

**Государственное бюджетное образовательное учреждение
высшего образования Московской области
«Университет «Дубна»
Филиал «Протвино»
Кафедра «Информационные технологии»**

В. В. Черноверская, Т. Н. Кульман

**Подготовка и оформление курсовых работ по дисциплине
«Проектирование информационных систем»**

Электронное методическое пособие

Рекомендовано
кафедрой информационных технологий
филиала «Протвино» государственного университета «Дубна»
в качестве методического пособия для студентов,
обучающихся по направлению
«Прикладная информатика»

Протвино
2017

ББК 65с51 я 73
Ч49

Рецензент:
кандидат физико-математических наук,
главный специалист ООО «Систел»
Е.В. Клименков

Черноверская, В.В.

Ч49 Подготовка и оформление курсовых работ по дисциплине «Проектирование информационных систем»: электронное методическое пособие / В.В. Черноверская, Т.Н. Кульман. — Протвино: 2017. — 27с.

Предназначено для студентов, обучающихся по направлению. «Прикладная информатика». Пособие устанавливает единые требования к содержанию, объему и правилам оформления курсовых работ по дисциплине «Проектирование информационных систем» для студентов очной и заочной форм обучения. Выполнение требований обязательно для всех преподавателей кафедры «Информационные технологии», осуществляющих руководство курсовыми работами, и для всех студентов, выполняющих курсовые работы.

ББК 65с51 я 73

© Государственное бюджетное образовательное учреждение высшего образования Московской области «Университет «Дубна», филиал «Протвино», 2017
© Черноверская В.В., Кульман Т.Н.

1 Задачи и требования к курсовой работе

Курсовая работа призвана выявить знания студентов по проектированию экономических информационных систем (ЭИС) с использованием технологий объектно-ориентированного подхода и автоматизированного проектирования современных информационных систем. Тематика курсовой работы предполагает разработку ЭИС для решения конкретных прикладных задач выбранной предметной области.

В процессе курсового проектирования студент должен проявить навыки к самостоятельной работе с научно-технической литературой, к обобщению накопленного опыта и умение делать научно-обоснованные выводы и рекомендации.

Целью курсового проектирования является:

- закрепление и развитие теоретических знаний, полученных студентами в процессе изучения курса “Проектирование экономических информационных систем” и смежных с ним курсов, изучаемых на кафедре;

- развитие умения осуществлять выбор варианта технологии проектирования ЭИС и соответствующих ей инструментальных средств проектирования;

- приобретение студентами практических навыков разработки методических материалов проектировщика с помощью построения технологических сетей проектирования на базе использования выбранных средств;

- умение вырабатывать и реализовывать решения при выполнении операций проектирования всех компонентов проекта ЭИС, включая информационное, программное, и технологическое обеспечение ЭИС, ориентированное на конкретную техническую платформу;

- закрепление умения выполнять технико-экономическое обоснование целесообразности предложенного варианта проектирования.

Основная задача курсовой работы — это построение моделей программного обеспечения (ПО) с помощью инструментального CASE-средства *IBM Rational Rose*. В качестве альтернативного варианта может использоваться свободно распространяемый программный продукт *StarUML* со схожим функционалом. Пояснительная записка должна быть оформлена в соответствии с требованиями к оформлению технической документации с применением актуальных офисных приложений.

2 Организация курсового проектирования

Студент выбирает тему курсовой работы в соответствии с рекомендуемой кафедрой тематикой.

Срок выбора темы и период написания курсовой работы определяется графиком работы студентов, утвержденным учебной частью, и уточняется кафедрой. Студенту предоставляется право свободного выбора темы.

Для руководства курсовым проектированием студенту выделяется руководитель из числа преподавателей кафедры. После выбора темы и утверждения ее на кафедре студент должен подобрать и изучить литературу, составить предварительный план, используя при этом материал настоящего методического пособия, а также список рекомендуемой литературы. Предварительный план обсуждается с руководителем, после чего руководитель выдает студенту задание, в котором указываются:

- **предметная область** (систему, подсистему, комплекс задач, задачу), которую можно выбрать самостоятельно;
- **цель проектирования** — разработку моделей программного обеспечения для выбранной предметной области;
- **инструментальное средство**;
- **исходные данные**, необходимые для разработки.

В соответствии с заданием студент составляет уточненный план работы и приступает к его выполнению.

В процессе работы по каждой теме могут быть найдены различные проектные решения; студент должен самостоятельно выявить их, показать отрицательные и положительные стороны каждого из них и обосновать свой выбор. Руководитель обязан оказывать методическую и научную помощь, систематически контролируя ход курсового проектирования.

Законченная курсовая работа демонстрируется руководителю на ЭВМ, а после исправления полученных замечаний и повторной демонстрации — оформляется в соответствии с требованиями настоящего методического пособия и передается руководителю на проверку.

После проверки курсовая работа защищается перед комиссией, назначенной кафедрой. На защиту курсовой работы студенту необходимо подготовить презентацию в среде *MS PowerPoint*, в которой должны быть отражены тема и цели проектирования, постановка задачи, все графические материалы: (структура разрабатываемого ПО, различного рода диаграммы, таблицы, графики и

т. д.), а также основные выводы по проведенной работе. В процессе защиты курсовой работы нужно кратко изложить поставленную задачу, методику ее решения, полученные результаты и сделать необходимые выводы.

Сдача выполненной работы руководителю и ее защита проводятся строго в сроки, установленные каждому студенту в задании по курсовому проектированию.

3 Структура и оформление курсовой работы

Курсовая работа должна состоять из четырех глав, заключения и приложения.

Первая глава — «Постановка задачи» должна содержать формулировку задания.

Вторая глава — «Анализ требований» должна содержать глоссарий, диаграмму вариантов использования, описания действующих лиц и вариантов использования.

Третья глава — «Анализ системы» должна содержать диаграммы взаимодействия между объектами (последовательности и кооперативные), соответствующие потокам событий вариантов использования. При необходимости можно включить диаграммы деятельности и сопроводить их пояснениями, указывающими, какому потоку событий они соответствуют (если это не ясно из их названия), и комментариями.

Четвертая глава — «Проектирование» должна содержать иерархию классов системы и описание пакетов. Для каждого класса системы дается описание, которое включает: ответственность класса, описание атрибутов в виде таблицы из трех столбцов: имя, описание, тип; таблицу с описанием операций (имя, описание, сигнатура). Должны быть приведены диаграммы классов системы, отображающие связи между классами, и диаграммы состояний, описывающие поведение экземпляров отдельных классов. Также приводится диаграмма размещения с комментариями. Если вариант предполагает создание схемы базы данных, то такая схема также должна быть включена в отчет.

