

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

Разработка сетевых приложений и веб-сервисов

Методические указания по выполнению курсового проекта
для студентов всех форм обучения
направления подготовки 09.04.04 Программная инженерия
(профиль «Промышленное проектирование программных систем»)

Краснодар
2019

Составитель канд. техн. наук, доц. А.Г. Мурлин

Разработка сетевых приложений и веб-сервисов: методические указания по выполнению курсового проекта для студентов всех форм обучения направления подготовки 09.04.04 Программная инженерия (профиль «Промышленное проектирование программных систем»)/ Сост.: А.Г. Мурлин; Кубан. гос. технол. ун-т. Каф. информационных систем и программирования. – Краснодар, 2019. – 26 с.

Приведены рекомендации по содержанию, порядку выполнения курсового проекта, список рекомендуемой литературы. Методические указания содержат перечень тем курсового проекта и примеры оформления основных компонентов пояснительной записки.

Рецензенты: канд. тех. наук, руководитель отдела телекоммуникаций
Краснодарского регионального информационного центра
сети «Консультант-Плюс» Н.Ф. Григорьев
канд.тех.наук, доц. зав. кафедрой ИСП КубГТУ М.В. Янаева

Содержание

1 Требования к структуре и содержанию разделов курсового проекта	4
1.1 Общие положения	4
1.2 Структура пояснительной записки.....	6
2 Методические рекомендации по выполнению курсового проекта.....	8
3 Наполнение основных разделов пояснительной записки	10
4 Требования к оформлению пояснительной записки	14
4.1 Общие требования.....	14
4.2 Оформление содержания.....	14
4.3 Нумерация разделов пояснительной записки	15
4.4 Иллюстрации	15
4.5 Таблицы.....	16
4.6 Формулы и уравнения.....	17
4.7 Список использованных источников	17
4.8 Оформление приложений.....	18
4.9 Ссылки.....	18
5 Варианты задания на курсовое проектирование.....	19
Список рекомендуемой литературы.....	23
Приложение А	24
Приложение Б	25
Приложение В.....	26

1 Требования к структуре и содержанию разделов курсового проекта

1.1 Общие положения

Курс дисциплины «Разработка сетевых приложений и веб-сервисов» предполагает выполнение студентами курсового проекта. Курсовой проект – серьезное учебно-научное исследование, предполагающее творческий подход студента к проработке его содержания, тщательность и грамотность оформления. Студент, выполняя курсовой проект, систематизирует, углубляет и закрепляет знания, полученные в процессе обучения.

Цель курсового проекта – обучение навыкам проектирования и создания сетевых корпоративных приложений на языке Java, прежде всего это сфера веб-приложений и веб-сервисов; раскрытие роли значения сетевых технологий в развитии современного общества; приобретение студентами знаний о процессах обработки, передачи и хранения информации как в корпоративной, так и в глобальной сети Интернет; привитие навыков рационального использования ресурсов глобальной сети Интернет; привитие основных навыков программирования при создании клиент-серверных приложений, веб-сервисов и веб-приложений с помощью платформы Java EE (Java Enterprise Edition).

Java EE состоит из набора API и среды выполнения, объединяющая ниже перечисленные API.

Java Servlets. Сервлеты представляют специальные модули, которые обрабатывают запросы от пользователей и отправляют результат обработки.

JavaServer Pages (JSP). Также модули на стороне сервера, которые обрабатывают запросы. Удобны для генерации большого контента HTML. По сути представляют собой страницы с кодом HTML/JavaScript/CSS с вкраплениями кода на Java

Enterprise JavaBeans (EJB) представляют классы, которые хранят бизнес-логику.

Contexts and Dependency Injection (CDI) предоставляет механизм для внедрения и управления зависимостями в другие объекты.

JSON Processing (JSON-P) позволяет работать со строками JSON в Java.

JSON Binding (JSON-B) предоставляет функционал для сериализации и десериализации JSON в объекты Java.

WebSocket позволяет интегрировать WebSocket в приложения на Java.

Java Message Service (JMS) - API для пересылки сообщений между двумя и более клиентами.

Security API - API для стандартизации и упрощения задач обеспечения безопасности в приложениях на Java.

Java API for RESTful Web Services (JAX-RS) - API для применения архитектуры REST в приложениях.

JavaServer Faces (JSF) предоставляет возможности для создания пользовательского интерфейса на стороне сервера.

Эти и ряд других API собственно и образуют Java EE.

Предтечей Java EE был проект JPE Project, который стартовал в мае 1998 года. А в декабре 1999 года вышел релиз платформы Enterprise Java Platform (J2EE 1.2), которая объединяла такие компоненты как сервлеты, JSP, EJB, JMS. В 2006 году с выходом 5-й версии она была переименована в Java Enterprise Edition (JEE). С тех пор периодически выходят новые версии платформы. Последняя текущая версия - Java EE 8 вышла в сентябре 2017 года.

В 2017 году произошла новая веха в развитии платформы: Oracle передал контроль над развитием Java EE организации Eclipse Foundation. А в апреле 2018 года Java EE была переименована в Jakarta EE.

Официальный сайт платформы <https://jakarta.ee/>.

Задачи курсового проектирования:

- изучение студентами языка Java как средство реализации систем распределённой обработки и сетевого программирования;
- освоение способов разработки клиент-серверных приложений;
- изучение способов использования архитектуры Java Enterprise Edition (JavaEE) для разработки сетевых информационных систем и веб-сервисов;
- формирование представлений об информации и информационных процессах, информационных основах управления в глобальной сети Интернет;
- приобретение студентами основ практических разработки программ для сети Интернет и использования программно-инструментальные средства для анализа, обработки и хранения информации с использованием современных СУБД.

