**Методические рекомендации по подготовке и защите курсовых работ по дисциплине «Проектный практикум»**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc507605026)

[1. ОРГАНИЗАЦИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ 6](#_Toc507605027)

[1.1. Определение темы курсового проектирования 6](#_Toc507605028)

[1.2. Руководство курсовым проектированием 8](#_Toc507605029)

[1.3. Защита проекта и критерии оценки 9](#_Toc507605030)

[2. ОФОРМЛЕНИЕ КУРСОВОГО ПРОЕКТА 12](#_Toc507605031)

[2.1. Структура пояснительной записки 12](#_Toc507605032)

[2.2. Порядок выполнения проекта 14](#_Toc507605033)

[2.3. Правила оформления пояснительной записки 15](#_Toc507605034)

[3. РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ПРОЕКТИРОВАНИЯ 17](#_Toc507605035)

[3.1. Анализ требований и предметной области. Прототипирование 18](#_Toc507605036)

[3.2. Выбор и обоснование средств разработки 29](#_Toc507605037)

[3.3. Моделирование динамической и статической структур проекта 31](#_Toc507605038)

[3.4. Проектирование физической структуры проекта 49](#_Toc507605039)

[3.5. Расчёт экономических параметров проекта 51](#_Toc507605040)

[3.6. Разработка плана тестирования 56](#_Toc507605041)

[ЗАКЛЮЧЕНИЕ 58](#_Toc507605042)

[ЛИТЕРАТУРА 59](#_Toc507605043)

[ПРИЛОЖЕНИЯ 60](#_Toc507605044)

[Приложение 1. Примерные темы курсовых работ 60](#_Toc507605045)

[Приложение 2. Титульный лист курсовой работы 62](#_Toc507605046)

[Приложение 3. Стандарты оформления курсовой работы 63](#_Toc507605047)

[Приложение 4. Отзыв на курсовую работу 64](#_Toc507605048)

[Приложение 5. Шаблон для написания сценария для отдельного варианта использования 66](#_Toc507605049)

# ВВЕДЕНИЕ

Методические указания разработаны на основе программ учебных дисциплин «Проектирование информационных систем», «Программная инженерия», «Проектный практикум» и имеют целью оказать помощь студенту при выполнении задания курсового проекта выполняемого в рамках дисциплины «Проектный практикум» и подготовке пояснительной записки к курсовому проекту.

Курсовой проект по дисциплине «Проектный практикум» является индивидуальной, самостоятельно выполненной работой студента. Его выполнение осуществляется на втором этапе изучения учебной дисциплины, после прохождения теоретической части курса. В процессе работы студент должен приобрести практические навыки анализа предметной области и моделирования системы с использованием унифицированного языка моделирования UML при помощи сред разработки в которые он встроен (Rational Rose, IBM Rational Software Architect, Microsoft Visual Studio, StarUML и других).

Курсовой проект выполняется с целью формирования у студента навыков самостоятельного решения профессиональных задач, требующих творческого и инновационного подхода.

3адачами выполнения курсового проекта являются:

* закрепление и расширение знаний по всем разделам дисциплины;
* систематизации знаний по смежным дисциплинам;
* выработки у студента навыков научно-исследовательской работы;
* обучения студентов методам самостоятельной аналитической и проектной работы в области информационных технологий;
* систематизации, обобщения и анализа фактического материала по проблемам проектирования информационных систем.

При выполнении курсового проекта студент должен продемонстрировать способности к таким видам деятельности, как:

* поиск требуемой информации по теме;
* изучение и критический анализ полученных материалов;
* систематизация и обобщение имеющейся информации;
* самостоятельное определение путей решения поставленных задач;
* оформление решения задач в виде итогового документа;
* логическое обоснование и формулировка выводов, предложений и рекомендаций по результатам работы.

Выполнение курсового проекта предполагает консультационную помощь со стороны преподавателя и творческое развитие студентом темы и разделов курсового проекта.

Курсовой проект выполняется и защищается в сроки, определенные и утверждённые учебным графиком.

# ОРГАНИЗАЦИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Настоящие методические рекомендации разработаны Департаментом анализа данных, принятия решений и финансовых технологий в целях организации проведения курсовых работ студентов бакалавриата, изучающих дисциплину «Проектный практикум».

Методические рекомендации разъясняют порядок подготовки и защиту курсовых работ студентами в сроки, определяемые соответствии с учебными рабочими планами (РУП) Финуниверситета. В учебном году, как правило, в РУП планируется не более одной курсовой работы студента.

Выполнение курсовой работы проводится с целью формирования общепрофессиональных компетенций и способности к научно-исследовательской и проектно-конструкторской работе.

## Определение темы курсового проектирования

На первой неделе срока, выделенного учебным планом для выполнения курсового проекта, до сведения студентов доводится список тем курсового проектирования. Тема может сопровождаться пояснениями, в зависимости от степени сложности. Примерный список тем можно увидеть в Приложении 1 настоящих методических указаний.

Студенту предоставляется право в течение недели выбрать любую тему из предложенного списка. Также студент может сам предложить интересующую его тему, если она соответствует изучаемому предмету и целям курсового проектирования.

При выборе темы курсового проекта рекомендуется проконсультироваться с преподавателями дисциплин "Проектирование информационных систем", «Программная инженерия» и «Проектный практикум».

Выполнять курсовые проекты на одну и ту же тему нескольким студентам из одной учебной группы не разрешается.

Выбранная студентом тема утверждается руководителем курсового проектирования, о чем вносится соответствующая запись в бланк задания на подготовку курсового проекта.

Задание на выполнение курсового проекта является нормативным документом, устанавливающим границы и глубину разработки темы, а также сроки представления работы на кафедру в завершенном виде.

Выполнение курсового проекта предполагает проведение анализа выбранной в соответствии с темой проекта предметной области и моделирования системы средствами унифицированного языка моделирования UML в выбранной студентом среде проектирования.

Студент самостоятельно подбирает необходимую литературу, практики, стандарты и подходы для успешной реализации проекта. При написании курсовой работы необходимо знание Законов, ГОСТов, технических и бизнес требований, относящихся к теме исследования.

## Руководство курсовым проектированием

Назначение руководителей курсовой работой осуществляется из числа преподавателей департамента (профессоров, доцентов, старших преподавателей) решением Руководителя департамента.

Основными функциями руководителя курсовой работы являются:

* + Консультирование по вопросам содержания и последовательности выполнения курсовой работы;
  + Рекомендации студенту в подборе необходимой литературы и других материалов;
  + Контроль хода выполнения курсовой работы;
  + Подготовка отзыва на курсовую работу, итоговая проверка работы на «Антиплагиат», организация защиты курсовой работы.

После утверждения темы курсового проекта студент обязан изучить исходные данные к курсовому проекту, подобрать и изучить литературу по теме проекта, составить план, регулярно посещать консультации руководителя, дорабатывать отдельные части проекта по замечаниям руководителя, своевременно подготовить и сдать на проверку законченный курсовой проект и защитить его.

На руководителя возлагается ответственность за постоянное наблюдение за разработкой всех разделов курсового проекта в соответствующие сроки и оказание студенту необходимой помощи на всех этапах выполнения проекта.

В процессе проектирования для студентов проводятся в соответствии с утвержденным департаментом графиком групповые и индивидуальные консультации.

Руководитель обязан:

* установить студенту календарный график выполнения этапов проектирования и занести его в бланк задания на подготовку курсового проекта;
* регулярно проводить консультации в соответствии с графиком при уточнении темы, разработке плана и проекта, составлении списка литературы, анализе материала и т.д.;
* контролировать соблюдение календарных сроков и качество выполнения как отдельных частей, так и проекта в целом. Если при проверке обнаружатся ошибки, неполнота объема, незавершенность проектирования или низкое качество как проектирования, так и оформления курсовой работы, то проект возвращается студенту для доработки;
* подготовить рецензию к курсовому проекту;
* принять (по возможности, совместно с комиссией) защиту курсового проекта.

## Защита проекта и критерии оценки

Завершающим этапом выполнения студентом курсовой работы является ее защита. Студент обязан явиться на защиту курсовой работы в назначенное руководителем время. Защита организуется руководителем в установленный срок (то есть **до 25 мая).** Расписание защит курсовых работ размещается на интернет-странице департамента и доводиться ведущим преподавателем до студентов.

Результат защиты курсовой работы (проекта) студента оценивается по балльно-рейтинговой и пятибалльной системам. В ходе защиты оцениваются компетенции студента, которые он должен приобрести при подготовке курсовой работы и продемонстрировать в ходе ее защиты, а также уровень знаний, умений, владений (навыков), которые студент должен продемонстрировать для подтверждения освоенных компетенций.

На защите руководитель оценивает, в том числе:

* понимание студентом темы;
* научный кругозор автора и глубину проработки вопросов;
* завершенность проекта и полноту раскрытия темы;
* обоснованность суждений автора, логику изложения материала;
* наличие фактического материала, его актуальность;
* знание последних тенденций в исследуемой области;
* знание соответствующей нормативно - правовой базы;
* адекватность полученных результатов проекта;
* правильность оформления работы.

Руководитель проверяет курсовую работу и составляет письменный Отзыв в формате Приложения 4 к настоящим Методическим рекомендациям.

Оценка качества выполнения курсовой работы и ее защиты осуществляется по 100 бальной системе в соответствии с утвержденными департаментом критериями и переводом итоговой суммы баллов в 5 бальную оценку.

|  |  |
| --- | --- |
| 100 бальная система | 5 бальная система |
| 86-100 | отлично |
| 70-85 | хорошо |
| 50-69 | удовлетворительно |
| менее 50 | неудовлетворительно |

Итоговая оценка по 100 бальной шкале складывается из 85 баллов, полученных за разработку курсовой работы, и 15 баллов, полученных на защите (см. Приложение 4)

На титульном листе курсовой работы руководитель: указывает оценку и дату защиты работы, ставит подпись. Руководитель передает курсовую работу вместе с отзывом сотрудникам департамента, курирующим проведение курсовых работ.