Процесс создания модели включает несколько

этапов. Этап 1. Составление глоссария.

Этап 2. Создание модели вариантов использования.

Этап 3. Анализ вариантов использования.

Этап 4. Проектирование системы.

Структура модели в браузере *Rose* должна соответствовать структуре, предусмотренной технологией *Rational Unified Process*.

После выполнения третьего этапа модель должна удовлетворять следующим требованиям:

- глоссарий проекта должен быть представлен в виде таблицы и храниться в отдельном файле;
- на диаграммах вариантов использования каждое действующее лицо и вариант использования должны сопровождаться описанием.

Описание действующего лица кратко (в одну-две строки) характеризует роль данного лица. Описание варианта использования включает пояснение, предусловие, потоки событий (основной и альтернативные, если таковые имеются) и постусловие. Описания представляют собой либо присоединенные текстовые файлы, либо текст, введенный в поле *Documentation* спецификации соответствующего элемента диаграммы;

- диаграммы взаимодействия, соответствующие потокам событий вариантов использования, должны содержать необходимые пояснения.

При проектировании системы требуется:

- создать иерархию классов системы;
- разместить классы по пакетам (в зависимости от постановки задачи);
- связать объекты с классами, сообщения на диаграммах взаимодействия – с операциями;
- сопроводить кратким описанием каждый класс (обязанности класса), с представлением атрибутов в виде таблицы (имя, описание, тип) и описанием операций (имя, описание, сигнатура);
- указать стереотипы для классов;
- построить диаграммы классов системы, отображающие связи между классами;
- построить диаграммы состояний для описания поведения объектов отдельных классов;
- разработать (если требуется) схему базы данных и отобразить ее на диаграмме «сущность-связь».

Следует также разработать диаграмму размещения. В зависимости от варианта задания, диаграмма размещения должна показывать расположение компонентов в распределенном приложении.

Заключение должно содержать оценку полученных результатов и изложение основных направлений дальнейшего совершенствования проекта для данного элемента ЭИС.

В приложении должен быть представлен в распечатанном виде программный код, разработанный с использованием выбранного инструментального средства (в объеме одного модуля).

К проекту прилагается список используемой литературы.

4 Методика проектирования ИС с применением языка моделирования *UML*

Язык *UML* обеспечивает поддержку всех этапов жизненно-го цикла ИС и предоставляет для этих целей ряд графических средств – диаграмм. На этапе создания *концептуальной модели* для описания деятельности используются *модели прецедентов* и диаграммы видов деятельности, для описания объектов – *модели объектов* и диаграммы последовательностей.

На этапе создания логической модели ИС описание требований к системе задается в виде модели и описания системных прецедентов, а предварительное проектирование осуществляется с использованием диаграмм классов, диаграмм последовательностей и диаграмм состояний.

На этапе создания физической модели детальное проектирование выполняется с использованием диаграмм классов, диаграмм компонентов, диаграмм развертывания.

Диаграммы прецедентов (диаграммы вариантов использования, *use case diagrams*) — это обобщенная модель функционирования системы в окружающей среде.

Диаграммы видов деятельности (диаграммы деятельностей, *activity diagrams*) — модель бизнес-процесса или поведения системы в рамках прецедента.

Диаграммы взаимодействия (*interaction diagrams*) — модель процесса обмена сообщениями между объектами, представляется в виде диаграмм последовательностей (*sequence diagrams*) или кооперативных диаграмм (*collaboration diagrams*).

Диаграммы состояний (*statechart diagrams*) — модель динамического поведения системы и ее компонентов при переходе из одного состояния в другое.

Диаграммы классов (*class diagrams*) — логическая модель базовой структуры системы, отражает статическую структуру системы и связи между ее элементами.

Диаграммы базы данных (*database diagrams*) — модель структуры базы данных, отображает таблицы, столбцы, ограничения и т. п.

Диаграммы компонентов (*component diagrams*) — модель иерархии подсистем, отражает физическое размещение баз данных, приложений и интерфейсов ИС.

Диаграммы развертывания (диаграммы размещения, *deployment diagrams*) — модель физической архитектуры системы, отображает аппаратную конфигурацию ИС.

На рис. 1 показаны отношения между различными видами диаграмм *UML*. Указатели стрелок можно интерпретировать как отношение «является источником входных данных для...» (например, диаграмма прецедентов является источником данных для диаграмм видов деятельности и последовательности). Приведенная схема является наглядной иллюстрацией итеративного характера разработки моделей с использованием *UML*.

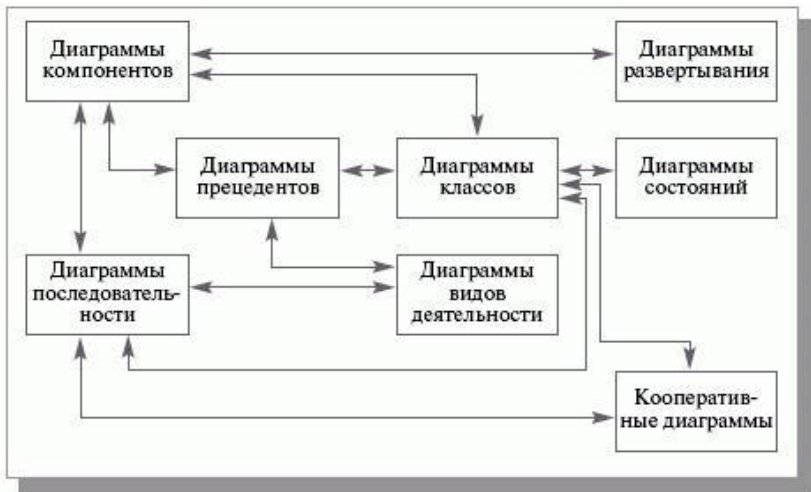


Рис. 1 Взаимосвязи между диаграммами *UML*

Ниже приводятся описания последовательных этапов проектирования ИС с использованием *UML*.

