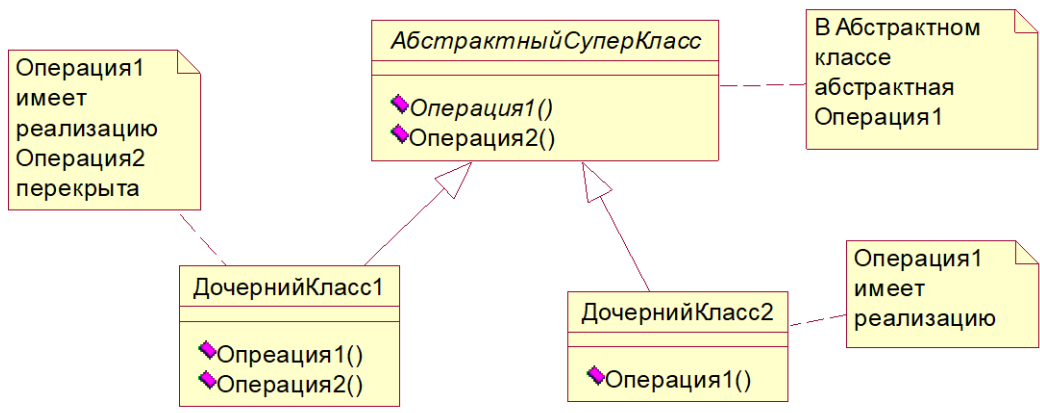
Ограничение между ассоциациями



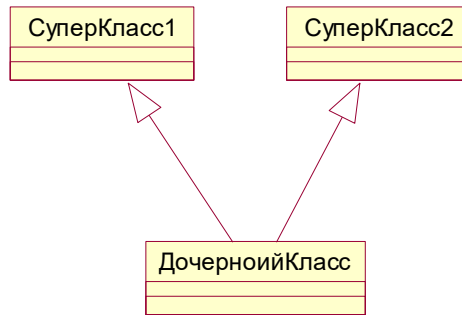
Производная ассоциация



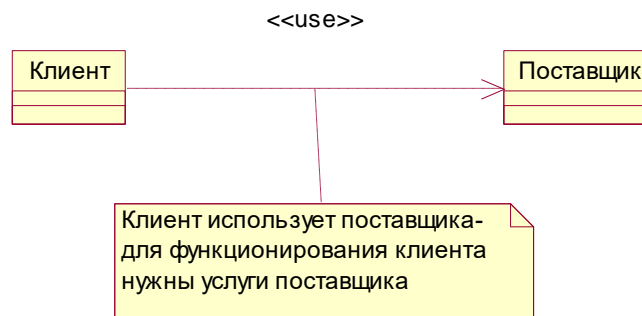
СВЯЗЬ



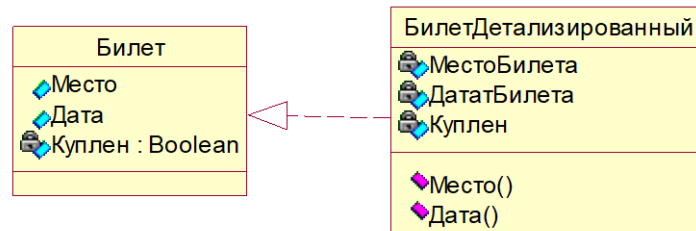
Генерализация и пример применения абстрактного класса