Проектирование должно выполняться на основе Java EE 8 либо более поздней версии. Для работы с Java EE нам необходима среда разработки или IDE. Есть различные среды разработки, которые ориентированы на корпоративную разработку под Java. Это IntelliJ IDEA, NetBeans и Eclipse.

Также для разработки понадобится веб-контейнер сервлетов Tomcat, который предназначен для работы с рядом технологий Java EE, в частности, с JSP, сервлетами и рядом других. Нередко Tomcat называют веб-сервером. Для установки Tomcat необходимо воспользоваться официальным сайтом данного контейнера <https://tomcat.apache.org/download-90.cgi>.

Итог курсового проекта – разработка сетевого приложения, веб-сервиса или веб-приложения.

Результат выполнения курсового проекта оформляется в виде пояснительной записки. Общий объем пояснительной записки не должен превышать 40 листов, в том числе введение - не более 2 листов. К пояснительной записке прилагается носитель с созданным сетевым приложением, базой данных, если она имеется) и иной необходимой информацией. Пояснительная записка к курсовому проекту должна содержать описание предметной области: основные процессы, используемые технологии и сетевые протоколы, средства разработки и сопровождения приложения. Выходные данные должны формироваться на основе потребностей проектируемой информационной структуры. В пояснительной записке необходимо привести модель разработанной базы данных и модель сетевого приложения, работающего с этой базой.

Записка иллюстрируется схемами, рисунками с копиями экрана, выходными документами, листингом программного кода приложения.

1.2 Структура пояснительной записки

Пояснительная записка должна быть оформлена на листах формата А4 (210 × 297 мм), имеющих поля. Все листы следует сброшюровать и пронумеровать.

Пояснительная записка к курсовому проекту должна содержать обязательные разделы представленные далее.

Титульный лист (приложение А).

Задание на курсовое проектирование (приложение Б), которое оформляется и подписывается преподавателем, студентом и заведующим в первую неделю семестра.

Реферат (приложение В), должен содержать сведения об объёме курсового проекта, количестве иллюстраций, таблиц, приложений, использованных источников и страниц. Перечень ключевых слов варьируется от 5 до 15. Объем реферата - не более 0,75 страницы.

Текст реферата отражает следующие аспекты содержания курсового проекта:

- цель работы;
- метод или методологию проведения работы;
- результаты работы;
- значимость работы;
- предложения по дальнейшему использованию программного продукта.

Содержание. Содержание пояснительной записки включает наименования структурных элементов: «Введение», заголовки разделов, подразделов и пунктов (если они имеют наименования) основной части.

«Заключение», «Список использованных источников», «Приложение». После заголовка каждого структурного элемента, раздела ставят отточие, затем указывают номер страницы.

Введение. Во введении описывается актуальность курсового проекта, обосновывается её связь с наукой и техникой в конкретной области. Включает краткое описание предметной области, формулируются цели, задачи, теоретическое и практическое значение проектирования и разработки сетевого приложения, предполагаемые инструментальные средства проектирования и реализации приложения, предполагаемые результаты и их применение в предметной области. Введение должно описывать актуальность и практическую значимость решаемой задачи.

Основная часть.

1 Анализ предметной области.

1.1 Общее описание предметной области.

1.2 Исследование существующих способов и средства построения сетевых приложений.

1.3 Описание механизмов, предназначенных для сетевого взаимодействия программ. Модели архитектуры разработки сетевого приложения.

2 Назначение проектируемого приложения.

3 Концептуальное проектирование сетевого приложения.

3.1 Описание функциональных особенностей и пользователей приложения (диаграмма прецедентов).

3.2 Взаимодействие классов приложения (диаграмма классов).

3.3 Описание последовательности взаимодействия объектов приложения в динамике (диаграмма последовательностей).

3.4 Описание состояний и активности объектов приложения (диаграмма состояний и диаграмма активности).

3.5 Описание рационально распределения компонентов системы по узлам сети (диаграмма развёртывания).

4 Обоснование выбора вида программного обеспечения.

5 Модель базы данных (при её наличии).

5.1 Модель СУБД приложения.

5.2 Модель функционирования СУБД приложения.

5.2 Механизмы работы со спецификацией СУБД приложения.

5.3 Технология проектирования и построения СУБД.

5.4 Диаграммы потоков данных.

6 Реализация сетевого приложения.

6.1 Подключение базы данных к приложению.

6.2 Схема функциональной структуры приложения.

6.3 Схема меню приложения.

6.4 Функции и процедуры обработки данных.

6.5 Описание работы с приложением.

6.7 Руководство пользователя.

Заключение. Заключение должно содержать общие выводы и предложения по использованию результатов, включая предложения по их реализации. В разделе перечисляются результаты проделанной работы. Дается заключение о качестве и полноте решения задачи разработки приложения. Высказываются соображения о направлениях развития разработанной системы и путях дальнейшего развития сетевого приложения. Следует указать социальную ценность выполненного исследования и его практическую значимость.

Список использованных источников. Список должен содержать сведения об источниках, использованных в ходе выполнения курсового проектирования. Источники в списке следует располагать в порядке появления ссылок на них в тексте пояснительной записки. Описания источников приводятся в соответствии с требованиями ГОСТ Р 7.0.5-2008 СИБИД. «Библиографическая ссылка. Общие требования и правила составления», и ГОСТ 7.82-2001 СИБИД. «Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления».