4.1 Разработка модели прецедентов

Модель прецедентов описывает процессы с точки зрения внешнего пользователя, т.е. отражает взгляд на деятельность организации извне. Проектирование системы начинается с изучения и моделирования деятельности организации. На этом этапе вводятся и отображаются в модели ряд понятий, свойственных объектно-ориентированному подходу:

Исполнитель (Действующее лицо, *Actor*) — личность, организация или система, взаимодействующая с информационной системой (ИС); различают внешнего исполнителя (который использует или используется системой, то есть порождает прецеденты деятельности) и внутреннего исполнителя (который обеспечивает реализацию прецедентов деятельности внутри системы). На диаграмме исполнитель представляется стилизованной фигуркой человека.

Прецедент — законченная последовательность действий, инициированная внешним объектом (личностью или системой), которая взаимодействует с ИС и получает в результате некоторое сообщение от ИС. На диаграмме представляется овалом с надписью, отражающей содержание действия.

Класс — описание совокупности однородных объектов с их атрибутами, операциями, отношениями и семантикой. На диаграмме представляется прямоугольником, содержащим описание атрибутов и операций класса.

Ассоциация — связь между двумя элементами модели. На диаграмме представляется линией.

Обобщение — связь между двумя элементами модели, когда один элемент (подкласс) является частным случаем другого элемента (суперкласса). На диаграмме представляется стрелкой-треугольником.

Агрегация — отношение между элементами модели, когда один элемент является частью другого элемента (агрегата). На диаграмме представляется стрелкой с ромбовидным концом.

В качестве примера в данном методическом пособии рассмотрен проект ИС медицинского центра. Назначение ИС — автоматизация ведения и использования клинических записей о пациентах. В настоящее время эта работа производится вручную персоналом центра. На рис. 2 представлена общая модель деятельности центра в виде диаграммы прецедентов.

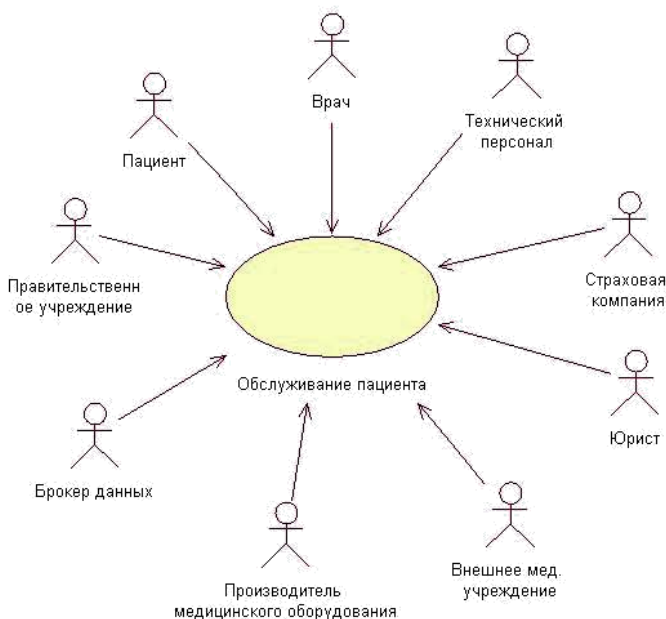


Рис. 2 Общая диаграмма деятельности медицинского центра по обслуживанию пациента

Прецедент «Обслуживание пациента» реализуется через множество других, более ограниченных прецедентов (рис. 3), отражающих детализацию представления функционирования центра.

Для включения в диаграмму выбранные прецеденты должны удовлетворять следующим критериям:

- прецедент описывает, *что* нужно делать, а не *как*;
- прецедент представляет действия с точки зрения исполнителя;
- прецедент возвращает исполнителю некоторое сообщение;
- последовательность действий внутри прецедента представляет собой одну неделимую цепочку.

Исходя из цели создания системы, для дальнейшего исследования и моделирования отбираются только те прецеденты, которые связаны с использованием клинических записей.

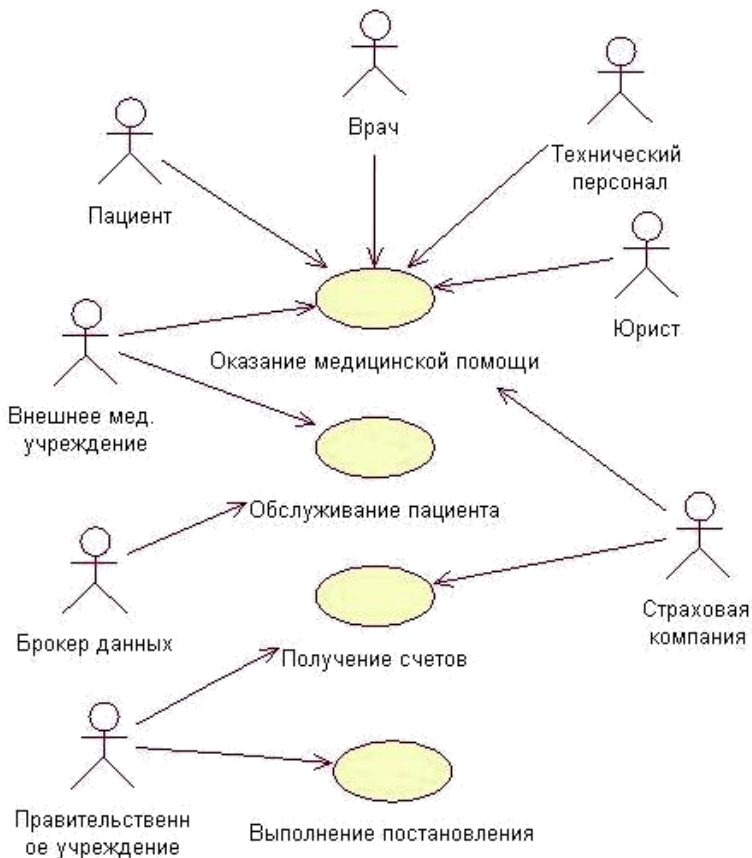


Рис. 3 Модель прецедентов, составляющих обслуживание пациента

Выполнение прецедента описывается с помощью диаграмм видов деятельности, которые отображают исполнителей и последовательность выполнения соответствующих процессов (рис. 4).