В случае несогласия студента с оценкой курсовой работы руководителем, он, в течение трех календарных дней, следующих за датой защиты, подает апелляцию на имя руководителя департамента - для рассмотрения в апелляционной комиссии, создаваемой департаментом. Апелляционная комиссия проводит заседание, студент уведомляется о предстоящем заседании комиссии и имеет право присутствовать на нем.

Студент, не выполнивший в срок курсовую работу или получивший неудовлетворительную оценку па защите, не допускается к сдаче экзамена по соответствующей дисциплине.

Студент, не защитивший курсовую работу в установленный срок, должен подготовить и защитить курсовую работу в период ликвидации академической задолженности.

# ОФОРМЛЕНИЕ КУРСОВОГО ПРОЕКТА

## Структура пояснительной записки

Структура курсовой работы должна включать следующие разделы:

* титульный лист;
* оглавление;
* введение;
* основная часть (3 главы и более);
* заключение;
* список использованных источников;
* приложения.

Формат титульного листа – в Приложении 2.

Во введении обосновывается актуальность темы, формулируются цели и задачи работы. Обычно введение составляется после написания основной части работы и заключения. Его объем 1–3 страницы. Приводится краткий обзор использованных практик и методов реализации проекта.

Основная часть обычно состоит минимум из трёх глав примерное содержание которых приведено ниже:

* в первой главе содержатся обследование автоматизируемого объекта, включая характеристику объекта и/или (предмета) исследования, описание бизнес-процессов организации, с выделением из них автоматизируемых бизнес-процессов, расчёта временных и ресурсных параметров проекта;
* во второй главе содержатся подбор инструментальных средств, планируемые к использованию при решении поставленной задачи, детальная проработка проекта, выделение акторов, написание сценариев и оценка их реалистичности, прототипирование;
* в третьей главе содержаться проектные решения динамической и статических частей проекта, описания полученных классов, их взаимодействие между собой, данные и сообщения передаваемые между классами и объектами, деятельность и действия отдельных объектов, физическое размещение отдельных модулей на физических устройствах.
* в четвёртой главе содержаться экономические расчёты проекта, план и программы тестирования разрабатываемого программного обеспечения.

Каждая глава может включать два и более параграфов. Название главы не должно дублировать название темы, а название параграфов - названия глав.

В основной части автор должен раскрыть сущность тех вопросов, которые он обозначил в плане. При раскрытии темы наряду с проработкой конкретного направления студент должен показать знания теоретических и практических основ проектной деятельности, оценить передовые подходы и практики и возможности их использования в проекте.

Следует дать обзор существующих практик и подходов по реализации аналогичных проектов и произвести выбор наиболее оптимальных для данного проекта. Обязательным требованием работы является наличие табличного, графического и цифрового материала, их анализ и обобщение. Приводимый в работе проектный материал (схемы) должен быть тесно увязан с текстом. По возможности весь материал необходимо свести в таблицы, графики, диаграммы. Все приведенные схемы должны быть пронумерованы.

Основная часть должна содержать изложение детальных шагов процесса проектирования, детальное описание решения выявленных проблем и обоснование принятых технических и программных решений.

Основную часть работы завершает Заключение, в котором содержатся выводы и рекомендации относительно практического применения материалов работы. В заключении автор должен в сжатом виде привести основные выводы, сформулированные в результате проектной деятельности, и предложения, если такие имеются. Заключение не должно служить логическим продолжением работы и содержать новые моменты, не рассмотренные в основной части работы. Оно может повторять выписки из основной части работы. Объем заключения 1–3 страницы.

Объем курсовой работы составляет не менее 25-30 страниц.

## Порядок выполнения проекта

Проект выполняется студентами индивидуально. Допускается выполнение одного большого или реального проекта по автоматизации несколькими студентами при условии, что проект разбит на отдельные, чётко ограниченные, подпроекты (подсистемы) и студенты разрабатывают каждый свою.

Проект требует разработки и документирования программного средства, используемого для решения определённой задачи. При этом рекомендуется применять готовые ERP или MRP системы как платформу на которой и будет использоваться разрабатываемая подсистема.

При этом следует учитывать следующие рекомендации:

1. Используйте системный подход. Помните, что методология UML построена на его основе.

2. Разделите задачи проекта так, чтобы получить равномерную загрузку над проектом на всё время его выполнения.

4. Разработайте прототип проектируемого программного средства, иначе обеспечить полноту анализа исходной задачи сложно.

5. Сохраняйте логику проекта от постановки задачи до формирования тестовых заданий: все рисунки и модели должны быть связаны; функции, попавшие в алгоритм, должны прослеживаться на мнемосхеме, в функциональной структуре и в структуре модулей программы; документы с экранных форм должны фигурировать, как минимум, в перечне документов проекта, в мнемосхеме и в алгоритме.

6. Используйте Интернет для пополнения сведений об исследуемом процессе и отраслевой специфике. Пояснительные записки, более чем на 75% совпадающие с уже оцененными преподавателем в текущем или прошлые годы, считаются плагиатом и оцениваются на «неудовлетворительно».

7. Следите за сроками проекта. Потребуется несколько консультаций для получения положительной оценки.

## Правила оформления пояснительной записки

Курсовая работа выполняется с использованием компьютерной техники, средствами текстового редактора (Microsoft Word, Libra Writer и др.), на одной стороне листа белой бумаги формата А4.

Стандарты используемые при оформлении курсовой работы перечислены в Приложении 3.

Рекомендуется размечать структуру работы средствами Microsoft Word: «уровень 1» (разделы), «уровень 2» (параграфы), «основной текст», и пользоваться функцией «автособираемое оглавление». В структуре работы не используются элементы третьего и последующего уровней.

Шрифт - Times New Roman, размер 14, цвет шрифта должен быть черным. Межстрочные интервалы и интервалы до (после) абзаца 1,5. Каждый абзац начинается с отступа 1,25, выравнивание текста «по ширине». Размеры полей страницы: правое - 10 мм, верхнее и нижнее - 20 мм, левое – 30 мм.

Названия разделов, являющиеся заголовками, печатаются прописными буквами, а названия параграфов (подзаголовки) - строчными буквами (кроме первой прописной). Заголовки и подзаголовки при печатании текста письменной работы на принтере выделяются полужирным шрифтом. Заголовки, подзаголовки и подстрочные сноски (состоящие из нескольких строк) печатаются через одинарный интервал.

Страницы курсовой работы должны нумероваться арабскими цифрами, нумерация должна быть сквозная, по всему тексту работы. Номер страницы проставляют в центре нижней части листа. ТИТУЛЬНЫЙ ЛИСТ включается в общую нумерацию страниц работы, однако номер страницы на нем не ставится.

Каждый раздел (главу) работы следует начинать с нового листа. Параграф начинать с нового листа не нужно.

Если в работе имеются схемы, таблицы, графики, диаграммы, фотоснимки, то их следует располагать непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. Иллюстрации следует нумеровать арабскими цифрами сквозной нумерацией (то есть по всему тексту) - 1,2,3, и т.д.

При наличии в работе таблицы ее наименование (краткое и точное) должно располагаться над таблицей без абзацного отступа в одну строку. Таблицу, как и рисунок, располагать непосредственно после текста, в котором она упоминаются впервые, или на следующей странице. Таблицы в тексте следует нумеровать сквозной нумерацией арабскими цифрами, но всему тексту или в рамках главы (2,1 и т.д.). Если таблица вынесена в приложение, то она нумеруется отдельно арабскими цифрами с добавлением перед номером слова «Приложение».

Если таблица имеет заголовок, то он пишется с прописной буквы, и точка в конце не ставится. Разрывать таблицу и переносить часть ее на другую страницу можно только в том случае, если целиком не умещается на одной странице, при этом на другую страницу переносится и шапка таблицы, а также заголовок «Продолжение таблицы».

В курсовых работах используются ссылки в форме подстрочных сносок. Подстрочные сноски оформляются внизу страницы, на которой расположен текст (цитата). Для этого в конце текста (цитаты) ставится цифра или звездочка, обозначающая порядковый номер сноски па данной странице. Нумерация подстрочных сносок может быть сквозной по всему тексту работы.

При цитировании необходимо соблюдать следующие правила: текст цитаты заключается в кавычки, и приводится в той грамматической, форме, в какой он дан в источнике, с сохранением особенностей авторского написания. Цитирование должно быть полным, без произвольного сокращения цитируемого фрагмента и без искажения смысла. Пропуск слов, предложений, абзацев при цитировании допускается, если не влечет искажение всего фрагмента, и обозначается многоточием, которое ставится на место пропуска. Если цитата включается в текст, то первое слово пишется со строчной буквы. Если цитата выделяется из основного текста, то ее пишут от левого поля страницы на расстоянии абзацного отступа, при этом каждая цитата должна сопровождаться ссылкой на источник.

При написании текста можно использовать только общепринятые сокращения и условные обозначения.

При наличии в работе значительного количества сокращений или специальных терминов, требующих расшифровки (пояснения) и повторяющихся многократно в тексте работы – их следует вынести в глоссарий, оформляемый в виде отдельного приложения.

Рекомендуется следующий порядок расположения материалов в СПИСКЕ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ:

* Законы, нормативные акты, ГОСТ;
* Монографии, учебники и периодические издания;
* Интернет-ресурсы.

После защиты курсовая работа представляется в департамент в печатном виде, сброшюрованная в папке «под дырокол», без прозрачных файлов, в одном экземпляре, а также в электронном виде.

# РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ПРОЕКТИРОВАНИЯ

Тематика курсовой работы по дисциплине "Проектный практикум» ориентирована на проработку проекта разработки программного обеспечения для решения конкретных прикладных задач в заданной предметной области.

Основное задание курсовой работы заключается в построении моделей программного обеспечения при помощи CASE-средств (рекомендуемое – StarUML). Процесс создания модели состоит из нескольких этапов, примерное описание которых дано ниже.