Приложения (листинг структуры файлов, листинг компонент реализованных форм, меню и т.д.). В приложения рекомендуется включать материалы, связанные с выполненным курсовым проектом, которые по каким-либо причинам не могут быть включены в основную часть. В приложения могут быть включены программный код и иллюстрации вспомогательного характера. Приложения оформляют как продолжение пояснительной записки на последующих ее страницах, располагая их в порядке появления ссылок в тексте.

2 Методические рекомендации по выполнению курсового проекта

В рамках выполнения курсового проекта предполагается разработка и реализация сетевых информационных систем, клиент-серверных и корпоративных приложений на языке Java, прежде всего это сфера веб-приложений и веб-сервисов, согласно выбранной теме. Выполнение анализа данных конкретной предметной области, формирование необходимой отчетной документации.

Тема курсового проекта выбирается студентом из перечня, приведенном в разделе «Варианты заданий на курсовое проектирование» и согласовывается с преподавателем. Также, по инициативе студента, может быть выдано персональное задание, которое соответствует уровню курсового проекта и задачам дисциплины «Разработка сетевых информационных систем». Тема курсового проекта закрепляется за студентом в начале семестра приказом директора института.

Для достижения поставленной цели курсового проектирования студенту необходимо решить указанные ниже задачи.

1. Выбрать тему курсового проекта.
2. Согласовать тему курсового проекта и составить план работ.
3. Выполнить проектирование и реализовать приложение.
4. Оформить пояснительную записку к курсовому проекту.
5. Составить презентацию для демонстрации работы.
6. Публично защитить результаты курсового проектирования.

Рекомендуемые операционные системы: операционные системы могут быть любого рода, так как используются технологии являются кроссплатформенными.

Рекомендуемые среды проектирования: IDEF/ Design, MS Visio, Erwin, BPwin.

Веб-сервер Apache Tomcat® 9.x.

Рекомендуемые СУБД: MySQL, Oracle, PostgreSQL и т.д.

Рекомендуемые инструментальные среды создания приложений:
IntelliJ IDEA Community Edition 2019.2
(<https://www.jetbrains.com/idea/download>) и Eclipse IDE 2019-09
(<https://www.eclipse.org/downloads/>).

Курсовой проект считается выполненным, если пояснительная записка прошла проверку на антиплагиат, нормоконтроль, а также есть работающая программа, реализующая задание на выполнение курсового проекта. Все этапы курсового проектирования должны быть завершены в сроки, указанные в графике выполнения курсового проекта. Нормоконтролер и члены комиссии, присутствующие при защите курсового проекта, назначаются на каждую дисциплину перед началом учебного семестра. Даты защиты так же определяются заранее. Поэтому необходимо спланировать свою учебную деятельность по выполнению курсового проекта.

Пояснительная записка может быть выполнена в виде текстового файла в формате *.docx, а презентация в формате *.pptx, могут быть так же использованы другие свободно распространяемые программные продукты, которые необходимы для обработки информации и дальнейшего представления ее в виде рисунков и таблиц.

После защиты курсового проекта члены комиссии ставят подписи и оценку на титульном листе распечатанной пояснительной записки. Титульный лист и лист задания на курсовой проект необходимо отсканировать в хорошем качестве, вставить в документ word пояснительной записки, преобразовать документ в pdf формат и отдать преподавателю, после чего он сможет поставить запись с оценкой в зачётную книжку. Это необходимо сделать с целью размещения каждого выполненного курсового проекта в портфолио соответствующего студента.

3 Наполнение основных разделов пояснительной записки

Пояснительная записка пишется и оформляется на протяжении всего выполнения курсового проекта. Наполнение разделов записки должно быть следующим.

Раздел «Анализ предметной области»

При анализе предметной области студент должен:

- собрать и обобщить материал, всесторонне характеризующий деятельность объекта исследования;
- ознакомиться с перспективами развития объекта исследования;
- обосновать необходимость применения средств обработки информации и/или управления;
- выявить возможности автоматизации информационных процессов для повышения эффективности, надежности и снижения трудоемкости работ;
- исследовать существующие модели архитектур сетевого приложений, способы и средства их построения;
- описать модель приложения и механизмов, предназначенных для сетевого взаимодействия программ.

В процессе анализа предметной области необходимо раскрыть вопросы необходимости разработки сетевого приложения и перспектив дальнейшего совершенствования существующей системы или технологии.

В результате выполнения анализа предметной области должно быть:

1. Получено подробное описание информации об объектах предметной области, которая требуется для решения конкретных задач.
2. Сформулированы конкретные задачи, которые будут решены, с использованием сетевых технологий с описанием методов и технологий используемых при разработке приложения.
3. Выполнено описание входных документов, которые служат основанием для работы приложения.
4. Выполнено описание выходных документов, которые должны генерироваться при работе системы.

Перед описанием документов необходимо привести представление основных процессов и построение диаграммы потоков данных.

Раздел «Назначение проектируемого приложения»

В данном разделе необходимо описать, с какой целью создаётся разрабатываемое приложение, каковы его основные функции, его необходимость при решении задач объекта исследования в выбранной предметной области.

Раздел «Концептуальное проектирование сетевого приложения»

Концептуальное моделирование сетевого приложения – это процесс проектирования объектной модели сетевого приложения, разработки архитектурных решений для приложения и спецификации системы. Построение модели процессов для описания предметной области, подлежащей автоматизации и информатизации. Первый этап процесса проектирования заключается в определении действующих лиц и функциональных особенностей приложения. Он заключается в создании диаграммы прецедентов приложения в соответствии с объектом исследования. Эта модель создается на основе информации, записанной в спецификациях требований пользователей. Концептуальное проектирование базы данных абсолютно не зависит от таких подробностей ее реализации, как тип используемых инструментальных средств, набор создаваемых прикладных программ, используемые языки программирования, тип выбранной вычислительной платформы, а также от любых других особенностей физической реализации.