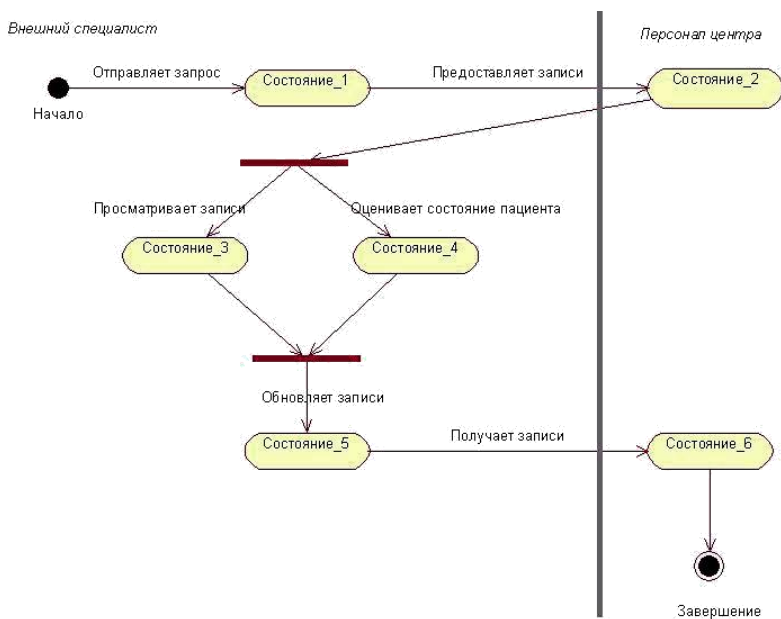


Рис. 4 Диаграмма видов деятельности для прецедента «Оказание медицинской помощи»

Несмотря на то, что оказание медицинской помощи предусматривает множество разнообразных действий исполнителей, для нашей задачи существенными являются только процессы обмена информацией между этими исполнителями, и именно они отображаются в создаваемых моделях. Поэтому на диаграмме отражен процесс оценки состояния пациента на основании имеющейся в центре информации о нем.

Общее поле диаграммы деятельности делится на несколько «дорожек», каждая из которых содержит описание действий одно-го из исполнителей. Основными элементами диаграмм видов деятельности являются обозначения состояния («начало», «конец»), действия (овал) и момента синхронизации действий (линейка синхронизации, на которой сходятся или разветвляются несколько стрелок). Диаграмма подходит для описания действий как внешнего, так и внутреннего специалиста центра.

Этап завершается после разработки диаграмм видов деятельности для всех выделенных в модели прецедентов. Естественно, на последующих этапах анализа и проектирования будут выявлены какие-то важные подробности в описании деятельности объ-

екта автоматизации. Поэтому разработанные на данном этапе модели будут еще неоднократно корректироваться.

4.2 Выявление обобщающих классов

Следующим этапом проектирования ИС является выявление обобщающих классов. Понятие «*обобщение*» (*generalization* — это отношение между более общим классом (родителем или предком) и более специальным классом (дочерним или потомком). Согласно принципу наследования, класс-потомок обладает всеми свойствами и поведением класса-предка, а также может иметь дополнительные свойства и поведение, которые отсутствуют у класса-предка. На диаграмме отношение обобщения обозначается сплошной линией со стрелкой в форме не закрашенного треугольника на одном из концов. Отношение обобщения обеспечивает наследование атрибутов и операций классов.

В диаграмме, представленной на рис. 5, появилось новое действующее лицо — Отправитель запроса. С запросом о состоянии пациента могут обращаться в систему многие из действующих лиц: юрист, страховая компания, технический персонал и даже сам пациент. Поэтому понятие «Отправитель запроса» служит для обобщенного представления всех этих действующих лиц при описании прецедента «Ответ на запрос». «Отправитель запроса» становится обобщающим (суперклассом) по отношению к обобщаемым понятиям (подклассам).

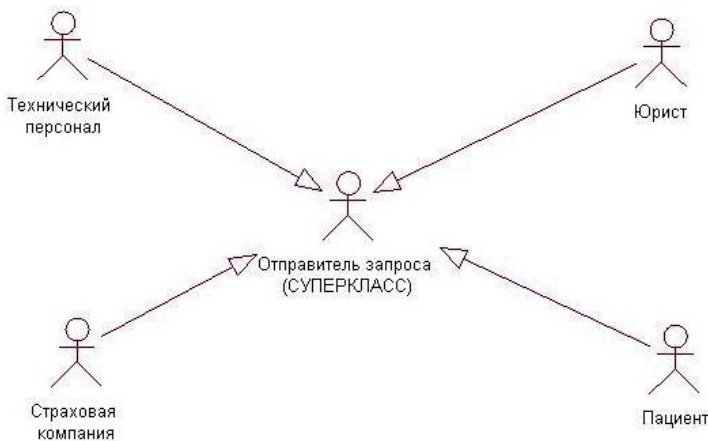


Рис. 5 Обобщение классов

Для детального описания выполнения процессов используются диаграммы последовательностей (рис. 6).

Основными элементами диаграммы последовательностей являются обозначения объектов (прямоугольники), вертикальные линии, отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами.

Результатом этого этапа являются согласованные с заказчиком и достаточно подробные описания действий специалистов организации, внедряющей ИС, необходимые для обеспечения исполнения ее функций.

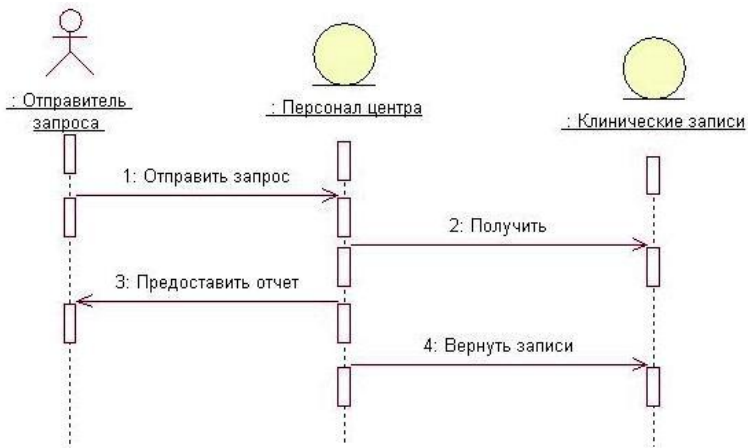


Рис. 6 Диаграмма последовательностей для прецедента «Ответ на запрос»

4.3 Разработка концептуальной модели данных

На основе информации, выявленной на этапах моделирования, выполняется разработка *концептуальной модели данных*, которые будут использоваться в разрабатываемой системе. На рис. 7 представлена, в виде диаграммы классов, модель данных для объекта «Клинические записи».