При выполнении работы следует опираться на практикум по изучению UML, выполняемый в рамках курса.

## Анализ требований и предметной области. Прототипирование

**Тема курсовой работы:** Создание информационной системы ВУЗа для тестирования студентов преподавателями.

**Детализация требований:**

Учебному заведению потребовалось создать систему, содержащую комплекс тестов для контроля знаний слушателей и ведения статистики успеваемости.

Пользователями системы являются преподаватели и слушатели.

Преподаватели имеют возможность создавать тесты и редактировать уже имеющиеся тесты и материалы. Каждый преподаватель имеет доступ только к зарегистрированным им тестам. Преподаватель имеет возможность просмотра общей статистики по прохождению слушателями созданного им теста.

Каждый тест имеет название, краткое описание. Одно тестовое задание включает в себя формулировку и четыре варианта ответа, один из которых - верный. Каждому заданию назначается коэффициент (вес, уровень сложности), который учитывается при формировании оценки знаний слушателя.

Слушатели могут проходить контрольное тестирование, предварительно зарегистрировавшись. Результат тестирования фиксируется в системе. Слушатели не имеют права проходить контрольное тестирование более одного раза, изменять тесты, править личную статистику. Также слушатели не должны иметь доступ к общей статистике. Если слушатель уже проходил выбранный тест, то система должна известить его об этом.

Система должна обеспечивать многопользовательский режим работы. Должна быть в работоспособном состоянии 12 часов в день 7 дней в неделю. **Анализ требований и описание предметной области:**

Этап анализа состоит в исследовании предметной области и требований к разрабатываемой системе. Следует учесть, что анализ требований к разрабатываемой системе является важнейшим среди всех этапов ее жизненного цикла, он оказывает существенное влияние на все последующие этапы. На этом этапе необходимо понять, что должна делать будущая система, что предполагается сделать и задокументировать это. Целью анализа является преобразование общих, неясных знаний о требованиях к будущей системе в четкие определения.

При описании процессов автоматизируемой предметной области, определении требований к будущей программной системе используются диаграммы вариантов использования (диаграммы прецедентов). Такие диаграммы отражают объекты как системы, так и предметной области, и задачи, ими выполняемые. Часто этот вид диаграмм называют диаграммой функций, потому что на основе набора таких диаграмм создается список требований к системе и определяется множество выполняемых системой функций.

Диаграмма вариантов использования, как правило, отражает требования к системе с точки зрения пользователя, она описывает типичное взаимодействие между пользователем и системой. Таким образом, варианты использования (прецеденты) – это функции, выполняемые системой, а действующие лица (актеры) – это заинтересованные лица по отношению к создаваемой системе. На данном этапе следует выполнить несколько задач:

1) определить рамки системы;

2) идентифицировать основных действующих лиц (актеров), потребности (цели) которых удовлетворяются с помощью системы или на которых система оказывает влияние;

3) для каждого исполнителя определить его задачи;

4) определить прецеденты (варианты использования), удовлетворяющие

потребности каждого исполнителя, и присвоить им имена в соответствии с задачами; На этом этапе осуществляется составление глоссария проекта (см. таблицу 1) и написание укороченного технического задания (далее – ТЗ) на разработку системы.

|  |  |  |
| --- | --- | --- |
| № п/п | Актор | Определение |
| 1 | Администратор (Administrator) | Специалист, контролирующий работу системы |
| 2 | Пользователь (AnyUser) | Слушатель, преподаватель или администратор, которому необходимо авторизоваться в системе |
| 3 | Преподаватель (Professor) | Преподаватель, автор и администратор теста |
| 4 | Результат | Результат прохождения тестирования слушателем |
| 5 | Слушатель (Student) | Человек проходящий тестирование |
| 6 | Список результатов  (ResultTest) | Общий список результатов тестирования |
| 7 | Тест (Test) | Набор заданий, разрабатываемых преподавателем, доступный слушателям для выполнения |

Студент формулирует техническое задание, в котором основное внимание уделяется тому, что именно должно быть сделано в курсовом проекте. То есть последовательно излагается полное описание продукта, который будет выполнен, его пользовательского интерфейса, принципов функционирования.

Техническое задание предоставляется преподавателю для проверки. На данном этапе важно определить, что студент и преподаватель одинаково понимают задачу и будет выполняться требуемая работа. Согласовывается объём работ и необходимые детали. Техническое задание является основным документом при разрешении споров относительно направления разработки и объема выполненной работы и практически полностью входит в первую главу работы. Техническое задание считается принятым и согласованным после его подписания преподавателем.

**Диаграмма вариантов использования для рассматриваемого примера представлена на Рисунке 1.**

Каждое действующее лицо (actor) и вариант использования диаграммы вариантов использования должны сопровождаться описанием. Описание действующего лица должно коротко сообщать о роли данного лица. Описание варианта использования должно включать в себя пояснение, предусловие, потоки событий (основной и альтернативные, если таковые есть) и постусловие. Описания представляют собой либо присоединенные текстовые файлы, либо текст, введенный в поле Documentation спецификации соответствующего элемента диаграммы.

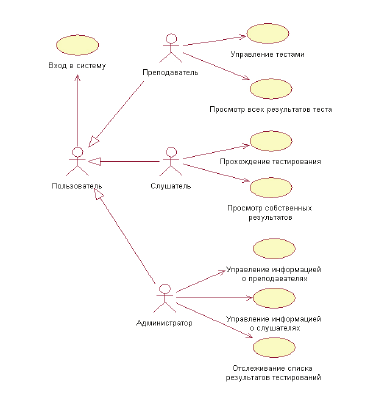


Рисунок 3.1. Диаграмма вариантов использования для системы тестирования.

**Описание действующих лиц**

Пользователь (AnyUser) - входит в систему (идентификация).

Преподаватель (Professor) - автор обучающих материалов (тестов). Имеет право создавать, изменять, удалять тесты. Может просматривать результаты прохождения теста.

Слушатель (Student) - проходит тестирование и просматривает результат

тестирования. Администратор (Administrator) - ведет информацию по преподавателям и

слушателям; отслеживает список результатов тестирования, удаляет устаревшие.

**Описание вариантов использования**

С учётом того что изобразительных средств языка UML явно не хватает для того, чтобы учесть на диаграммах вариантов использования особенности функционального поведения сложной системы, рекомендуется дополнять этот тип диаграмм текстовыми сценариями в комментариях или описаниях, которые уточняют или детализируют последовательность действий, совершаемых системой при выполнении ее вариантов использования.

Сценарий (scenario) - определенная последовательность действий, которая описывает действия актеров и поведение моделируемой системы в форме обычного текста.

В контексте языка UML сценарий используется для дополнительной иллюстрации взаимодействия актеров и вариантов использования. Существуют различные способы представления подобных сценариев. В рамках курсового проектирования предлагается применять шаблон, общий вид которого приведен в Приложении 5.

В рамках указанного шаблона каждый сценарий описывается тремя разделами: главный раздел, типичный ход событий и исключения.

Главный раздел – таблица, содержащая основные характеристики варианта использования.

Типичный ход событий обеспечивает наглядное представление общения с системой. Как правило, типичный ход событий описывают с использованием таблицы, где в первом столбце приводятся действия внешних исполнителей (актеров), а во втором столбце - отклик системы на действия актеров.

Исключения представляются в виде таблицы, где описываются альтернативные потоки действий в рамках варианта использования. Перед главным разделом указываются предусловия и постусловия.

Предусловия – это перечень предпосылок, которые всегда должны выполняться до начала сценария прецедента. Выполнение этих условий не проверяется в рамках логики выполнения данной операции, а предполагается, что они истинны.

Постусловия описывают, какие условия должны выполняться в случае успешного завершения сценария. Эти результаты должны удовлетворять интересам всех заинтересованных лиц. Выделение постусловий - наиболее важная часть описания системных операций.

При написании сценариев вариантов использования важно понимать, что текст сценария должен дополнять или уточнять диаграмму вариантов использования, но не заменять ее полностью. В противном случае будут потеряны достоинства визуального представления моделей.

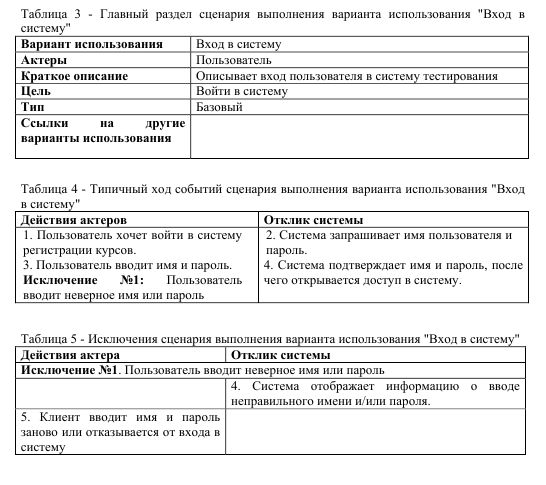
Ниже приведены сценарии, представляющие варианты использования рассматриваемого примера (см. таблицы 2 - 18).

**Вариант использования "Вход в систему"**

Предусловия: отсутствуют.

Постусловия: если вариант использования выполнен успешно, пользователь входит в систему. В противном случае состояние системы не изменяется.

Таблица 2 - Главный раздел сценария выполнения варианта использования "Вход в систему"



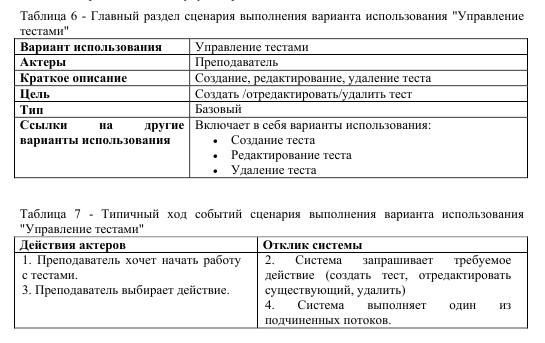
3.2.3.5 Вариант использования "Управление тестами"

Предусловия: пользователь должен быть зарегистрирован и обладать

необходимыми правами.