При разработке модель постоянно подвергается тестированию и проверке на соответствие требованиям пользователей. Созданная функциональная диаграмма является источником информации для последующих этапов проектирования приложения.

Информационная система является олицетворением модели, и правильность ее функционирования зависит от точности и непротиворечивости модели, построенной на этапе проектирования. При разработке приложения особенно важно осознавать, какие данные необходимо хранить и как они должны относиться друг к другу. Особое внимание следует уделить документированию принятых решений. При создании моделей следует быть особенно внимательным, поскольку исправление ошибок, допущенных на этом этапе, требует самых больших затрат.

Построение логической модели системы в виде диаграммы классов является центральным местом в проектировании программного обеспечения. Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Концептуальная модель характеризует статические свойства разрабатываемого программного обеспечения. Для описания особенностей его поведения, т. е. возможных действий системы, целесообразно использовать диаграммы последовательностей системы, системные события, системные операции, диаграммы деятельности, а при необходимости и диаграммы состояний объектов. Диаграмма последовательностей системы — графическая модель, которая для определённого сценария варианта использования показывает динамику взаимодействия объектов во времени. Для моделирования процесса

выполнения операций используются так называемые диаграммы деятельности. На этапе анализа требований и уточнения спецификаций диаграммы деятельности позволяют конкретизировать основные функции разрабатываемого программного обеспечения. Диаграмма состояний позволяет описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Корпоративные приложения часто требуют для своей работы некоторой ИТ-инфраструктуры, хранят информацию в базах данных, расположенных где-то на серверах компании, вызывают веб-сервисы, используют общие ресурсы и т. д. В таких случаях неплохо бы иметь графическое представление инфраструктуры, на которую будет развёрнуто приложение. Вот для этого-то и нужны диаграммы развёртывания, которые иногда называют диаграммами размещения.

Раздел «Обоснование выбора вида программного обеспечения»

В данном разделе кратко характеризуется современное состояние средств информационных сетевых технологий, обосновывается выбор вида информационной технологии для выполнения курсового проекта.

Далее проводится обзор существующих средств, относящихся к выбранному виду информационной технологии. Обзор должен завершиться выводами о возможности использования известных решений по теме курсового проекта или о необходимости проектирования оригинальных решений. Анализируется соотношение положительных и отрицательных качеств выбранного для выполнения курсового проекта программного обеспечения, языков программирования. В разделе обязательно должны присутствовать ссылки на источники информации (книги, журналы, каталоги, техническую документацию, патенты и т. п.). Особое внимание рекомендуется обратить на публикации в журналах и других периодических изданиях по теме курсового проекта.

Раздел «Модель базы данных»

Одним из основных факторов, влияющих на производительность программ, которые взаимодействуют с базой данных, является способ хранения и доступа к данным. Обычно в дополнение к специализированным методам доступа в рамках внешней модели СУБД используют несколько методов доступа к внутренней модели. Внутренняя модель – это модель физическая, внешняя же отражает представление пользователя о базе данных. Физический уровень данных – это по существу отображение системного каталога, который зависит от конкретной реализации системы управления базой данных. На физическом уровне объекты базы данных могут называться так, как того требуют ограничения системы управления базой данных.

После выбора используемого программного обеспечения возможно описание физической модели базы данных: описание каждой таблицы с

указанием типа, параметров и примечаний для каждого поля, полей с первичными и внешними ключами.

Раздел «Реализация сетевого приложения»

При представлении физической реализации решения конкретной задачи необходимо описать словесно и представить графически алгоритм решения спроектированной задачи, представить и пояснить вид окон и элементов управления для всех режимов работы (ввод исходных данных, вывод результатов работы).

При проектировании сетевого приложения необходимо в первую очередь продумать внешний вид и дизайн сайта. Далее выполняют вёрстку сайта и создание html-страницы или html-шаблона. Вёрстка – это процесс размещения на странице различных элементов – текста, графиков и т. п. Далее выполняют программирование сайта, добавляя к нему необходимые функции.

Проектируемое приложение должно обеспечить выполнение следующих требований:

1. Интерфейс должен включать в себя средства редактирования всех используемых для расчета данных и быть простым и понятным в работе не только для разработчика, но и для обычного пользователя.

2. Система должна обладать максимальной гибкостью — возможностью изменения любых настроек и параметров программы. И хотя данное требование в основном реализуется при действии программы, основа этого должна быть заложена уже на этапе проектирования.

3. Необходимо ввести авторизацию и четкое разграничение прав доступа, отслеживать любое изменение данных с возможностью выявления даты и ответственного за введенные изменения.

4. Ввод данных для расчета должен быть максимально автоматизирован. Необходимо предусмотреть защиту от некорректного ввода данных во всех формах интерфейса. Использовать методы верификации и валидации данных. Если пользователь не имеет представления о корректности введенных данных, то в результате возникает множество ошибок, которые приводят к неправильному результату.

Раздел содержит описание модели функционирования приложения, диаграмм потоков данных, модульной структуры приложения (модули, модули данных). Приводится схема меню и эскизы оконных форм с аргументацией выбора типа диалога, интерфейса или окна, элементов управления. Согласно структуре сайта, приводятся копии окна браузера выполняющегося приложения: главное окно, окно заполнения данных, окно генерирования запросов, окно формирования отчетов, окно диаграмм, главное меню.

Для демонстрации формирования и выполнения запросов необходимо описать исходный код приложения и привести изображение результатов выполнения запросов.