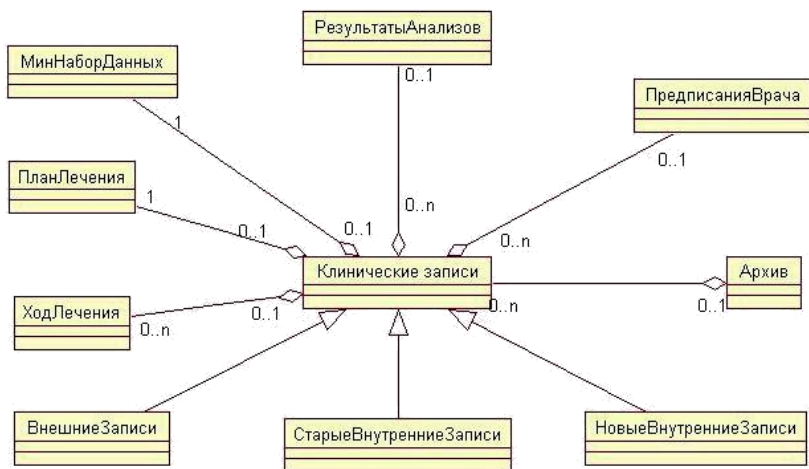


Рис. 7 Концептуальная модель данных

Модель показывает, что клинические записи включают (агрегируют) ряд блоков. При этом «минимальный набор данных» и «план лечения» могут быть включены в каждую клиническую запись в единственном экземпляре, а блоки «результаты анализов», «предписания врача», «ход лечения» могут повторяться неограниченное число раз. Архив состоит из множества клинических записей (агрегирует клинические записи), но может быть и пустым. Поскольку пациент может предварительно проходить лечение в других учреждениях, или несколько раз проходить лечение в центре, появляются дополнительные разновидности (подклассы) клинических записей: внешние, старые внутренние, новые внутренние.

Этот этап завершает процедуры моделирования и позволяет представить команде проектировщиков в едином формате ту информацию, которая будет необходима для создания системы. Разработанные диаграммы являются отправной точкой в процессах проектирования баз данных и приложений системы, обеспечивают согласованность действий аналитиков и разработчиков в процессе дальнейшей работы над системой. Эти диаграммы, конечно же, будут претерпевать изменения в процессе последующего проектирования, однако эти изменения будут фиксироваться в формате, уже привычном для всей команды разработчиков, и будут автоматически отражаться в последующих моделях

4.4 Разработка требований к системе

На этапе формирования требований, прежде всего, необходимо определить область действия разрабатываемой системы и получить точное представление о желаемых возможностях системы.

Основой разработки требований является *модель системных прецедентов*, отражающая выполнение конкретных обязанностей внутренними и внешними исполнителями с использованием ИС.

Источником данных для создания *модели системных прецедентов* являются разработанные на предыдущем этапе модели. Однако при создании модели полезно предварительно составить детальные описания прецедентов, содержащие определения используемых данных и точную последовательность их выполнения. Описание осуществляется в соответствии с принятым в организации шаблоном, который обычно включает следующие разделы:

- заголовок (название прецедента, ответственный за исполнение, дата создания шаблона/внесения изменений);
- краткое описание прецедента;
- ограничения;
- предусловия (необходимое состояние системы или условия, при которых должен выполняться прецедент);
- постусловия (возможные состояния системы после выполнения прецедента);
- предположения;
- основная последовательность действий;
- альтернативные последовательности действий и условия, их инициирующие;
- точки расширения и включения прецедентов.

В процессе создания *модели системных прецедентов* осуществляется преобразование и перенос компонентов моделей на новые диаграммы. Типовые преобразования по технологии *Rational Unified Process* приведены в табл. 1.

Таблица 1

Элементы бизнес-модели	Элементы <i>модели системных прецедентов</i>
Бизнес-прецеденты	Подсистемы
Внешние исполнители	Исполнители
Внутренние исполнители	Исполнители или прецеденты
Процессы, выполняемые внутренними исполнителями	Прецеденты

На рис. 8 представлена модель системных прецедентов для прецедента «Оказание медицинской помощи». Исходя из цели создания системы, в модели системных прецедентов отражены только те действия исполнителей, которые связаны с предоставлением доступа и обновлением клинических записей.

Описываемые моделью функции характерны только для одного вида деятельности – оказания медицинской помощи, и в основном не используются в других видах деятельности Центра. Это позволяет объединить выделенные функции в некую единую под-систему проектируемой ИС.

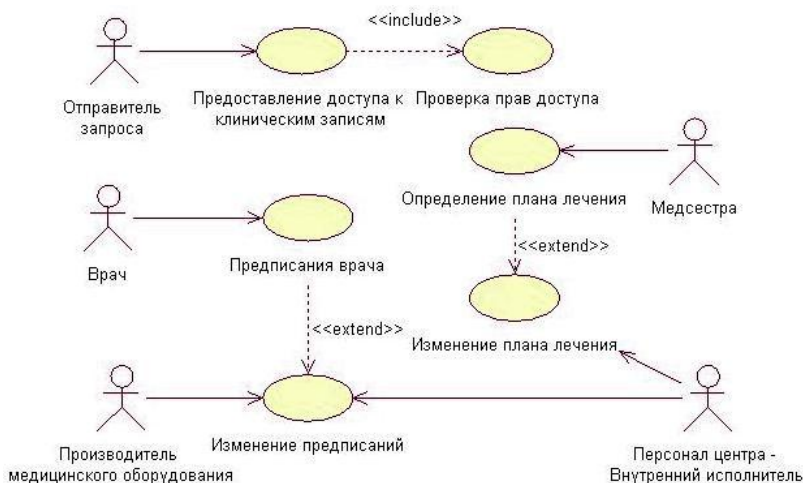


Рис. 8 Модель системных прецедентов

Внутренний исполнитель «Персонал центра» (рис. 6) и выполняемый им ручной процесс преобразован в системный прецедент «Предоставление доступа к клиническим записям». Внешние исполнители (например, «Производитель медицинского оборудования») непосредственно взаимодействуют с проектируемой системой, т.е. превращаются в исполнителей.

В модели отражены два специальных типа связи между прецедентами:

- «включает» (<<include>>) — один прецедент в процессе своего исполнения обязательно выполняет некий блок действий, составляющих другой прецедент;

- «расширяет» (<<extend>>) — когда прецеденты подобны по своим действиям, но один несет несколько большую функциональную нагрузку.