Постусловия: если вариант использования выполнен успешно, изменения,

внесенные преподавателем, будут сохранены системой.

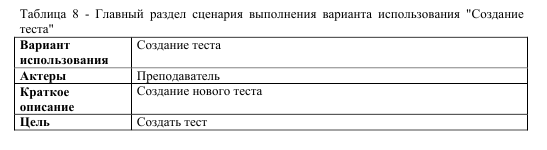


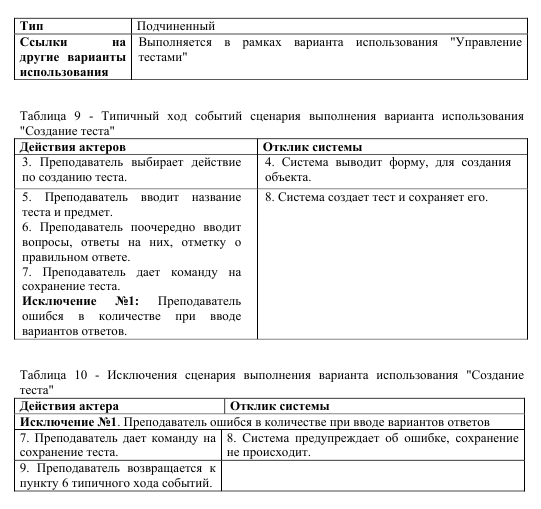
3.2.3.5.1 Вариант использования "Создание теста"

Предусловия: пользователь выбрал действие.

Постусловия: если вариант использования выполнен успешно, изменения,

внесенные преподавателем, будут сохранены системой. В противном случае состояние

системы не изменяется. 

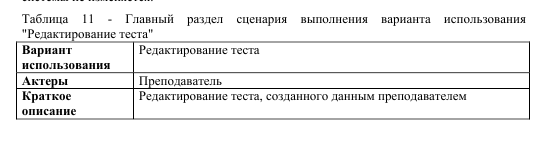


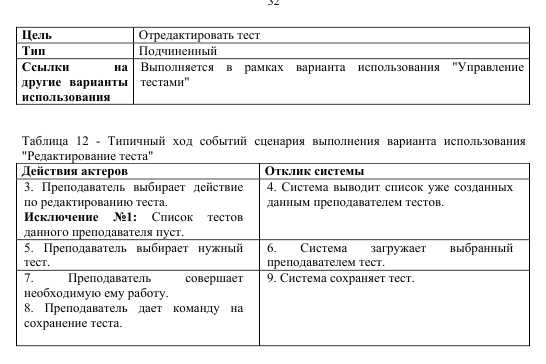
3.2.3.5.2 Вариант использования "Редактирование теста"

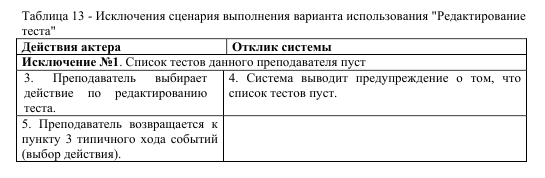
Предусловия: пользователь выбрал действие.

Постусловия: если вариант использования выполнен успешно, изменения,

внесенные преподавателем, будут сохранены системой. В противном случае состояние

системы не изменяется. 



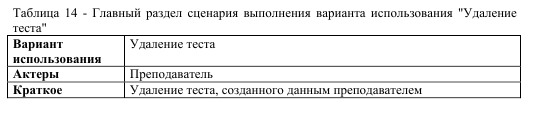


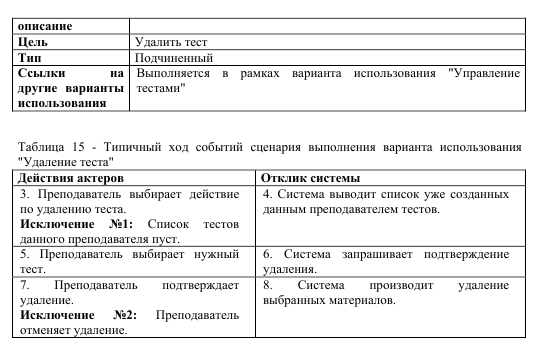
3.2.3.5.3 Вариант использования "Удаление теста"

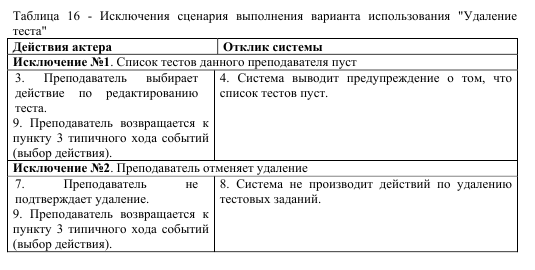
Предусловия: пользователь выбрал действие.

Постусловия: если вариант использования выполнен успешно, изменения,

внесенные преподавателем, будут сохранены системой. В противном случае состояние

системы не изменяется. 





3.2.3.6 Вариант использования "Прохождение тестирования"

Предусловия: пользователь должен быть зарегистрирован и обладать

необходимыми правами.

Постусловия: если вариант использования выполнен успешно, слушатель получит

оценку за прохождение теста, информация об этом будет сохранена системой.



Сценарии для остальных вариантов использования расписываются аналогичным образом.

## Выбор и обоснование средств разработки

В конце первой или начале второй главы студенту необходимо обосновать выбор инструментов проектирования и разработки, языка программирования и СУБД, используемой для хранения промежуточных результатов.

Содержание главы:

Выбор метода разработки

В данном разделе студент должен указать обоснование и причины использования того или иного инструментария в данном проекте (среды разработки Microsoft Visual Studio, Rational Software Architect, 1C, StarUML, Eclipse и другие).

Выбор и обоснование языка программирования

На этом этапе экспертными или расчётными методами выбирается язык или среда программирования.

Пример:

Для выбора языка программирования методом морфологического анализа произведён выбор из следующих альтернатив:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Язык программирования** | **Вес**  **критерия** | **Язык 1** | **Язык 2** | **Язык 3** |
| Дороговизна лицензии | 3 | 3 | 2 | 1 |
| Сложность освоения | 2 | 2 | … | … |
| Оптимальность кода | 1 | 2 |  |  |
| … |  |  |  |  |
| Суммарный  приоритет | - | 15 |  |  |

Суммарный приоритет рассчитывается методом построчного суммирования произведений значений ячеек и соответствующих весов.

1. Описание языка программирования или среды разработки, включающее наименование языка (среды), основные особенности и причины выбора.

Пример:

Для автоматизации процесса снабжения использован язык программирования Visual Basic for Application, входящий в состав СУБД Microsoft Access, представляющий собой алгоритмический язык программирования …

Пример:

Для выбора СУБД произведён выбор из следующих альтернатив:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вид СУБД** | **Вес**  **критерия** | **СУБД 1** | **СУБД 2** | **СУБД 3** |
| Сложность освоения | 1 | … | … | … |
| Аппаратные требования | 3 |  |  |  |
| Скорость работы | 3 |  |  |  |
| … |  |  |  |  |
| Суммарный  приоритет | … |  |  |  |

1. Описание СУБД, содержащее наименование СУБД и причины её выбора.

На этом этапе экспертными или расчётными методами выбирается СУБД, либо доказывается отсутствие необходимости в её использовании.

Пример:

Для автоматизации процесса снабжения выбрана СУБД Microsoft Access 2003, которая позволяет создать персональную базу данных и формы для работы с ней. Выбор СУБД обусловлен …

## Моделирование динамической и статической структур проекта

**Моделирование поведения**

Следующим важным этапом является описание функций, которые должна выполнять разрабатываемая информационная система. Диаграмма вариантов использования (прецедентов) дает ответ на вопрос, что должна делать система. Для того, чтобы выяснить, как система функционирует, следует построить диаграммы взаимодействия.

При необходимости (особенно это касается разрабатываемых аппаратно-программных систем) можно включить диаграммы деятельности (или состояний) и сопроводить их пояснениями, указывающими, какому потоку событий они соответствуют (если это не ясно из их названия), и комментариями.

Диаграммы взаимодействия между объектами (последовательности или кооперативные), соответствуют потокам событий вариантов использования. Одна диаграмма, как правило, применяется для отображения объектов, участвующих в одном потоке событий варианта использования. Каждая диаграмма должна сопровождаться необходимыми пояснениями.

Диаграммы последовательности отражают упорядоченный по времени поток событий, происходящих в рамках варианта использования. На диаграмме последовательностей иллюстрируются события, инициированные в системе актерами. На диаграмме последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами. Для данных диаграмм ключевым моментом является именно динамика взаимодействия объектов во времени.

Моделирование временной упорядоченности потока управления осуществляется следующим образом:

1. Выбирается вариант использования (прецедент), в рамках которого взаимодействие объектов системы представляет наибольший интерес.
2. Определяется, какие объекты принимают в нем участие. Их следует разместить на диаграмме последовательностей слева направо так, чтобы более важные объекты были расположены левее.
3. Чаще всего объекты существуют на протяжении всего взаимодействия. Для тех же объектов, которые создаются или уничтожаются в ходе взаимодействия, на линиях жизни явно отмечаются моменты рождения и смерти с помощью подходящих стереотипных сообщений.
4. Сначала размещается сообщение, инициирующее взаимодействие. Затем располагаются все последующие сообщения сверху вниз между линиями жизни объектов. Чтобы объяснить семантику взаимодействия, следует показать свойства каждого сообщения (например, его параметры).

В первую очередь строятся диаграммы, описывающие основной поток событий и его подчиненные потоки. Для каждого альтернативного потока событий строится отдельная диаграмма. Нецелесообразно описывать тривиальные потоки событий (например, если в потоке участвует только один объект).