4 Требования к оформлению пояснительной записки

4.1 Общие требования

Пояснительная записка должна быть оформлена на листах формата А4. Цвет шрифта должен быть черным, гарнитура шрифта Times New Roman, размер шрифта - 14 пунктов, межстрочный интервал полуторный, абзацный отступ - 1,5 см. Текст ПЗ следует печатать, соблюдая поля следующих размеров:

- правое - 15 мм;
- левое - 30 мм;
- верхнее - 15 мм;
- нижнее - 25 мм.

Общий объем пояснительной записки не должен превышать 40 листов, в том числе введение – не более 2 листов. К пояснительной записке прилагается носитель с созданным программным обеспечением и иной необходимой информацией.

Записка иллюстрируется схемами, копиями экрана, выходными документами, листингом программного кода приложения. Все листы пояснительной записки следует пронумеровать.

Нумерация страниц начинается с титульного листа и заканчивается последним листом пояснительной записки. Номер страницы проставляется в центре нижней части листа без точки. На титульном листе номер страницы не указывается. Заголовки глав, пунктов и подпунктов должны быть краткими, при этом заголовки должны точно отражать содержание соответствующего раздела. Заголовки глав записывают в виде предложения с абзацного отступа, с прописной буквы, без точки в конце, не подчеркивая. Заголовки пунктов и подпунктов записывают с абзацного отступа строчными буквами (кроме первой прописной). Переносы слов в заголовках не допускаются. Точка в конце заголовка не ставится. Каждую новую главу пояснительной записки и каждый структурный элемент рекомендуется начинать с нового листа. Титульный лист является первым листом пояснительной записки. Титульный лист выполняется на листе формата А4 по форме, приведенной в приложении А.

4.2 Оформление содержания

Слово "Содержание" записывают в виде заголовка (симметрично тексту) прописными буквами.

Содержание включает:

- введение;
- номера и названия разделов, подразделов, пунктов и подпунктов строчными буквами кроме первой прописной;
- заключение;
- список использованных источников;
- номера и полные названия всех приложений.

Для каждого названия указывается номер страницы, на которой оно находится. Название и номер страницы разделяются отточием. Номера страниц выровнены по правому краю.

4.3 Нумерация разделов пояснительной записки

Разделы нумеруются в пределах пояснительной записки порядковыми номерами арабскими цифрами. Пункты нумеруются по порядку в пределах раздела, например, 3.2 – пункт 2 раздела 3.

Подпункты нумеруются в пределах пункта, например, 3.2.1.

Введение, заключение и список использованных источников не нумеруются. Внутри пунктов и подпунктов могут быть перечисления. Перед каждым перечислением следует ставить дефис или, при необходимости ссылки в тексте на одно из перечислений, строчную букву (за исключением з, о, г, ь, й, ы, ъ), после которой ставится скобка. Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа.

4.4 Иллюстрации

Иллюстрации (графики, схемы, диаграммы) следует располагать непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. На все иллюстрации должны быть даны ссылки.

Чертежи, графики, диаграммы, схемы должны соответствовать требованиям Единой системы конструкторской документации (ЕСКД).

Иллюстрации, за исключением иллюстрации приложений, нумеруются арабскими цифрами сквозной нумерацией. Допускается нумерация в пределах раздела (разбиение на пункты во внимание не принимается). Например, рисунок 3.1 — рисунок первый в третьем разделе. В общем случае рисунок может содержать:

- поясняющие надписи, расположенные под рисунком (могут отсутствовать);
- номер рисунка и название, расположенные под пояснительными данными по центру следующим образом: Рисунок 2.1 – Функциональная схема.

Если рисунок располагается на нескольких листах, то на каждом последующем листе указывается номер рисунка, за которым следует слово

"Продолжение". Например, Рисунок 2.1 Продолжение. Точки после номера и названия рисунка не ставятся.

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

При ссылках на иллюстрации следует писать «... в соответствии с рисунком 2».

4.5 Таблицы

Таблицу следует располагать непосредственно после текста, в котором она упоминается впервые, или на следующей странице. На все таблицы должны быть ссылки. При ссылке следует писать слово «таблица» с указанием ее номера. Название таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тир, например: Таблица 2.1 – Перечень элементов. Точка после названия не ставится. При переносе части таблицы на другой лист слово «Таблица», ее номер и название указывают один раз над первой частью таблицы, нижнюю горизонтальную черту, ограничивающую таблицу, не проводят. Над другими частями пишут слово «Продолжение» и указывают номер таблицы, например, «Продолжение таблицы 1». Точка после номера не ставится. Заголовки граф таблицы начинают с прописной буквы, а подзаголовки со строчной, если они составляют одно предложение с заголовком. В конце заголовков и подзаголовков знаки препинания не ставят. Заголовки указывают в единственном числе. Диагональное деление головки таблицы не допускается. При переносе таблицы на следующую страницу и для облегчения ссылок в тексте записки допускается нумерация граф. Единицы измерения физических величин указываются через запятую после заголовка строки или заголовка (подзаголовка) графы.

Повторяющийся в графе таблицы текст, состоящий из одного слова, допускается заменять кавычками. Если повторяющийся текст состоит из нескольких слов, то при первом повторении его заменяют словами "То же", а далее кавычками. Цифровые и подобные им данные заменять кавычками нельзя. Если какие-либо данные в таблице не приводят, то в соответствующей графе ставят прочерк.

Числовые значения величин в одной графе должны иметь, как правило, одинаковое количество десятичных знаков.

Таблицы, за исключением таблиц приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Допускается нумерация в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

Таблицы каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения, например: «Таблица В.1».