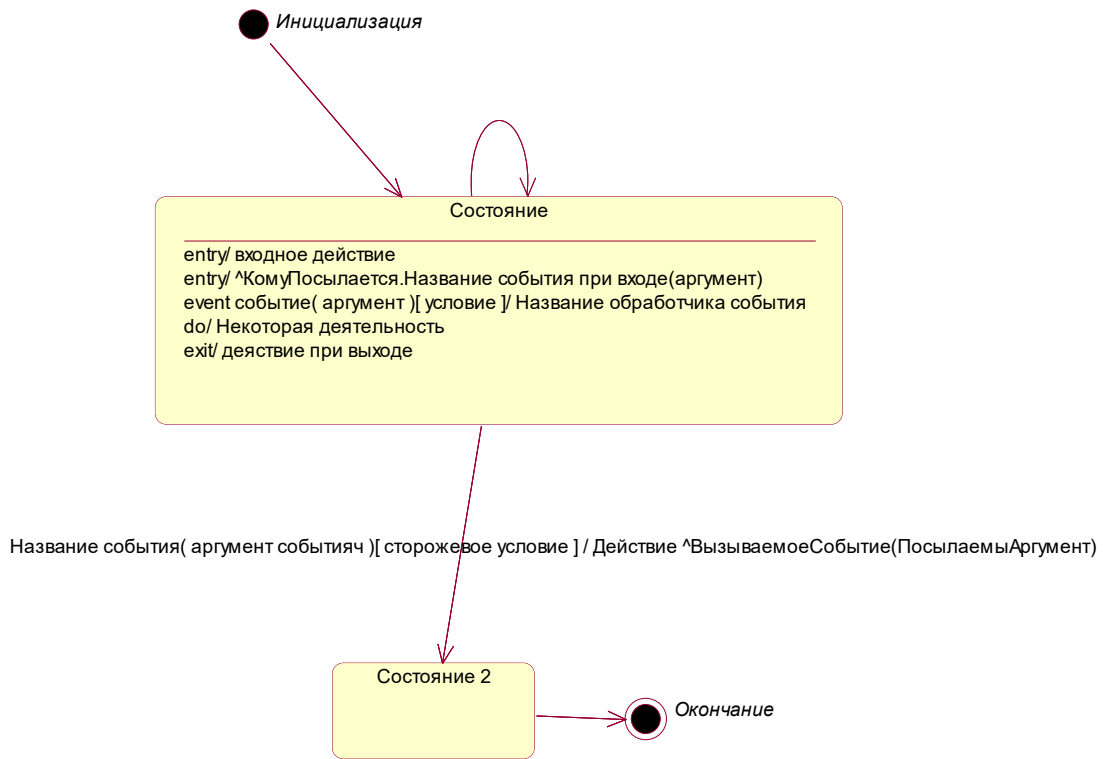
Множественное наследование (Object Pascal не поддерживает)



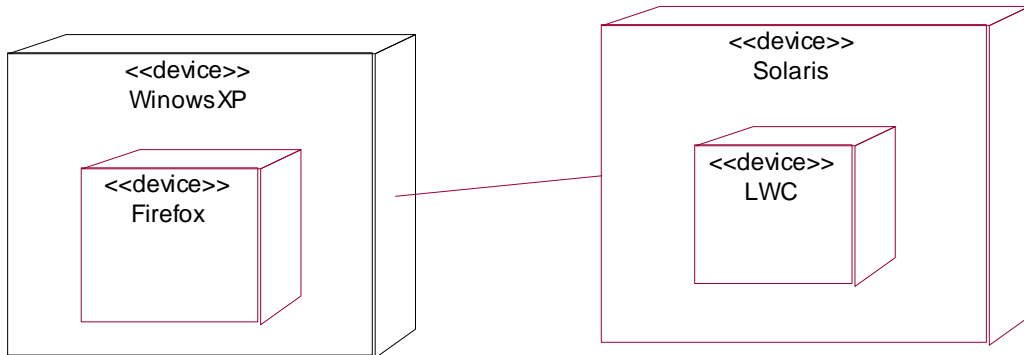
Зависимость со стереотипом <<use>>.



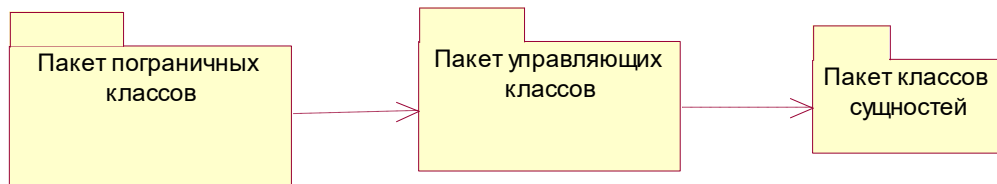
Отношение «Реализация». В данном случае, как может быть реализован класс билет с помощью класса «БилетДетализированный». Применение реализации для интерфейсов показано выше (изображение обеспеченного интерфейса).