На рисунках в пояснительной записке представляется окончательный вариант диаграмм последовательности.

Под рисунками приводятся комментарии, содержащие краткое описание классов и сообщений, представленных на диаграммах.

**Моделирование структуры**

В процессе разработки диаграмм взаимодействия следует идентифицировать классы, их атрибуты, обязанности и отношения.

Класс – описание множества объектов с одинаковыми атрибутами, связями и семантикой. Изображается в виде прямоугольника (до назначения стереотипа).

Обозначение класса состоит из трех частей, в которых указываются имя класса, его атрибуты и методы.

Атрибут – именованное свойство класса, описывающее диапазон значений, которые может принимать экземпляр атрибута. Можно уточнить спецификацию атрибута, указав его класс и начальное значение по умолчанию.

Атрибуты классов анализа определяются, исходя из знаний о предметной области, требований к системе и глоссария.

Метод – реализация услуги, которая может быть запрошена у любого объекта данного класса, чтобы вызвать определенное его поведение. Можно специфицировать операцию, указав тип и значение по умолчанию всех параметров, а для функций тип возвращаемого значения.

Обязанность – действие, которое объект обязан выполнять по запросу других объектов. Обязанность преобразуется в одну или более операций класса на шаге проектирования. Обязанности определяются исходя из сообщений на диаграммах взаимодействия и документируются в классах как операции "анализа", которые появляются там автоматически в процессе построения диаграмм взаимодействия (соотнесения сообщений с операциями).

Отношения между классами (ассоциации) определяются в два этапа:

1. Начальный набор связей определяется на основе анализа диаграмм взаимодействия. Если два объекта обмениваются сообщениями, между ними на диаграмме должна существовать связь, которая преобразуется в двунаправленную ассоциацию между соответствующими классами. Если сообщения между некоторой парой объектов передаются только в одном направлении, то для соответствующей ассоциации вводится направление навигации.
2. Анализируются и уточняются ассоциации между классами-сущностями. Задаются мощности ассоциаций, могут использоваться множественные ассоциации, агрегации, обобщения и ассоциации-классы.

В потоках событий каждого варианта использования выявляются классы трех типов (стереотипы классов):

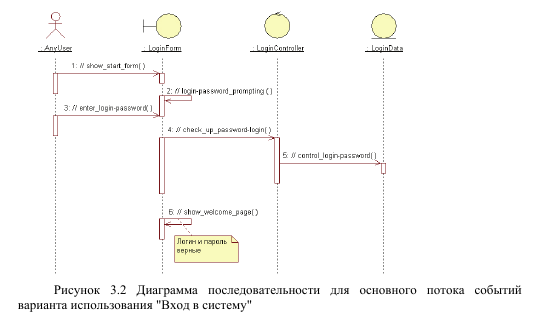
1. Граничные классы (Boundary) – служат посредниками при взаимодействии внешних объектов с системой. Как правило, для каждой пары "действующее лицо - вариант использования" определяется один граничный класс. Типы граничных классов: пользовательский интерфейс (обмен информацией с пользователем, без деталей интерфейса - кнопок, списков, окон), системный интерфейс и аппаратный интерфейс (используемые протоколы, без деталей их реализации).
2. Классы-сущности (Entity) – представляют собой ключевые абстракции (понятия) разрабатываемой системы. Источники выявления классов-сущностей: ключевые абстракции, созданные в процессе архитектурного анализа, глоссарий, описание потоков событий вариантов использования.
3. Управляющие классы (Control) – обеспечивают координацию поведения объектов в системе. Могут отсутствовать в некоторых вариантах использования, ограничивающихся простыми манипуляциями с хранимыми данными. Как правило, для каждого варианта использования определяется один управляющий класс. Примеры управляющих классов: менеджер транзакций, координатор ресурсов, обработчик ошибок.

**Создание диаграмм в рамках примера**

На рисунке 3.2 представлена диаграмма последовательности для основного потока событий варианта использования "Вход в систему". Под рисунком приводится краткое описание данной диаграммы.

Это только одна из диаграмм, необходимых для моделирования варианта использования "Вход в систему". Она соответствует успешному варианту хода событий.

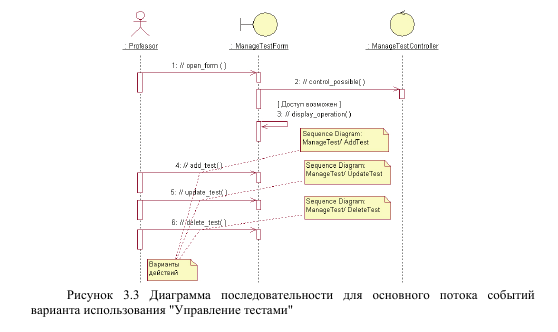
Для описания того, что случится, если Пользователь введет неверное имя или пароль, разрабатывается другая диаграмма. Каждый альтернативный поток варианта использования может быть промоделирован с помощью своих собственных диаграмм последовательности.



Краткое описание диаграммы:

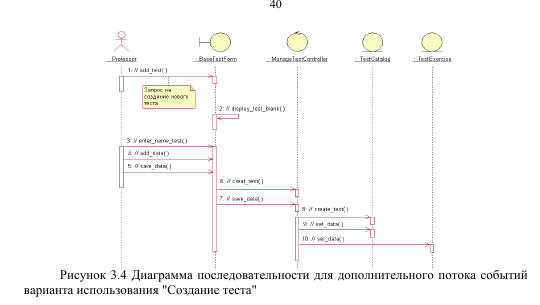
Пользователь является инициатором данного варианта использования. Обращение к меню происходит посредством мыши или клавиатуры, что вызывает появление формы для ввода логина и пароля пользователя. После получения информации от пользователя, система проверяет ее. Информация для проверки предварительно заносится в систему Администратором. При положительном исходе проверки система открывает доступ пользователю в соответствии с его статусом и выводит на экран форму приглашения.

На рисунках 3.3-3.6 представлены диаграммы последовательности для основного потока событий варианта использования "Управление тестами".



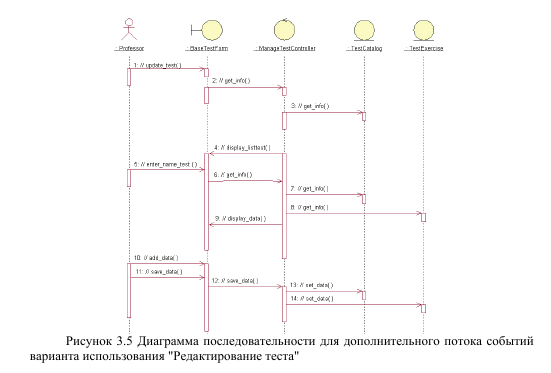
Краткое описание диаграммы:

Преподаватель является инициатором варианта использования. После успешной авторизации в системе открывается форма, содержащая перечень возможных последующих операций. Если функция control\_posible() возвращает значение, соотвествующее доступности выбора, то Преподаватель имеет возможность выбрать одно из предложенных действий. После получения выбора Преподавателя, система открывает соответствующую форму.



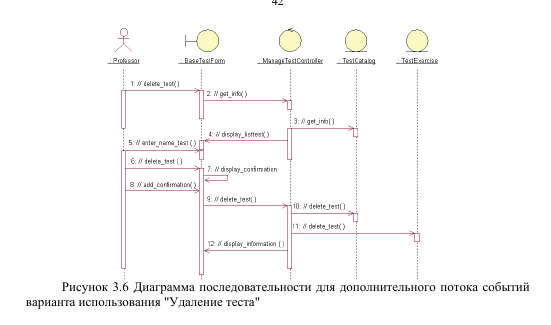
Краткое описание диаграммы:

Инициатором дополнительного потока событий является Преподаватель, осуществляющий выбор действия "Создание теста". Далее происходит открытие формы для ввода общей информации о тесте и тестовых заданий с вариантами ответов. После получения команды на сохранение данных происходит их проверка и сохранение в базе тестов.



Краткое описание диаграммы:

Инициатором дополнительного потока событий является Преподаватель, осуществляющий выбор действия "Редактирование теста". Далее происходит открытие формы для отображения списка тестов, созданных ранее данным Преподавателем. Преподаватель выбирает тест для редактирования, после чего загружается форма, в которой содержится общая информация о тесте и задания с вариантами ответов. Пользователь вносит изменения и дает команду на сохранение данных. Результаты редактирования проверяются и сохраняются в базе тестов.

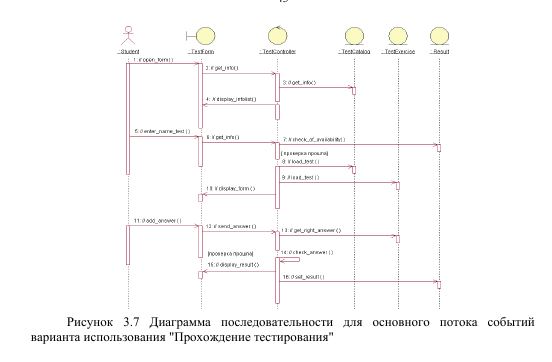


Краткое описание диаграммы:

Инициатором дополнительного потока событий является Преподаватель, осуществляющий выбор действия "Удаление теста". Далее происходит открытие формы для отображения списка тестов, созданных ранее данным Преподавателем.

Преподаватель выбирает тест для удаления, после чего система запрашивает подтверждение на удаление. После подтверждения тест удаляется из базы тестов. По завершении действия выводится сообщение об успешном выполнении действия.

На рисунке 3.7 представлена диаграмма последовательности для варианта использования "Прохождение тестирования".



Краткое описание диаграммы:

Инициатором данного варианта использования является Слушатель. Далее происходит открытие формы для отображения списка тестов. Слушатель выбирает тест. Система проверяет, не проходил ли ранее данный Слушатель выбранный тест. Если не проходил, то отображается форма с тестовыми заданиями. Слушатель последовательно отвечает на вопросы теста путем выбора одного из ответов. После выполнения всех заданий система подсчитывает результат, отображает на экране и сохраняет его.