4.6 Формулы и уравнения

Формула располагается в отдельной строке (строках) текста. Выше и ниже каждой формулы должно быть оставлено не менее одной свободной строки. Обозначения расшифровываются сразу после формулы в последующих строках текста в порядке появления обозначений в формуле. При этом пояснение для каждого обозначения начинается с новой строки, в первой строке перед обозначением пишется слово "где".

Пример:

$$I_n = \sqrt{\sum_{i=1}^n k_i^2}, \quad (2.1)$$

где k_i - i – й элемент последовательности k .

Формулы, на которые есть ссылки в тексте, нумеруются арабскими цифрами сквозной нумерацией или в пределах раздела. Номер заключается в круглые скобки и ставится справа от формулы в последней (или единственной) строке, занимаемой формулой. Формулы, помещаемые в приложениях, должны нумероваться отдельной нумерацией с добавлением обозначения приложения, например: (А.1).

Ссылки в тексте на порядковые номера формул дают в скобках.

4.7 Список использованных источников

Использованные при написании курсового проекта источники обязательно должны быть указаны в списке с обязательной ссылкой в тексте. Количество использованных источников должно быть не менее 5. Источники могут быть расположены в алфавитном порядке или порядке их упоминания в тексте. Источником может быть книга, статья в периодическом издании, отчет о научно–исследовательской работе, методические указания, учебные пособия, стандарты, патенты.

На каждый источник, приведённый в списке источников, должна быть, по крайней мере, одна ссылка в тексте пояснительной записки. При ссылке в тексте на источник ставится его порядковый номер по списку использованных источников, заключённый в квадратные скобки. При необходимости ссылки на конкретный фрагмент текста источника, в квадратных скобках проставляют, кроме номера источника, номер страницы, на которой помещён фрагмент. При ссылке на статьи из периодических изданий указание страниц обязательно. Возможно указание

ссылок на WEB-сайты авторитетных в области разработки интернет-приложений фирм.

4.8 Оформление приложений

Вспомогательный материал, необходимый для полноты изложения результатов курсового проектирования (иллюстрации, таблицы или текст вспомогательного характера) допускается оформлять в виде приложений. В тексте должны быть ссылки на все приложения. Приложения располагают в порядке появления ссылок на них. Каждое приложение должно начинаться с нового листа с указанием наверху посередине страницы слова «ПРИЛОЖЕНИЕ» и его обозначения.

Приложения оформляют как продолжение пояснительной записки на последующих ее страницах. Для приложений можно использовать кегль 8-10.

Приложение должно иметь заголовок, который располагается симметрично относительно текста с прописной буквы отдельной строкой.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь. Если в документе одно приложение, оно обозначается «ПРИЛОЖЕНИЕ А»

Ниже заголовка располагается текст приложения. Текст приложения может состоять из разделов, пунктов и подпунктов, которые нумеруются в пределах данного приложения. Рисунки, таблицы и формулы, помещаемые в приложении, нумеруют в пределах данного приложения, например: Рисунок Б.1 — рисунок 1 в приложении Б.

Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

4.9 Ссылки

В пояснительной записке можно использовать ссылки на любые рисунки, таблицы, формулы, приложения, литературные источники, которые приведены в записке. Рисунки, таблицы, формулы располагаются сразу после появления первой ссылки на них, то есть на текущем или следующем листе записки. Порядок номеров приложений и литературных ссылок должен соответствовать порядку появления ссылок на них. При ссылке на литературный источник указывается его порядковый номер, заключенный в квадратные скобки, например, [4] или [4, 5, 6]. При первой ссылке на рисунок пишется, например, рисунок 1.4 или (рисунок 1.4). При повторной ссылке на рисунок пишется, например, см. рисунок 1.4 или (см. рисунок 1.4). При первой ссылке на таблицу пишется, например, в таблице 2.3 или (таблица 2.3). При повторной ссылке добавляется слово "см.", например, см. таблицу 2.4 или (см. таблицу 3.1). Ссылки в тексте на номер формулы дают в скобках, например, ... в формуле (2.5). При ссылке на

приложение пишется полностью слово «приложение» и указывается его номер, например, «... в приложении А» или (приложение Б).

5 Варианты задания на курсовое проектирование

Тематика курсового проекта может быть связана с разделом выпускной квалификационной работы. В таком случае тема курсового проекта определяется индивидуально для каждого студента в соответствии с научным исследованием, проводимым в рамках ВКР. При разработке сетевого приложения необходимо использовать язык JAVA.

Примерный перечень основных тем курсовых проектов.

1. Разработка сетевой многопоточковой программы «On-line конференция».

Разработать сетевую многопоточковую программу «On-line конференция». Количество участников конференции неограниченно. Для каждого участника, создается свой рабочий поток. Все сообщения отображаются в графическом окне клиентов. Первый участник задаёт тему конференции.

Функции сервера:

- создает сокетное соединение и поток клиента;
- регистрирует участника конференции;
- пересылает сообщения всем участникам;
- останавливает потоки клиентов по желанию участника,

задавшему тему конференции.

Функции клиента:

- создает сокетное соединение с сервером;
- регистрируется;
- задает тему конференции (если он первый) и/или отправляет свое сообщение всем участникам;
- подписывается на определенные конференции;
- отправляет сообщение об окончании конференции серверу.

2. Разработка приложения ленты новостей.

Предусмотреть 2 режима работы: администратор, пользователь

Новое сообщение (новость) может добавить, удалить и редактировать только администратор. На странице отображается новость только в виде нескольких слов. Остальная информация выводится в отдельном окне, где можно подробнее прочитать новость. Остальные пользователи могут только просматривать новости и оставлять небольшие комментарии. Комментарии можно просматривать.