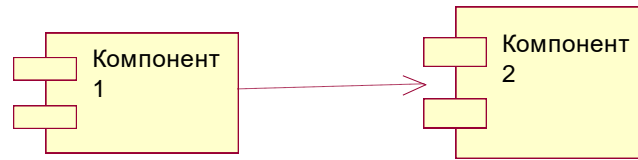
Пример диаграммы состояния, которое моделирует динамическое поведение классов, прецедентов, подсистем и системы в целом



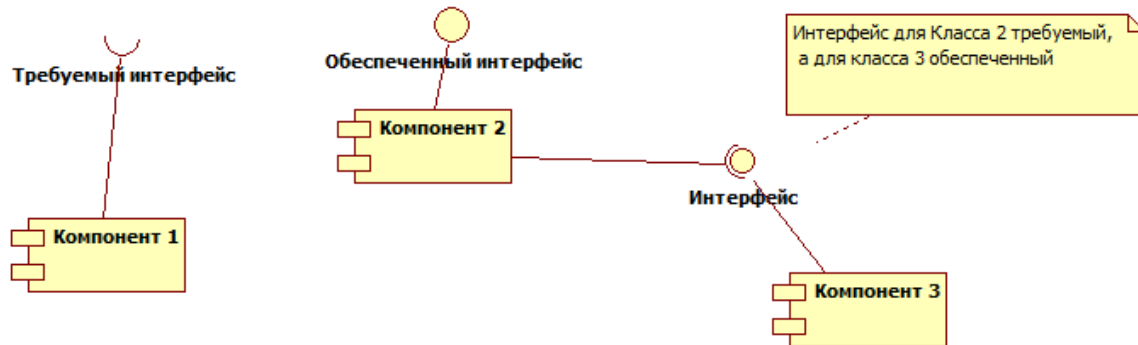
Вариант диаграммы развертывания



Вариант диаграммы пакетов (реализация модели ВСЕ)



Вариант диаграммы компонентов



Вариант диаграммы компонентов

Таблица Д.1- Итог расчета часов и ставок по кафедре

Общий итог нагрузки	
Количество аудиторных часов	
Количество ставок	
Почасовой фонд	

Таблица Д.2 – Планируемый штат по кафедре

Должность	Количество ставок
Профессор	
Доцент	
Старший преподаватель	
Ассистент	

Таблица Д.3 - Планирование штата преподавателей по кафедре

ФИО преподавателя	Должность	Ученая степень	Количество ставок

Таблица Д.4 - Профессорско-преподавательский состав (ставки)

Должность	Профессора	Доценты	Ст. преподаватели	Ассистенты	Итого
Утверждено					
Фактически					
Из них занято совместителями: внешними внутренними					

Таблица Д.5- Учебная нагрузка кафедры

Объем учебных поручений	Всего	Лекций	Практических, лабораторных	Другие виды нагрузки (неаудиторная)
Штатные ППС	п			
	ф			
Совместители	п			
	ф			
Почасовики	п			
	ф			
Итого	п			
	ф			

Таблица Д.6- Распределение годового объема учебных поручений преподавателей кафедры (плановая)

№ п/п	Фамилия, инициалы, должность, степень, ставка	Наименование читаемых курсов	Специальность																					
				Лекции																				
				Практ. и сем. занят.																				
				Лабораторные занятия																				
				Консультации по дисциплине																				
				Консультации перед экзаменом																				
				Консультации по магистерским дисс-ции																				
				Индивидуальные консультации																				
				Зачеты																				
				Семестровые экзамены																				
				Рецензирование контрольных работ																				
				ГЭК и ГЭК																				
				Реценз-е реферата при поступлении в ас- пирантуру																				
				Руководство																				
				уч., произв., и руководство курсовой работой																				
				Руководство дипломной работой																				
				Руководство подготовкой студента в маги- стратуре																				
				Руководство программой спец. подг. в ма- гистр-ре																				
				Руководство аспирант., доктор., соискателем																				
				Всего																				
Итого по кафедре																								

Таблица Д.7 - Распределение годового объема учебных поручений преподавателей кафедры (план/факт)

№ п/п	Фамилия, инициалы, должность, степень, ставка	Наименование читаемых курсов	Специальность	Лекции	Практ. и сем. занятия	Лабораторные занятия	Консультации по дисциплине	Консультации перед экзаменом	Консультации по магистерским дисс-ции	Зачеты	Экзамены	Рецензирование контрольных работ	ГЭК и ГЭК	Рецензирование реферата при поступлении в аспирантуру	Руководство уч., произв., и	Руководство курсовой работой	Руководство дипломной работой	Руководство подготовкой студента в магистратуре	Руководство программой спец. подг. в магистратуре	Руководство аспирант., доктор., соискателем	План	Факт.	
																							План
Итого по кафедре																						План	Факт

Таблица Д.8 - Учебное поручение преподавателю кафедры.