При необходимости, можно построить диаграммы кооперации. Кооперативные диаграммы отражают практически ту же самую информацию, что и диаграммы последовательности, только с другой точки зрения. Если диаграммы последовательности упорядочены по времени, то кооперативные диаграммы заостряют внимание на связях между объектами, рассматривают структурные особенности взаимодействия объектов.

Затем строятся диаграммы классов.

Диаграммы классов являются центральным звеном объектно-ориентированных методов. Они иллюстрируют взаимоотношения программных элементов, а не понятий из предметной области. Диаграммы классов предназначены для статического моделирования объектов.

Для составления диаграммы классов необходимо выполнить следующие действия:

1. Выделить программные классы.
2. Отобразить их на диаграмме классов.
3. Добавить необходимые атрибуты, ассоциации и методы.

В качестве примера на рисунке 3.8 представлена диаграмма классов для варианта использования "Создание теста".



На заключительной стадии этапа следует проверить согласованность и корректность всех диаграмм. В случае наличия ошибок вернуться к начальным шагам и повторить необходимые действия.

**Проектирование системы**

В процессе проектирования основное внимание уделяется концептуальному решению (в виде программного обеспечения или аппаратных средств), обеспечивающему выполнение основных требований. Например, на этапе проектирования описываются программные объекты или схема базы данных.

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

При проектировании системы требуется:

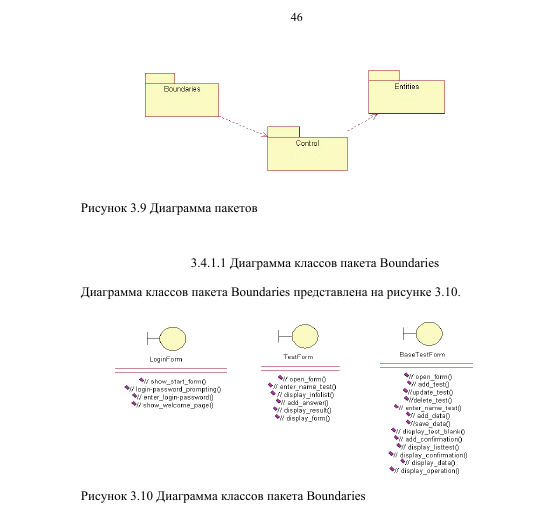
* разместить классы по пакетам (использовать деление: пользовательский интерфейс – управление – данные; или другое в зависимости от постановки задачи);
* связать объекты с классами, сообщения на диаграммах взаимодействия – с операциями;
* каждый класс снабдить описанием, которое должно включать в себя краткое описание (ответственность класса), описание атрибутов и операций;
* построить диаграммы классов системы, отображающие связи между классами;
* для описания поведения экземпляров отдельных классов построить диаграммы состояний;
* разработать (если это требуется вариантом задания) схему базы данных и отобразить ее на диаграмме "сущность – связь".

**Размещение классов системы по пакетам**

Предлагается объединить классы в пакеты по стереотипу. Для этого создаются пакеты:

* Boundaries (Границы) – интерфейсные объекты системы.
* Control (Управление) – управляющие объекты системы.
* Entities (Сущности) – информационные объекты системы.

В пакеты помещаются соответствующие классы. Строится диаграмма пакетов. Затем для каждого пакета следует создать диаграммы классов.

Для рассматриваемой задачи диаграмма пакетов представлена на рисунке 3.6. 

**Описание класса LoginForm**

LoginForm – представляет форму для организации диалога с пользователем в процессе авторизации.

Атрибуты: нет.

Операции:

// show\_start\_form () - открытие стартовой формы,

// login-password\_prompting () - отображение формы для ввода имени пользователя и пароля,

// enter\_login-password () - ввод имени пользователя и пароля,

// show\_welcome\_page () - открытие формы-приглашения с информацией об

успешной авторизации.

**Описание класса TestForm**

TestForm – объект, позволяющий просматривать и выполнять тест.

Атрибуты: нет.

Операции:

// open form () - открытие начальной экранной формы,

// display\_infolist () - отображение списка тестов,

// enter\_name\_test () - выбор нужного теста из списка,

// display\_form () - отображение формы с заданием теста,

// add\_answer () - ввод ответа,

// display\_result () - отображение результата тестирования.

**Описание класса BaseTestForm**

BaseTestForm – формы для организации диалога с пользователем в процессе управление тестами. Позволяет преподавателю выбирать основные операции над тестами.

Атрибуты: нет.

Операции:

// open\_form () - открытие начальной экранной формы,

// display\_operation () - отображение меню с доступными вариантами действий,

// add\_test () - активизация подчиненного потока варианта использования «Добавление теста»,

// update\_test () - активизация подчиненного потока варианта использования «Редактирование теста»,

// delete\_test () - Позволяет запустить подчиненный поток варианта использования «Удаление теста»,

// enter\_name\_test () - ввод названия теста,

// add data () - добавление данных в тест,

// save\_data () - сохранение введенных данных,

// display\_data () - отображение данных теста,

// display\_confirmation () - отображение запроса на подтверждение удаления теста,

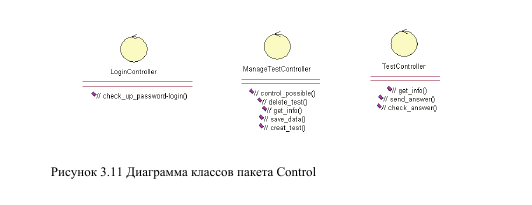
// add\_confirmation () - ввод подтверждения удаления данных,

// display\_information () - вывод информации об успешном завершении операции.

// display\_listtest () - отображение списка тестов.

**Диаграмма классов пакета Control**

Диаграмма классов пакета Control представлена на рисунке 3.11.



**Описание класса LoginController**

LoginControler – обеспечивает управление доступом к ресурсам системы, проверку правильности ввода данных для авторизации.

Атрибуты: нет.

Операции:

// check\_up\_password-login () - проверка правильности введенных данных, открытие доступа в систему при отсутствии ошибок.

**Описание класса ManageTestController**

ManageTestController – обеспечивает управление данными при создании, редактировании и удалении теста.

Атрибуты: нет.

Операции:

// control\_possible () - запрос на доступ к операциям, позволяет удостовериться в возможности работы менеджера,

// creat\_test () - позволяет создать новый тест,

// save\_data () - позволяет сохранить данные,

// get\_info () - запрос на предоставление информации (списка тестов),

// delete\_test () - позволяет удалить тест.

**Описание класса TestController**

TestController– обеспечивает управление данными при прохождении теста.

Атрибуты: нет.

Операции:

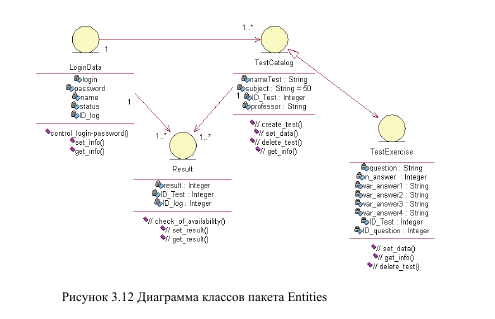
// get\_info () - запрос на предоставление информации (списка тестов),

// send\_answer () - позволяет принять данные тестирования,

// check\_answer () - позволяет обработать данные тестирования.

**Диаграмма классов пакета Entities**

Диаграмма классов пакета Entities представлена на рисунке 3.12.



**Описание класса LoginData**

LoginData – обеспечивает хранение информации о пользователях.

Атрибуты:

ID\_log - уникальный идентификационный номер пользователя,

login - имя учётной записи пользователя,

password - пароль для входа в систему,

name - полное имя пользователя,

status - статус пользователя (администратор, преподаватель, слушатель).

Операции:

// control\_login-password () - получение данных для проверки правильности введенных данных.

**Описание класса Result**

Result – обеспечивает хранение данных о прохождении тестов (результаты тестирования).

Атрибуты:

ID\_log - уникальный идентификационный номер пользователя,

ID\_Test - уникальный номер теста,

result - результат тестирования,

Операции:

// check\_of\_availability () - получение данных для проверки правильности введенных данных.

// set\_result () - сохранение результата,

// get\_result () - получение результата.

**Описание класса TestCatalog**

TestCatalog – обеспечивает хранение основной информации о готовых тестах.

Атрибуты:

ID\_Test - уникальный номер теста,

nameTest - название теста,

subject- тематика теста,

professor - полное имя преподавателя, создавшего тест.

Операции:

// creat\_test () - создание нового теста,

// set\_data () - сохранение данных,

// get\_info () - передача информации (списка тестов),

// delete\_test () - удаление теста.

**Описание класса TestExercise**

TestExercise – обеспечивает хранение заданий тестов.

Атрибуты:

ID\_Test - уникальный номер теста, ID\_question - номер вопроса теста,

question - фформулировка вопроса теста,

var\_answer1 - вариант ответа на вопрос,

var\_answer2 - вариант ответа на вопрос,

var\_answer3 - вариант ответа на вопрос,

var\_answer4 - вариант ответа на вопрос,

n\_answer - номер правильного ответа.

Операции:

// set\_data () - сохранение данных,

// get\_info () - передача информации,

// delete\_test () - удаление теста.

## Проектирование физической структуры проекта

**Моделирование распределенной конфигурации системы**

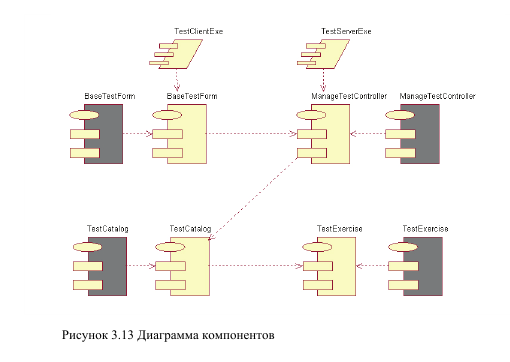
При реализации системы необходимо построить диаграммы развертывания: диаграммы компонентов и диаграмму размещения. Должна быть произведена проверка корректности модели и автоматическая генерация кода средствами CASE-среды.

