

Министерство науки и высшего образования РФ
ФГБОУ ВО «Ульяновский государственный университет»
Факультет математики, информационных и авиационных технологий

Жаркова Г.А.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ
РАБОТЫ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ «МЕТОДЫ ОБЪЕКТНО-
ОРИЕНТРОВАННОГО ПРОГРАММИРОВАНИЯ»**

Для студентов магистратуры по направлению подготовки 02.04.03 -
«Математическое обеспечение и администрирование информационных
систем», (профиль/специализация) Технология программирования
очной формы обучения

Методические указания для выполнения курсовой работы студентов по дисциплине «Методы объектно-ориентированного программирования» / составитель: Г.А. Жаркова. - Ульяновск: УлГУ, 2019. Настоящие методические указания предназначены для магистратуры по направлению подготовки 02.04.03 - «Математическое обеспечение и администрирование информационных систем», (профиль/специализация) Технология программирования очной формы обучения

В работе приведены основные темы курсовых работ, правила оформления текста, таблиц, рисунков. Студентам очной формы обучения они будут полезны при подготовке курсовой и выпускной квалификационной работы.

Рекомендованы к введению в образовательный процесс Ученым советом факультета математики, информационных и авиационных технологий УлГУ (протокол № 2/19 от 19 марта 2019г.).

Содержание

Введение	3
2. Требования к курсовой работе	6
3. Требования к оформлению курсовой работы	9
4. Оформление списка использованной	14
5. Оценивание курсовой работы.....	22
Приложение 1. Образец титульного листа курсовой работы	24
Приложение 2. Образец содержания курсовой работы.....	25
Приложение 3. Примерные темы курсовых работ.....	26

Введение

Курсовой работе как одной из форм самостоятельной учебно-исследовательской работы отводится особая роль при формировании компетенции будущего специалиста и бакалавра. Курсовая работа по данной дисциплине представляет собой самостоятельную аналитическую и практическую работу студента и показывает знания и навыки работы, приобретенные в ходе изучения дисциплины Методы объектно-ориентированного программирования.

Важным фактором подготовки специалистов в области информационных технологий является умение программировать, используя современные языки, включающие объектные возможности, знакомство с основными методами и современными технологиями программирования, в том числе с использованием объектных библиотек конкретных языков.

Целью курсового проектирования является развитие данных навыков в ходе решения конкретной практической задачи – разработки оконного приложения для тестирования спроектированной системы классов. При этом студентам предоставляется право выбора конкретного языка и средства разработки. Целью такого подхода является развитие способностей к обоснованному принятию самостоятельных решений в ходе проектной работы и получение базового опыта разработки информационных систем. В ходе выполнения курсовой работы студенты должны практически освоить общий методологический подход, используемый при проектировании и программной реализации системы классов, соответствующей объектно-ориентированной парадигме программирования. Создаваемая система классов описывает (моделирует) определённую предметную область и может служить основой для полноценной информационной системы, решающей задачи данной области. Спроектированная система классов должна быть не только реализована в виде программы на одном из Си-подобных объектноориентированных языков, но и протестирована.

Методические рекомендации по выполнению курсовой работы предназначены для студентов по направлению подготовки бакалавриата 09.03.03 «Прикладная информатика», (профиль/специализация) Информационная сфера.

Процесс изучения дисциплины направлен на формирование следующих компетенций, представленных в таблице. Перечень планируемых результатов обучения по дисциплине «Методы объектно-ориентированного программирования», соотнесенных с планируемыми результатами освоения образовательной программы

Таблица 1. Перечень планируемых результатов обучения по дисциплине

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ОПК-2 способностью проектировать, разрабатывать и внедрять программные продукты и программные комплексы различного назначения	<p>Знать: основные понятия и принципы структурного и объектно-ориентированного проектирования; методы внедрения программных систем; инструментальные средства поддержания процесса разработки.</p> <p>Уметь: исследовать различные типы задач, применять современные средства разработки программного обеспечения; использовать методологии проектирования информационных систем; применять при решении поставленных задач современные компьютерные и информационные технологии.</p> <p>Владеть: методологией объектно-ориентированного проектирования и программирования при создании программных продуктов.</p>
ПК-4 способностью применять современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	<p>Знать: современные информационные технологии, принципы их работы, возможности для реализации информационных систем.</p> <p>Уметь: исследовать различные типы программных систем, проводить оценку качества разработанного программного обеспечения; применять полученные знания для проектирования и разработки информационных систем</p> <p>Владеть: методологией объектно-ориентированного программирования; методами оценки качества и анализа эффективности программного обеспечения.</p>
ПК-5 способностью использовать основные	<p>Знать: методы и средства автоматизации проектирования информационных систем;</p>

<p>методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов</p>	<p>методологии структурного и объектно-ориентированного проектирования и разработки программных продуктов; способы оценки качества программного обеспечения.</p> <p>Уметь: использовать методологии проектирования информационных систем, инструментальные средства разработки программных продуктов, сопровождать и модернизировать программный продукт.</p> <p>Владеть: методами и средствами автоматизации проектирования, оценки качества, сопровождения и администрирования информационных систем.</p>
<p>ПК-7 способностью использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений</p>	<p>Знать: основные понятия и принципы объектно-ориентированного, обобщённого программирования; знать инструментальные средства поддержания процесса разработки.</p> <p>Уметь: исследовать различные типы задач, применять современные методологии разработки программного обеспечения; применять при решении поставленных задач современные компьютерные и информационные технологии.</p> <p>Владеть: методологией объектно-ориентированного проектирования и программирования при создании программных продуктов.</p>

2. Требования к курсовой работе

Тематика курсовых работ разрабатывается, утверждается и ежегодно актуализируется кафедрой информационных технологий. В рамках предлагаемой тематики студентам предоставляется право выбора темы, но ее необходимо согласовать с преподавателем. Студент может самостоятельно предложить тему курсовой работы с обоснованием ее целесообразности.