Задание № 3. Разработка программы голосования на сайте.

Предусмотреть 2 режима работы: администратор, пользователь.

Администратор ведет учет голосований: задание темы голосования и вариантов ответов на них. Учесть возможность просмотра, редактирования и удаления.

Пользователь: выбор темы и варианта ответа.

Темы голосований и результаты хранятся в базе данных.

По окончанию ввода голосования на экране обновляется статистика результатов голосования по теме.

4. Организация диалога нескольких пользователей (чат) на Java. Иметь возможность создания приватного и группового диалога.

5. Разработка гостевой книги.

Регистрация пользователя на сайте.

Пользователь, открыв гостевую книгу, видит все сообщения в убывающем порядке по дате (предусмотрите постраничный вывод на экран). Оставить сообщение может только зарегистрированный пользователь. Для него на этой же странице есть форма для ввода нового сообщения. Сообщение хранится в базе данных. Записывается следующая информация:

- дата и время отправки сообщения;
- e-mail пользователя;
- само сообщение.

6. Приложение, представляющее домашнюю страницу преподавателя кафедры ИСП. В окно приложения введено несколько панелей, которые поочередно выходят на первый план. Первая панель - заглавная страница содержит фотографию преподавателя и его краткое резюме. Последующие панели содержат информацию о читаемых курсах и т.д. Иметь возможность добавлять учебные и методические материалы преподавателем. Материалы можно просмотреть и скачать студентом. Хранить статистику закачек. Информация сохраняется в базе данных. Использовать графический интерфейс пользователя (GUI) при оформлении приложения.

7. Контроль успеваемости студентов. Разработать программу, реализующую страницу приема заданий по дисциплине. Содержит таблицу со списком студентов и количеством заданий по дисциплине. Студенты могут только посмотреть текущее состояние дел. Преподаватель, после ввода пароля, может устанавливать факт сдачи задания. Должна быть предусмотрена возможность сортировки списка студентов по алфавиту или по рейтингу. Предусмотреть возможность проставления экзаменационной оценки студентам. Дисциплина и количество контрольных точек вводятся преподавателем. Информация хранится в базе данных.

8. Создать «клиент-серверное» приложение «Электронная система учета пункта проката видео». Название фильма, стоимость проката, даты выдачи и возврата предмета, фамилия. Несданные предметы – пустая дата

возврата (или выделение красным цветом). Подсчет дохода пункта за заданный год/месяц. Поиск по фамилии, по названию фильма. Все фильмы хранятся в справочнике и разбиты по жанрам в файлах.

9. Создать апплет двойного назначения, представляющий кафедру ИСП, в окне отображать несколько панелей, которые поочередно выходят на первый план. Первая панель - заглавная страница кафедры, несколько следующих - рекламная информация о кафедре. Последующие панели содержат информацию о преподавателях кафедры, их фото и читаемые дисциплины. После поочередного вывода всех панелей процесс повторяется вновь. Так же иметь возможность работы с панелями по щелчку мыши. Информация хранится в базе данных

10. Разработать клиент-серверное, автоматизирующий продажу билетов в кинотеатре. Данные продаж сохранять в базе данных. Реализовать различные запросы к базе данных и отображение результатов в удобной форме.

11. Разработка и реализация информационной системы «Электронная библиотека». Каталог электронных изданий хранится в базе данных на сервере. Имеется возможность поиска нужных изданий и их чтение на компьютере клиента.

12. Автоматизированная клиент-серверная справочная система «Домашняя аудиотека» на Java. Иметь возможность прослушать найденную запись.

13. Автоматизированная система «Аукцион продаж произведений искусства» на Java. Информацию сохранять в базе данных. Иметь следующие возможности:

Регистрация участников.

Регистрация произведения искусства, выставленного на продажу. Деление по категориям.

Даты проведения аукционов.

Получения прибыли от продаж.

Рассылка постоянным клиентам объявлений о дате аукциона.

14. Разработать многопоточковую программу, моделирующую работу товарного склада. Программа, реализует модель работы склада, отвечающего за хранение и продажу некоторого товара (одного). Склад содержит N помещений, каждый из которых может хранить определённое количество единиц товара. Поступающий товар помещается в помещения специальным погрузчиком. За товаром прибыло K покупателей, каждому из которых требуется по L_k единиц товара. Площадка перед складом мала и на ней может в один момент времени находиться либо погрузчик, либо один из покупателей. Если покупателям требуется больше товара, чем имеется на складе, то они ждут новых поступлений, периодически проверяя склад. Время работы склада ограничено.

15. Разработка «Интернет-магазина»

Клиентская часть:

Отображение каталога товаров по разделам и подразделам (в виде дерева).

Реализация "корзины заказа" - отображение выбранных товаров и их количества с пересчетом суммы заказа.

Регистрация и идентификация заказчика.

Формирование счетов на оплату.

Административная часть:

Администрирование каталога товаров: просмотр информации о товаре, добавление, удаление товаров и подразделов, редактирование характеристик и фотографий, изменение структуры каталога.

Фиксация в базе и администрирование заказов: просмотр информации о заказе и заказчике, добавление, удаление заказов, редактирование параметров.

Фиксация в базе и администрирование пользователей: просмотр информации о зарегистрированных пользователях, добавление, удаление пользователей, редактирование контактной информации о них.

Перечень дополнительных обобщённых тем курсовых проектов.

1 Разработка web-приложения с использованием средств CGI-программирования и Java-апплетов.