Наименование дисциплин и видов учебной работы по семестрам	Специальность, семестр		Кол-во студентов	Лекции	Практ. и сем. занят.	Лабораторные занятия	Консультации по дисциплине	Консультации перед экзаменом	Консультации по магистерским дисс-ции	Индивидуальные консультации	Зачеты	Семестровые экзамены	Рецензирование контрольных работ	ГАК и ГЭК	Реценз-е реферата при поступлении в аспи-ру	Руководство уч., произв., и пред., практиками	Руководство курсовой работой	Руководство дипломной работой	Руководство подготовкой студента в магистратуре	Руководство программой спец. подг. в магис-ре	Руководство аспирант., доктор., соискателем	Всего
	п	ф																				
Итого																						

Квалификация специалиста:
информатик в области экономики
Срок обучения 5 лет

УЧЕБНЫЙ ПЛАН
Специальность: 080801.65 - ПРИКЛАДНАЯ ИНФОРМАТИКА (в экономике)

УТВЕРЖДАЮ:
Ректор ТюмГУ

_____ Г.Ф.Куцев
" " "

№ п/п	Наименование дисциплин	Распределение по семестрам форм контроля				Объем работы студентов (час.)						Распределение по курсам и семестрам										Кафедра									
						с преподавателем					Самостоятельная	I курс		II курс		III курс		IV курс		V курс											
		экзамен	зачет	к.р./колл.	курсе.пр.	Всего	аудиторные занятия			1 сем		2 сем	3 сем	4 сем	5 сем	6 сем	7 сем	8 сем	9 сем	10 сем											
							лекции	практ.	всего	индив.		Учебных недель в семестре																			
												18	17	18	17	18	17	18	17	13											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
ГСЭ.	Общие гуманитарные и социально-экономические дисциплины																														
Ф.	Федеральный компонент				1260	142	760	902	62	296																					
01	Иностранный язык	4	1,2,3		340		280	280	35	25	4		4		4		4														Ин.языков ЕНФ

Рисунок Д.2 – Вид учебного плана.

Таблица Д.9 - Распределение аудиторной нагрузки для преподавателя.

Специальность	Предмет	Лекции			практика /лаб.		
		потоки	студенты	часы	группы	студенты	часы

Таблица Д.10 – Контингент (на определенную дату).

Факультет и специ- альность	Курсы														
	1			2			3			4			5		
	Кол. потоков	Кол. групп	Кол. студентов	Кол. потоков	Кол. групп	Кол. студентов	Кол. потоков	Кол. групп	Кол. студентов	Кол. потоков	Кол. групп	Кол. студентов	Кол. потоков	Кол. групп	Кол. студентов

Таблица Д.11 – Распределение студентов по специализациям
(оформляется распоряжением по деканату/ институту)

Кафедра	Название спе- циализации	Курсы					
		1		2		3	
		Кол. групп	Кол. студентов	Кол. групп	Кол. студентов	Кол. групп	Кол. студентов

Таблица Д.12 -Нормы времени для расчета объема учебной работы, выполняемой профессорско-преподавательским составом в Тюменском государственном университете

№п.п	Виды работ	Нормы времени в часах для расчета нагрузки	Примечание
1	2	3	4

Входные и выходные документы темы «Информационная система поддержки дистанционного проведения практических и индивидуальных занятий»

Договор № _____

об оказании образовательных услуг по программам,
реализуемым с применением дистанционных технологий

г. Тюмень

« » _____ 200__ г.