Диаграмма компонентов показывает организацию и взаимосвязи программных компонентов, представленных в виде исходного кода, двоичных или выполняемых файлах, динамически подключаемых библиотек и т.д.

В общем случае каждому классу должны соответствовать два компонента — компонент спецификации и компонент реализации. В будущем каждому компоненту будет соответствовать свой файл. Например, в языке C++ классу соответствуют два файла-компонента: h-файл (файл спецификации) и срр-файл (файл реализации), на диаграмме компонентов они отображаются как Package specification и Package body.

Пример диаграммы компонентов для организации управления тестами представлен

на рисунке 3.13.



Диаграммы размещения - диаграммы, используемые при моделировании физических аспектов объектно-ориентированной системы. Такая диаграмма показывает конфигурацию узлов, где производится обработка информации, и то, какие компоненты размещены на каждом узле.

Распределенная конфигурация системы моделируется с помощью диаграммы размещения. Ее основные элементы:

* узел (node) - вычислительный ресурс (процессор или другое устройство (дисковая память, контроллеры различных устройств и т.д.). Для узла можно задать выполняющиеся на нем процессы;
* соединение (connection) - канал взаимодействия узлов (сеть).

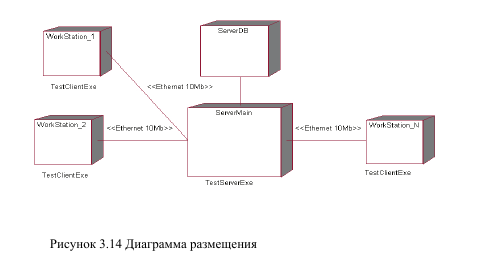
Узел, может относиться к одному из двух типов:

* узел устройства – физический вычислительный ресурс с памятью и процессорным элементом, на котором работает программное обеспечение;
* исполняющий узел окружения – программный вычислительный ресурс, работающий в рамках другого узла и обеспечивающий выполнение других выполняемых программных элементов.

Распределение процессов по узлам сети производится с учетом следующих факторов:

* используемые образцы распределения т(рехзвенная клиент-серверная конфигурация, "толстый" клиент, "тонкий" клиент, равноправные узлы (peer-to-peer) и т.д.);
* время отклика;
* минимизация сетевого трафика;
* мощность узла;
* надежность оборудования и коммуникаций.

Пример диаграммы размещения приведен на рисунке 3.14. Это конфигурация системы с распределением процессов по узлам.



## Расчёт экономических параметров проекта

Расчет затрат на разработку программного продукта включает следующие статьи расходов:

1. Расходные материалы;
2. Затраты на эксплуатацию специального оборудования;
3. Затраты на оплату труда (основная + дополнительная заработная плата);
4. Затраты на отчисления в социальные фонды;
5. Накладные расходы;
6. Прочие расходы;
7. Издержки производства (себестоимость);
8. Прибыль;
9. Цена.

**Расчет затрат на оплату труда разработчиков ПП**

Рассчитаем затраты на оплату труда разработчиков программного продукта. Основная заработная плата производственного персонала рассчитывается исходя из произведения оклада исполнителя (согласно штатному расписанию сотрудников) на время выполнения работы.

Расчет основной заработной платы производится по формуле:

где:

- оклад работника согласно штатному расписанию сотрудников (таблица ), руб.;

- время, затраченное на разработку программного продукта, мес.

Результаты расчетов заработной платы исполнителей за все время разработки представлены в таблице.

Таблица - Расчёт основной и дополнительной заработной платы

|  |
| --- |
|  |
| Наименование  Должности | Кол-во  человек | Месячный оклад | Повышающий коэффициент  квалификац.  уровня | Время разработки, мес. | Общая  зарплата, р. |  |
| Инженер-системотехник | 1 | 15000 | 1,5 | 11 | 247500 |  |
| Старший программист | 1 | 12000 | 1,3 | 10 | 156000 |  |
| Основная зп |  |  |  |  | 403500 |  |
| Дополнительная зп | 15% |  |  |  | 60525 |  |
| Итого |  |  |  |  | 464025 |  |

Произведем расчет отчислений в социальные фонды, результаты представлены в таблице.

Таблица - Определение отчислений на социальные нужды

|  |
| --- |
|  |
| Название отчислений | Процент от заработной платы исполнителей, % | Сумма, руб. |  |
| Фонд социального страхования | 2,9 | 13456,7 |  |
| Пенсионный фонд | 22 | 102085,5 |  |
| Федеральный ФОМС | 5,1 | 23665,3 |  |
| Итого | 30 | 139207,5 |  |
|  |  |  |  |

**Экономическая эффективность программного изделия**

Эффективность любого программного продукта определяется такими факторами, как эффективность процесса разработки и сопровождения и качество самого ПП.

В свою очередь качество ПП включает в себя следующие составляющие:

с точки зрения специалиста-пользователя данного программного изделия;

с позиции использования ресурсов и их оценки;

по выполнению требований на программное изделие.

**Годовой экономический эффект от использования ПП**

Годовой экономический эффект от использования ПП, как элемента новой или усовершенствованной технологии проектирования и внедрения вычислительного процесса или процесса создания нового ПП определяется по формуле:

где:

* приведенные затраты на единицу работ (функций), выполняемых с помощью базового и нового ПП, руб.;
* годовой объем выполняемых с помощью нового программного продукта работ (функций) в расчетном году, натур. ед.

Приведенные затраты на единицу работ (функций), выполняемых по базовой и новой технологиям ведения вычислительного процесса, рассчитывается по формуле:

,

где:

* себестоимость единицы работ, руб. (складывается из затрат на электроэнергию при работе принтера и ноутбука, затрат на бумагу при распечатке и затрат на оплату работника при выполнении единицы работ);
* удельные капитальные вложения, связанные с использованием программного продукта, руб. (руб.)
* нормативный коэффициент эффективности капитальных вложений (0,15).

Определим себестоимость единицы работ, которая складывается из затрат на печать 1 листа бумаги, стоимости 1 операции и затрат на ноутбук и принтер. Затраты на печать сократим за счет уменьшения масштаба. Так, если в базовом ПП печать 1 листа стоила 5 рублей, то в новом ПП - 3 рубля. Для расчета затрат на оплату работника при выполнении единицы работ необходимо найти его часовой заработок. Зарплата программиста, который будет работать с программой, составляет 12000 рублей в месяц при 8-ми часовом рабочем дне. Отсюда рассчитаем оплату его труда за час: 12000/ (22•8) =68,2р.

Скорость обработки информации в новом ПП будет гораздо выше, чем в базовом. Примем, что время типовой операции в новом продукте занимает всего лишь 2 минуты, а в базовом ПП - 4 минуты. Получается, что в час на новой программе оператор успевает сделать 30 операций, а на старой 15 операций. Тогда стоимость одной операции, выполняемой в новом продукте, равна 68,2/30=2,27 руб., а в старом - 68,2/15=4,55 руб.

Таким образом, имеем следующую себестоимость единицы работ

Следовательно:

Годовой экономический эффект от использования ПП равен:

**Расчет экономии затрат на оплату машинного времени**

Расчет экономии затрат на оплату машинного времени при замене базового ПП на новый производится по формуле:

где:

* годовая экономия затрат на оплату времени работы i-го ресурса вычислительного комплекса или коммуникационных средств, связанных с выполнением функций, исследуемых ПП, руб.;
* стоимость одного часа работы i-ого ресурса вычислительного комплекса или коммуникационных средств применительно к конкретному контуру и режиму оборудования, руб. /ч.;
* время выполнения j-х операций на i-ом ресурсе вычислительного комплекса или коммуникационных средств в базовом и новом вариантах исследования ПП;
* количество j-х операций, выполняемых по новому варианту ПП в течение года.

=4,21\* (0, 19+0,38) +3 = 5,4руб. /ч.

Время выполнения операций базового и нового программных продуктов 0,067 и 0,033ч соответственно.

**Расчет условной годовой экономии затрат на оплату труда работников**

Расчет условной годовой экономии затрат на оплату труда работников при изменении трудоемкости процессов подготовки и обработки информации в результате внедрения ПП определяются по формуле:

где:

* годовой фонд заработной платы работника на подготовке и обработке данных, руб.;
* трудоемкость подготовки и обработки единицы данных в базовом и новом вариантах ПП, чел/ч. Под подготовкой и обработкой единицы данных - понимается введение исходных данных. В новом ПП на это уходит 15 минут, следовательно, трудоемкость =4 чел/ч. В базовом ПП из-за более низкой скорости работы и менее удобного интерфейса введение данных занимает на 20 % больше времени, т.е.18 минут, отсюда =3,3 чел/ч.
* годовой объем работ по подготовке и обработке данных на новом варианте ПП, натур. ед.;
* годовой фонд времени одного работника (программист) по подготовке и обработке данных, ч.

.

**Расчет коэффициента экономической эффективности**

Коэффициент экономической эффективности показывает величину годового прироста прибыли, образующуюся в результате производства или эксплуатации нового ПП, на один рубль единовременных капитальных вложений (К), т.е.:

Годовой прирост прибыли () находят как разность между ценой (ЦПР) и себестоимостью единицы программного изделия.

Отсюда найдем:

Поскольку значение коэффициента экономической эффективности превышает нормативное значение (), т.е. выполняется условие, то можем утверждать, что производство и внедрение нового программного продукта является эффективным.