Прецедент «Проверка прав доступа» впервые появился на диаграммах и реализуется средствами разрабатываемой ИС. Для него нужно разрабатывать диаграмму последовательностей, описывающую его исполнение (рис. 9). В результате в проектируемой ИС появляются два новых объекта – программный модуль «Менеджер защиты» и информационный блок «Набор прав».

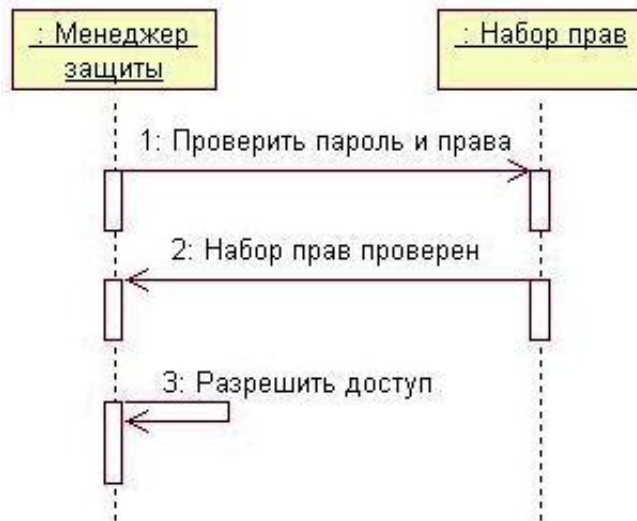


Рис. 9 Диаграмма последовательностей для прецедента «Проверка прав»

Результатом разработки модели системных прецедентов является исчерпывающий перечень функций, которые должны быть реализованы в проектируемой системе, а так же подробное описание необходимой реализации этих функций.

4.5 Анализ требований и предварительное проектирование системы

Основные задачи этапа:

- определить проект системы, который будет отвечать всем требованиям;
- разработать общий предварительный проект для всех команд разработчиков (проектировщиков баз данных, разработчиков приложений, системных архитекторов и пр.)

Основным инструментом на данном этапе являются диаграммы классов системы, которые строятся на основе разработанной *модели системных прецедентов*. Одновременно на этом этапе уточняются диаграммы последовательностей выполнения отдельных прецедентов, что приводит к изменениям в составе объектов и диаграммах классов. Это естественное отражение средствами *UML* итеративного процесса разработки системы.

Диаграммы классов системы заполняются объектами из *модели системных прецедентов*. Они содержат описание этих объектов в виде классов и описание взаимодействия между классами. Диаграмма классов, описывающая процедуры защиты доступа к данным, приведена на рис. 10.



Рис. 10 Диаграмма классов «Защита доступа»

В результате этого этапа проектирования появляется достаточно подробное описание состава и функций проектируемой системы, а также информации, которую необходимо использовать в

базе данных и в приложениях . Благодаря своему синтаксису, диаграммы классов оказываются хорошим средством структурирования и представления требований к функциональности, интерфейсам и данным для элементов проектируемой системы.

4.6 Разработка моделей базы данных и приложений

На данном этапе осуществляется отображение элементов полученных ранее моделей классов в элементы моделей базы данных и приложений:

- классы отображаются в таблицы;
- атрибуты — в столбцы;
- типы — в типы данных используемой СУБД;
- ассоциации — в связи между таблицами (ассоциации «многие-ко-многим» преобразуются в ассоциации «один-ко-многим» посредством создания дополнительных таблиц связей);
- приложения — в отдельные классы с окончательно определенными и связанными с данными в базе методами и атрибутами.

Поскольку модели базы данных и приложений строятся на основе единой логической модели, автоматически обеспечивается связность этих проектов (рис. 11).

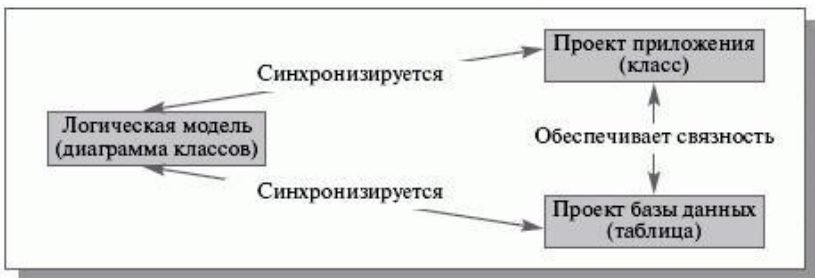


Рис. 11 Связь между проектами базы данных и приложений

В модель базы данных отображаются только перманентные классы, из которых удаляются атрибуты, не отображаемые в столбцах (например, атрибут типа «Общий объем продаж», который получается суммированием содержимого множества полей базы данных). Некоторые атрибуты (например, АДРЕС) могут отображаться в множество столбцов (СТРАНА, ГОРОД, УЛИЦА, ДОМ, ПОЧТОВЫЙ ИНДЕКС). Для каждого простого класса в мо-

дели базы данных формируется отдельная таблица, включающая столбцы, соответствующие атрибутам класса. Отображение классов подтипов в таблицы осуществляется одним из стандартных способов:

- одна таблица на класс;
- одна таблица на суперкласс;
- одна таблица на иерархию.

В первом случае для каждого из классов создается отдельная таблица, между которыми затем устанавливаются необходимые связи. Во втором случае создается таблица для суперкласса, а затем в каждую таблицу подклассов включаются столбцы для каждого из атрибутов суперкласса. В третьем — создается единая таблица, содержащая атрибуты как суперкласса, так и всех подклассов (рис. 12). При этом для выделения исходных таблиц подклассов в результирующую таблицу добавляется один или более дополнительных столбцов (на рисунке показан курсивом).