ГОУ ВПО «Тюменский государственный университет», именуемый в дальнейшем «ВУЗ» или ТюмГУ, в лице _____ с одной стороны, и

_____, именуемый(ая) в дальнейшем «Преподаватель», с другой стороны, при совместном упоминании именуемые Стороны, заключили настоящий Договор о нижеследующем.

1. Предмет Договора

1.1. Преподаватель принимает на себя обязательства по заданиям **ВУЗа** и в соответствии с учебным рабочим планом, графиком учебного процесса, индивидуальным планом работ и расписанием занятий, оказывать следующие услуги:

- проверка контрольных работ;
- проверка курсовых работ;
- руководство преддипломной практикой;
- руководство выпускными квалификационными работами;
- проверка тестовых заданий, выполняемых студентами на бумажных носителях,
- проведение аудиторных занятий (лекционных, семинарских, практических, лабораторных);
- проведение промежуточной аттестации;
- работа в государственной экзаменационной комиссии и государственной аттестационной комиссии;
- разработка тестовых заданий для промежуточной аттестации;
- предоставление конспектов лекций на электронном носителе.

1.2. ВУЗ обязуется известить **Преподавателя** о потребности в конкретных услугах, из числа перечисленных в п. 1.1 договора, принять и оплатить оказанные **Преподавателем** услуги.

Рисунок Е.1 – Фрагмент договора на оказание услуг

АКТ ПРИЕМА-ПЕРЕДАЧИ (ПРЕДОСТАВЛЕНИЯ) УСЛУГ

к договору № _____ от «___» _____ 200__ г.

**об оказании образовательных услуг по программам,
реализуемым с применением дистанционных технологий**

г. Тюмень

«___» _____ 200__ г.

ГОУ ВПО «Тюменский государственный университет» (далее ТюмГУ), в лице директора Института дистанционного образования ТюмГУ _____, с одной стороны, и _____, ученая степень _____, ученое звание _____, основное место работы _____, паспорт серия: _____ номер _____, выдан _____, именуемый(ая) в дальнейшем «Преподаватель», с другой стороны, подписали настоящий Акт о нижеследующем:

1. В соответствии с условиями Договора Преподаватель оказал ТюмГУ следующие услуги по программам, реализуемым с применением дистанционных образовательных технологий:

Рисунок Е.2- Фрагмент акта приема- передачи услуг

Таблица Е.1 - Описание процесса

Процесс	Роль	Вход	Выход	Сроки
Поиск преподавателя	Специалист ОУУП ИДО	Телефонный справочник кафедр Рабочий учебный план (дисциплины семестра) Бланк договора с преподавателем	Договор с преподавателем	2 месяца до сессии
Регистрация тем работ	Специалист ОУУП ИДО	Список тем по вариантам Журнал успеваемости работ	Принцип выбора варианта Файлы вариантов работ Заполненный журнал успеваемости работ с указанием вариантов	
Проверка оформления работ	Специалист ОУУП ИДО	Работы студентов Журнал успеваемости (ЖУ) работ на дисциплину	Проверенная на оформление работы студентов Сообщение студентам о некорректно оформленных работах Отметка в ЖУ работ о приеме работы студента	2 недели
Доставка работ преподавателю	Специалист ОУУП ИДО	Бумажные работы потока студентов Бланк экзаменационной ведомости на работу Электронные варианты работ	Отметка в ЖУ работ о приеме преподавателем работы	
Проверка работ	Преподаватель	Бумажные работы потока студентов Электронные варианты работ	Исправленные работы потока студентов Электронные письма студентам с замечаниями	
Сдача	Преподаватель	Бумажные корректные работы студентов Заполненная экзаменационная ведомость на работу ЖУ работ на дисциплину	Акт оплаты с количеством проверенных работ Подписанная экзаменационная ведомость Отметка в ЖУ работ о сдаче дисциплины	За месяц до начала сессии

Журнал успеваемости контрольных работ

Год поступления
Образовательная программа

Дисциплина УП " _____ ", _____ семестр, преподаватель _____

№ п/п	ФИО	Номер зачетки	Название темы	Дата сдачи	Роспись студента	Дата приема преподавателем	Роспись преп-ля	Дата сдачи	Оценка	Роспись преподавателя

Специалист ОУУП ИДО _____ / _____

Рисунок Е.2 – Журнал успеваемости контрольных работ