## Разработка плана тестирования

Процесс тестирования можно поделить на три этапа:

1. Проверка в обычных условиях. В данном случае тестируется основной функционал разработанного программного обеспечения. Полученный результат должен соответствовать ожидаемому.
2. Проверка в чрезвычайных условиях. В этих случаях подразумевается получение граничных данных, которые могут негативно повлиять на работоспособность созданного программного обеспечения. В качестве примера можно привести работу с чрезвычайно большими или малыми числами, или вообще, полное отсутствие получаемой информации.
3. Проверка при исключительных ситуациях. Она предполагает использование данных, которые лежат за гранью обработки. В таких ситуациях очень плохо, когда программное обеспечение воспринимает их как пригодные к расчету и выдаёт правдоподобный результат.

Необходимо позаботиться, чтобы в подобных случаях происходило отвержение любых данных, которые не могут быть корректно обработаны. Также необходимо предусмотреть информирование об этом пользователя.

# ЗАКЛЮЧЕНИЕ

Настоящие методические указания предназначены в помощь студенту при выполнении задания курсового проекта по дисциплине П"роектирование информационных систем", а также при написании пояснительной записки к нему, и предоставляют достаточную информацию для успешного завершения работы. Для получения более полных знаний по предмету студенту предлагается изучить литературу, перечисленную в списке литературы.

В процессе генерации Rational Rose отображает логическое описание класса в каркас программного кода — в коде появляются языковые описания имени класса, свойств класса и заголовки методов. Кроме того, для описания тела каждого метода формируется программная заготовка. Появляются и программные связи классов.

Предполагается дополнение этого кода в конкретной среде программирования, имеющей связь с системой Rational Rose. После каждого существенного дополнения с помощью возвратного проектирования, основанного на использовании связи, можно модифицировать диаграммы классов, вводя в них изменения, соответствующие результатам

программирования.

# ЛИТЕРАТУРА

Автоматизированные информационные технологии в экономике: Учебник/под ред. проф. Г.А. Титоренко. - М.: ЮНИТИ, 2015.

Вендров, CASE – технологии. Современные методы и средства проектирования информационных систем. / А.М. Вендров– М.: Финансы и статистика, 2014.

Вендров, Проектирование программного обеспечения экономических информационных систем: Учебник. / А.М. Вендров – М.: Финансы и статистика, 2010.

Гринберг, Информационные технологии управления: Учебное пособие для вузов / А.С. Гринберг, Н.Н. Горбачев, А.С. Бондаренко. – М.: ЮНИТИ-ДАНА, 2014.

Куликов, Автоматизированное проектирование информационно-управляющих систем. Системное моделирование предметной области. / Г.Г. Куликов, А.Н. Набатов, А.В. Речкалов – Уфа: УГАТУ, 2013. 176 с.

Маклаков, Bpwin и Erwin. CASE- средства разработки информационных систем. / С.В. Маклаков– М.: "ДИАЛОГ-МИФИ", 2009. – 256 с.

Сергеева, Информатика: учебник / И.И. Сергеева, А.А. Музалевская, Н. В. Тарасова.-М.: Форум: ИНФРА-М, 2016.-335 с.:

# ПРИЛОЖЕНИЯ

## Приложение 1. Примерные темы курсовых работ

Настоящий перечень тем курсовых работ носит примерный характер. Студент вправе предложить собственную тему курсовой работы, обосновав ее целесообразность и соответствие дисциплине «Проектный практикум».

Окончательно тема работы может быть сформулирована совместно студентом и руководителем.

Тематика курсовых работ актуализируется кафедрой ежегодно. В списке представлены примерные темы:

1. Система планирования расписания учебных занятий в вузе.
2. Подсистема учёта и контроля успеваемости студентов ИС ВУЗа
3. Подсистема кафедры ИС ВУЗа
4. Подсистема деканата ИС ВУЗа
5. Информационная система издательской компании
6. Информационная система ТВ-канала
7. Подсистема учёта и хранения научных работ студентов и аспирантов ВУЗа
8. Информационная система страховой фирмы
9. Интегрированная информационная система банка
10. Информационная система туристического агентства
11. Информационная система сельскохозяйственного предприятия (колхоза)
12. Информационная система автобусного парка
13. Информационная система завода по производству металлоконструкций (стелажи, шкафы и др.)
14. Информационная система библиотеки
15. Информационная система музея
16. Информационная система складского комплекса (3-5 складов)
17. Информационная система управления фитнес-центром
18. Подсистема учёта клиентов ИС фитнес-центра
19. Информационная система процессингового центра
20. Информационная система эквайрингового центра
21. Подсистема контроля прохода на территорию предприятия с разными уровнями допуска ИС предприятия
22. Система анализа происшедших или происходящих событий (аналитическая)
23. Информационная система поликлиники
24. Информационная система склада
25. Информационная система агентства недвижимости
26. Информационная система кадрового агентства
27. Информационная система заказа туристических путевок через Интернет
28. Информационная система бензозаправочной станции
29. Информационная система автосервиса
30. Информационная система бюро бытовых услуг
31. Информационная система детского дошкольного учреждения
32. Информационная система ресторана
33. Информационная система гостиницы
34. Информационная обучающая система
35. Информационная подсистема «Абитуриенты» для автоматизации работы приемной комиссии вуза.
36. Информационная подсистема «Зарплата» для автоматизации начислений заработной платы в бухгалтерии.
37. Информационная подсистема контроля исполнения поручений
38. Информационная подсистема «Снабжение магазинов» для оптовой базы.
39. Информационная подсистема учёта академической успеваемости в ВУЗе.
40. Информационная подсистема «Аренда помещений»
41. Проектирование облачного хранилища данных для ИТ-инфраструктуры университета
42. Создание мобильного приложения подсистемы продаж ИС компании

## Приложение 2. Титульный лист курсовой работы

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ

РОССИЙСКОЙ ФЕДЕРАЦИИ»

Департамент анализа данных,

принятия решений и финансовых технологий

курсовая РАБОТА ПО ДИСЦИПЛИНЕ «ПРОЕКТНЫЙ ПРАКТИКУМ»

на тему:

**РАЗРАБОТКА ИнформационнОЙ подсистемЫ «Снабжение магазинов» для оптовой базы**

Выполнил:

Студент группы ПИ3-2

Факультета прикладной математики   
и информационных технологий

Иванов Иван Иванович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

Руководитель:

к.э.н., доцент

Иванов Иван Иванович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Итоговая оценка)

Москва 2018

## Приложение 3. Стандарты оформления курсовой работы

Курсовая работа пишется на русском языке и оформляется в соответствии с ГОСТами «Системы стандартов по информации, библиотечному и издательскому делу» (СИБИД):

* ГОСТ 7.32-2001 (СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления);
* ГОСТ 7.1-2003 (СИБИД. Библиографическая запись. Библиографическое описание. Общие требования и правила составления);
* ГОСТ 7.80-2000 (СИБИД. Библиографическая запись. Заголовок. Общие требования и правила составления);
* ГОСТ 7.82-2001 (СИБИД. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления);
* ГОСТ Р 7.0.5-2008 (СИБИД. Библиографическая ссылка. Общие требования и правила составления).

## Приложение 4. Отзыв на курсовую работу

Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«ФинансоВЫЙ УНИВЕРСИТЕТ

при Правительстве Российской Федерации»

(Финансовый университет)

Департамент анализа данных,

принятия решений и финансовых технологий

**ОТЗЫВ НА КУРСОВУЮ РАБОТУ (ПРОЕКТ)**

Студента(ки)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ гр. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема: «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование критерия** | **Показатель критерия (на максимальный бал)** | **Макс. балл** | **Факт. балл** |
| **1** | **2** | **3** | **4** |
| 1. **Подготовительный этап выполнения курсовой работы (проекта)** | | **18** |  |
| Выбор темы, составление библиографии и плана в установленные сроки | Курсовая работа должна состоять из введения, двух-четырёх глав, заключения, списка литературы | 18 |  |
| 1. **Общая характеристика курсовой работы (проекта)** | | **24** |  |
| Полнота раскрытия темы | Тема курсовой работы (проекта) должна быть полностью раскрыта: подробно рассмотрены все аспекты данной темы | 14 |  |
| Логичность составления плана, изложения основных вопросов | Все вопросы должны быть рассмотрены логично, в соответствии с определённой последовательностью и взаимосвязью | 10 |  |
| 1. **Наличие элементов анализа** | | **22** |  |
| Наличие дискуссионных вопросов | В курсовой работе (проекте) должно быть отражено знание автором различных точек зрения по рассматриваемой теме | 6 |  |
| Наличие аргументированной точки зрения автора | В курсовой работе (проекте) должно присутствовать собственное, аргументированное мнение автора | 6 |  |
| Знание и отражение в работе изменений законодательства | Курсовая работа (проект) должна быть подготовлена с учётом последних изменений законодательства по рассматриваемой теме | 10 |  |
| 1. **Оформление курсовой работы (проекта) и соблюдение сроков** | | **21** |  |
| Аккуратность оформления | Курсовая работа (проект) должна быть аккуратно оформлена (с соблюдением предъявляемых требований) | 3 |  |
| Правильность оформления курсовой работы (проекта) | В курсовой работе (проекте) должны быть правильно оформлены цитаты, список использованной литературы, глоссарий и т.д. | 3 |  |
| Сроки представления | Курсовая работа (проект) отдельно по всем этапам и итого должна быть представлена в установленные сроки | 15 |  |
| 1. **Замечания по курсовой работе (проекту) и предварительная оценка работы** | | **-** |  |
| Замечания по тексту работы | См. замечания на стр.:  Общие замечания руководителя:  Оригинальность текста не менее 90% |  |  |
| Предварительная оценка | 86-100 баллов – «отлично»  70-85 баллов – «хорошо»  50-69 баллов – «удовлетворительно»  менее 50 баллов – «неудовлетворительно» |  |  |
| 1. **Защита курсовой работы** | | **15** |  |
| Вопросы которые необходимо подготовить к защите |  |  |  |
| Дополнительные вопросы заданные на защите |  |  |  |
| 1. **Оценка с учетом защиты** | |  |  |
| Дата и подпись руководителя | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Иванов И.И.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2015 года | | |

## Приложение 5. Шаблон для написания сценария для отдельного варианта использования