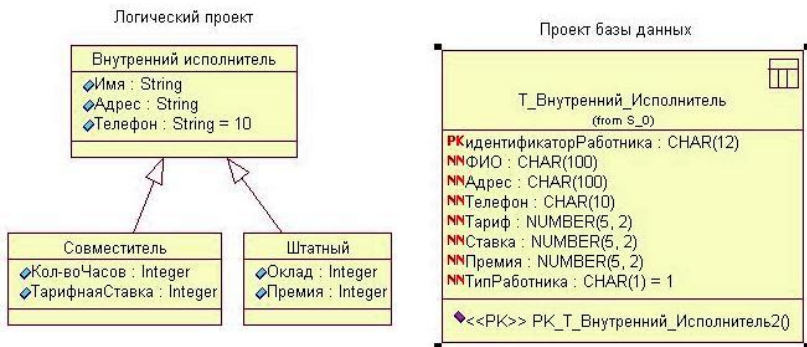


Рис. 12 Преобразование иерархии в таблицу

Разработка проекта базы данных осуществляется с использованием специального *UML*-профиля (*Profile for Database Design*), который включает следующие основные компоненты диаграмм:

- таблица — набор записей базы данных по определенному объекту;
- столбец — элемент таблицы, содержащий значения одного из атрибутов таблицы;
- первичный ключ (*PK*) — атрибут, однозначно идентифицирующий строку таблицы;

- внешний ключ (*FK*) — один или группа атрибутов одной таб-лицы, которые могут использоваться как первичный ключ другой таблицы;
- обязательная связь — связь между двумя таблицами, при ко-торой дочерняя таблица существует только вместе с родительской;
- необязательная связь — связь между таблицами, при которой каждая из таблиц может существовать независимо от другой;
- представление — виртуальная таблица, которая обладает всеми свойствами обычной таблицы, но не хранится постоянно в базе данных;
- хранимая процедура — функция обработки данных, выпол-няемая на сервере;
- домен – множество допустимых значений для столбца таблицы.

На рис. 13 представлен фрагмент модели базы данных — две таблицы, соответствующие классам «пациент» (рис. 3) и «мини-мальный набор данных» (рис. 7). Связь между ними обязательна, поскольку «минимальный набор данных» не может существовать без «пациента».

На диаграммах указываются дополнительные характеристики таблиц и столбцов:

- ограничения — определяют допустимые значения данных в столбце или операции над данными (ключ (*PK*, *FK*) — огра-ничение, определяющее тип ключа и его столбец; проверка (*Check*) — ограничение, определяющее правило контроля дан-ных; уникальность (*Unique*) — ограничение, определяющее, что в столбце содержатся неповторяющиеся данные);
- тип данных и пр.

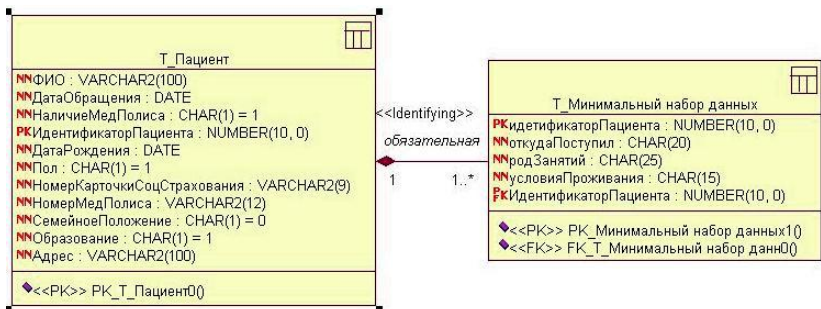


Рис. 13 Фрагмент модели базы данных

Результатом этапа является детальное описание проекта базы данных и приложений системы.

4.7 Проектирование физической реализации системы

Существуют два вида диаграмм физической реализации системы: диаграмма компонентов и диаграмма развертывания. На данном этапе проектирования модели баз данных и приложений дополняются обозначениями их размещения на технических средствах разрабатываемой системы. На рис. 14 приведено изображение разделения таблицы «пациент» на три части (<<Tablespace>>) в соответствии с первой буквой фамилии пациента.

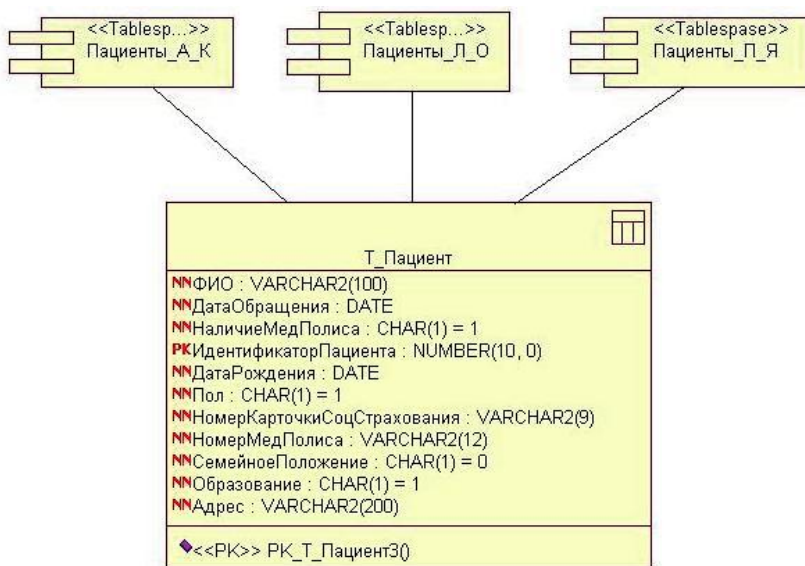


Рис. 14 Экстенды таблицы «Пациент»

Основными понятиями языка *UML*, которые используются на данном этапе, являются следующие:

- компонент — самостоятельный физический модуль системы;
- зависимость — связь между двумя элементами, при которой изменения в одном элементе вызывают изменения другого элемента;
- устройство — узел, не обрабатывающий данные;
- процессор — узел, выполняющий обработку данных;
- соединение — связь между устройствами и процессорами.

Диаграммы развертывания позволяют отобразить на единой схеме различные компоненты системы (программные и информационные) и их распределение по комплексу технических средств (рис. 15).

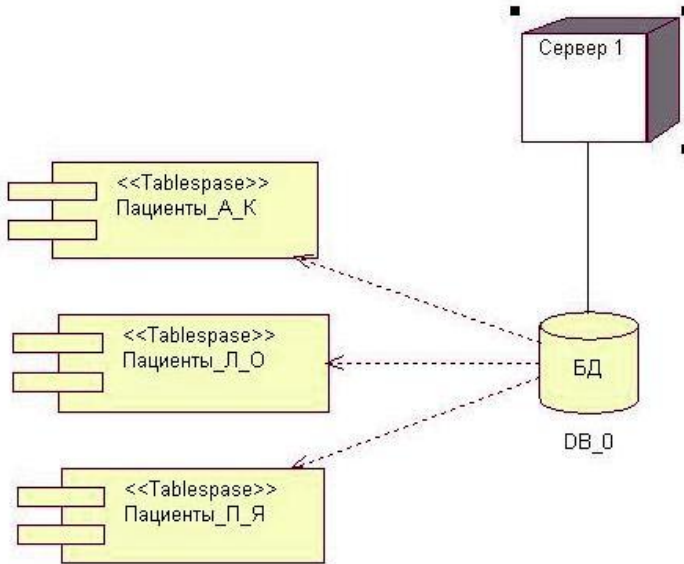


Рис. 15 Фрагмент диаграммы развертывания ИС

При проектировании сложной ИС она разделяется на части, и каждая из них затем исследуется и создается отдельно. В настоящее время используются два различных способа такого разбиения ИС на подсистемы: структурное (или функциональное) разбиение и объектная (компонентная) декомпозиция. Суть функционально-го разбиения может быть выражена известной формулой: «Про-грамма = Дан-ные + Ал-горитмы». При функциональной деком-позиции программной системы ее структура описывается блок-схемами, узлы которых представляют собой «обрабатывающие центры» (функции), а связи между узлами описывают движение данных.