2 Разработка веб-приложения для контроля сетевой активности предприятия.

3 Разработка Web-приложений с применением Java Servlet-технологии.

4 Разработка веб приложений с использованием динамических веб-страниц Java Server Pages (JSP).

5 Разработка серверных приложений с использованием Node JS.

Список рекомендуемой литературы

- 1 Вязовик, Н.А. Программирование на Java : учебное пособие / Н.А. Вязовик. — 2-е изд. — Москва : ИНТУИТ, 2016. — 603 с. — Текст : электронный // Электронно-библиотечная система «Лань» : [сайт]. — URL: <https://e.lanbook.com/book/100405>
- 2 Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6. - Режим доступа: <http://znanium.com/bookread2.php?book=1020593>
- 3 Дубаков, А.А. Введение в объектно-ориентированное программирование на Java : учебное пособие / А.А. Дубаков. — Санкт-Петербург : НИУ ИТМО, 2016. — 250 с. — Текст : электронный // Электронно-библиотечная система «Лань» : [сайт]. — URL: <https://e.lanbook.com/book/110468>
- 4 Интернет-программирование на Java: Пособие / Будилов В.А. - СПб:БХВ-Петербург, 2014. - 698 с. ISBN 978-5-9775-1931-1 - Режим доступа: <http://znanium.com/bookread2.php?book=940239>
- 5 Программирование на языке Java. Конспект лекций : учебное пособие / А.В. Гаврилов, С.В. Клименков, А.Е. Харитонов, Е.А. Цопа. — Санкт-Петербург : НИУ ИТМО, 2015. — 126 с. — Текст : электронный // Электронно-библиотечная система «Лань» : [сайт]. — URL: <https://e.lanbook.com/book/91488>
- 6 Хеффельфингер, Д. Разработка приложений Java EE 7 в NetBeans 8 / Дэвид Хеффельфингер ; пер. с англ. А.Н. Киселева. - Москва : ДМК Пресс, 2016. - 348 с. - ISBN 978-5-97060-329-1. - Текст : электронный. - URL: <https://new.znanium.com/read?id=341214>

Приложение А
Форма титульного листа курсового проекта

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО КубГТУ)

Институт _____

Кафедра _____

Направление подготовки _____

Профиль _____

КУРСОВОЙ ПРОЕКТ

по дисциплине _____
(наименование дисциплины)

на тему: «_____»
(тема курсового проекта)

Выполнил (а) студент (ка) _____ курса _____ группы _____
(ф.и.о.)

Допущен к защите _____

Руководитель (нормоконтролер) работы _____

Защищен _____ Оценка _____
(дата)

Члены комиссии _____

(подпись, дата, расшифровка подписи)

Краснодар
20__

Приложение Б

Форма листа задания на курсовое проектирование

ФГБОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО КубГТУ)

Институт _____

Кафедра _____

Направление подготовки _____

Профиль _____

УТВЕРЖДАЮ

Зав. кафедрой _____ М.В. Янаева.

«___» _____ 20__ г.

ЗАДАНИЕ

на курсовую работу

Студенту: _____ группы _____ курса
(Ф.И.О.) (№ группы и курса)

Тема проекта: «_____»
(утверждена указанием директора института № ___ от ____ 20 ____)

План работы:

1. _____
2. _____
3. _____

Объем работы:

а) пояснительная записка _____ с.

б) иллюстративная часть _____ лист (-ов)

Рекомендуемая литература

1. _____
2. _____
3. _____

Срок выполнения: с «___» _____ по «___» _____ 20__ г.

Срок защиты: «___» _____ 20__ г.

Дата выдачи задания: «___» _____ 20__ г.

Дата сдачи работы на кафедру: «___» _____ 20__ г.

Руководитель работы _____ Фамилия И.О.
(должность, подпись,)

Задание принял студент _____ Фамилия И.О.
(подпись)

Приложение В

Пример оформления реферата

Реферат

Курсовой проект: 30 страниц, 20 рисунков, 5 таблиц, 10 источников, 2 приложений, 10 иллюстративная часть.

САЙТ, ВЕБ-ПРИЛОЖЕНИЕ, ИНТЕРНЕТ ТЕХНОЛОГИИ, КЛИЕНТ-СЕРВЕР, АДМИНИСТРИРОВАНИЕ, БАЗЫ ДАННЫХ, УПРАВЛЕНИЕ ПРОЕКТАМИ, СИСТЕМА, ДИАГРАММА КЛАССОВ, ДИАГРАММА ВЗАИМОДЕЙСТВИЯ ДАННЫХ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.

Объектом исследования является деятельность предприятия, предоставляющего услуги для организации работы с клиентами сферы розничной торговли и бизнеса.

Целью работы является увеличение эффективности работы с клиентами за счёт обеспечения на техническом уровне следующих условий: интерактивного общения, стандартизации процессов работы с клиентами; наличия единого хранилища информации; оперативным и бухгалтерским учетом; единой системой безопасности; обеспечения инструментов анализа динамики продаж; интегрирование различных видов деятельности предприятия; обеспечение взаимодействия между организациями.

Основные полученные результаты:

- проведён анализ процессов работы с клиентами, позволяющих разработать интернет приложение для организации взаимодействия с ними;
- разработана концептуальная, логическая и физическая модель базы данных;
- спроектировано и разработано веб-приложение для взаимодействия с клиентами предприятия.

Об эффективности внедрения можно судить по изменению таких показателей, как время обработки обращения клиента, время заказа и приемки товара на головной склад и/или удаленный филиал, время на подготовку отчётов о динамике продаж и динамике клиентской базы, а также анализа продаж всего представленного ассортимента.