При объектном разбиении в системе выделяются «активные сущности» — объекты (или компоненты), которые взаимодействуют друг с другом, обмениваясь сообщениями и выполняя соответ-

ствующие функции (методы) объекта. Если при проектировании ИС разбирается на объекты, то для ее визуального моделирования следует использовать язык *UML*. Если в основу проектирования положена функциональная декомпозиция ИС, то *UML* не нужен, и следует использовать структурные нотации.

При выборе подхода к разработке ИС следует учитывать, что визуальные модели все более широко используются в существующих технологиях управления проектированием систем, сложность, масштабы и функциональность которых постоянно возрастают. Они хорошо приспособлены для решения таких задач, как:

- физическое перераспределение вычислений и данных,
- обеспечение параллелизма вычислений, репликация БД,
- обеспечение безопасности доступа к ИС,
- оптимизация балансировки нагрузки ИС,
- устойчивость к сбоям и т. п.

Визуализированные средствами *UML* модели ИС позволяют наладить плодотворное взаимодействие между заказчиками, поль-зователями и командой разработчиков. Они обеспечивают ясность представления выбранных архитектурных решений и позволяют понять разрабатываемую систему во всей ее полноте.

5 Предлагаемые темы курсовых работ

1. Автоматизация деятельности информационной библиотечной системы.
2. Разработка автоматизированной системы регистрации паци-ентов поликлиники.
3. Проектирование автоматизированной системы обработки клиентского заказа торговой фирмы.
4. Автоматизация деятельности структурных подразделений деканата ВУЗа.
5. Проектирование автоматизированной системы регистрации оплаты за дошкольное образовательное учреждение.
6. Реализация автоматизированной системы кредитования коммерческого банка.
7. Проектирование информационной системы туристического агентства.
8. Реализация автоматизированной системы регистрации кли-ентов спортивного комплекса.
9. Разработка информационной системы реализации функций сельскохозяйственного предприятия.

6 Список рекомендуемой литературы

Основная:

1. Вендров А. М. Проектирование ПО экономических ИС. / А. М. Вендров — М.: Финансы и статистика, 2006—544 с.
2. Вендров А. М. Практикум по проектированию ПО экономических ИС. / А. М. Вендров — М.: Финансы и статистика, 2006—192 с.
3. Федорова Г. С. Проектирование информационных систем: Учебное пособие / Г. С. Федорова/ Кафедра проектирования экономических информационных систем. — М.: МЭСИ, 2001.—142 с.

Дополнительная:

1. Смирнова Г. Н. Проектирование экономических информационных систем: Учебник. / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов; Под ред. Ю. Ф. Тельнова — М.: Финансы и статистика, 2005, 512 с.

Интернет-источники:

1. <http://www.intuit.ru> Национальный открытый университет. В. Гре-кул. Этапы проектирования ИС с применением UML.

Библиографический список

1. Вендров, А. М. Проектирование ПО экономических ИС. / А. М. Вендров — М.: Финансы и статистика, 2006—544 с.
2. Вендров, А. М. Практикум по проектированию ПО экономических ИС. / А. М. Вендров — М.: Финансы и статистика, 2006. —192 с.
3. Федорова, Г. С. Проектирование информационных систем: Учебное пособие / Г. С. Федорова/ Кафедра проектирования экономических информационных систем. — М.: МЭСИ, 2001.—142 с.
4. Смирнова, Г. Н. Проектирование экономических информационных систем: Учебник. / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов; Под ред. Ю. Ф. Тельнова — М.: Финансы и статистика, 2005. — 512 с.
5. Маклаков, С. В. Моделирование бизнес-процессов с AllFusion Process Modeler (BPWin 4/1) / С. В. Маклаков — М.: «Диалог-МИФИ», 2003. — 236 с.
6. Леоненков, А. В. Самоучитель UML.-2-е изд. перераб и доп. / А. В. Леоненков — СПб.: БХВ-Петербург, 2004. — 432 с.
7. Фаулер, М. UML. Основы. Пер. с англ. / М. Фаулер, К. Скотт СПб.: «Символ Плюс», 2002. — 192 с.
8. Черемных, С. В. и др. Структурный анализ систем: IDEF-технологии. / С. В. Черемных — М.: Финансы и статистика. — 2001.— 108 с.
9. Калянов, Г. Н. Case-технологии. Консалтинг в автоматизации бизнес-процессов. / Г. Н. Калянов. — М.: Горячая линия — Телеком, 2002. — 320 с.

Интернет-источники:

1. <http://www.intuit.ru> Национальный открытый университет. В. Гре-кул. Этапы проектирования ИС с применением UML.

Содержание

1	Задачи курсового проектирования и требования к курсовой работе	3
2	Организация курсового проектирования	4
3	Структура и оформление курсовой работы	5
4	Методика проектирования ИС с применением языка моделирования UML	19
7	4.1 Разработка модели бизнес-прецедентов	9
4.2	Выявление обобщающих классов	13
4.3	Разработка концептуальной модели данных	14
4.4	Разработка требований к системе	16
4.5	Анализ требований и предварительное проектирование системы	19
4.6	Разработка моделей базы данных и приложений	20
4.7	Проектирование физической реализации системы	23
5	Предлагаемые темы курсовых работ	26
6	Список рекомендуемой литературы	27

Электронное учебное издание

Черноверская Виктория Владимировна
Кульман Татьяна Николаевна

**Подготовка и оформление курсовых работ по
дисциплине
«Проектирование информационных систем»**

Электронное методическое пособие

Филиал «Протвино»
государственного университета «Дубна»
142281 г. Протвино Московской обл.,
Северный проезд, д. 9