Тематика курсовых работ должна:

- соответствовать задачам профессиональной подготовки бакалавров и специалистов;

- быть проблемной и актуальной;
- приобщать студентов к научно-исследовательским направлениям, разрабатываемым на кафедре;
- учитывать практическую направленность творческой работы над самостоятельно выбранной темой.

Курсовая работа должны соответствовать нормам научной речи и отвечать требованиям, предъявляемым к оформлению студенческих научных работ. Курсовая работа как массовая форма учебно-исследовательской работы студента и важное средство контроля самостоятельной работы студентов в любой области специальных знаний предполагает решение следующих задач:

- закрепление, углубление и расширение научных знаний студентов по избранной дисциплине;
- формирование исследовательских умений студентов: формулировка проблемы исследования, анализ различных подходов к решению проблемы; формулировка цели, объекта, предмета и определение задач работы, описание нестандартных научно обоснованных решений, сопоставление цели и результатов работы, формулировка выводов и т.д.;
- формирование умений использования различных методов анализа, сравнения, обобщения, классификации;
- закрепление практических навыков работы по созданию самостоятельного программного продукта. В ходе выполнения курсовой работы студент обязан:
 - научиться пользоваться библиографическими указателями и грамотно составлять библиографические списки литературы;
 - изучить и осмыслить определенный круг научной литературы, на основе анализа которой сделать обстоятельный обзор по избранной проблеме;
 - самостоятельно собрать и, используя методику научного исследования, проанализировать материал по теме.

- на основе изучения литературы собственного анализа и практического опыта разработать программный продукт, провести его тестирование;
- проявить достаточно высокий опыт общей и профессиональной культуры.

В соответствии с поставленными задачами этапы выполнения курсовой работы имеют определенную последовательность:

1. Анализ задания. Описание предметной области.
2. Разработка библиотеки классов:
 - 2.1. Определение набора классов, их свойств и методов, иерархии наследования.
 - 2.2. Формализация описания классов в виде диаграммы классов.
 - 2.3. Выбор средств реализации библиотеки классов – языка программирования и среды разработки.
 - 2.4. Программная реализация классов.
3. Разработка оконного приложения для тестирования системы классов:
 - 3.1. Определение методики тестирования для демонстрации использования всех методов созданных классов.
 - 3.2. Проектирование тестового оконного приложения.
 - 3.3. Программная реализация тестового приложения.
4. Тестирование системы классов. Документируемая демонстрация работы тестового приложения.
5. Подготовка расчётно-пояснительной записки.
6. Демонстрация работы программы преподавателю и защита курсовой работы.

Курсовая работа имеет следующую структуру:

1. Введение:

обоснование темы, изложение цели, формулирование объекта и предмета исследования, определение задач работы, общая характеристика источников и методов изучения.
2. Основная часть:

содержит анализ предметной области, этапы проектирования и непосредственно описание разработки программного решения. Анализ предметной области обязательно включает: краткое описание предметной области (чему посвящена), какие в ней есть термины и понятия, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов и закономерности. Завершается раздел анализом существующих подобных программных продуктов. Разработка программного решения обязательно содержит: техническое задание, постановку задачи, описание объектов разработки.

3. Заключение содержит общие выводы, практические рекомендации, перспективы дальнейшей разработки программного продукта.

4. Список использованной литературы.

Список включает названия только тех статей, книг и других источников, на которые есть ссылки в работе. Список должен содержать не менее 15 источников.

5. Приложение – это необязательная часть работы, которая может содержать схемы, таблицы, графики, код программного продукта, и т.п.

3. Требования к оформлению курсовой работы

Рекомендуемый объем курсовой работы 25-40 страниц. Текст работы излагается только печатным способом на стандартных листах белой бумаги формата А4 по ГОСТ 9327 (210x297) с одной стороны, без рамки. При оформлении работы рекомендуется использовать шрифт Times New Roman, размер шрифта 14, межстрочный интервал – 1,5. Шрифт должен быть четким. Текст следует печатать, соблюдая следующие размеры полей: левое - 30 мм; правое - 15 мм; верхнее - 20 мм; нижнее - 20 мм.

Текст работы необходимо делить на абзацы, то есть части, начинающиеся с новой строки, это позволяет лучше воспринимать смысл изложенного материала. Каждый абзац должен выражать самостоятельную мысль и быть продолжением предыдущей мысли и абзаца. Формат абзаца текста должен быть выровнен «по ширине» положения на странице с отступами слева и права – 0 см. Абзацный отступ первой строки каждого абзаца должен быть равен 1,25 см.

Не допускается:

- при переходе на новую страницу отрывать одну строку текста или слово от предыдущего абзаца;
- начинать одну строку нового абзаца на заканчивающейся странице;
- начинать в конце страницы слово с переносом.

Нумерация страниц работы должна быть сквозной, включая список использованной литературы и приложения. Нумерация начинается со страницы 3 (введение), первой страницей является титульный лист, второй – содержание и так далее, последней – первая страница приложения. Номер страницы проставляется арабскими цифрами в центре верхней части листа без точки. На титульном листе и странице «Содержание» номер страницы не ставится. Если в работе содержатся иллюстрации (рисунки, таблицы и т. д.), которые располагают на отдельных страницах, их необходимо включать в общую нумерацию. Если рисунок или таблица расположены на двух и более страницах, то каждая страница нумеруется отдельно.

Текст работы разбивается на разделы (главы), подразделы (параграфы) и пункты, которые должны иметь порядковые номера. Заголовки разделов (глав), соответствующие теме и плану работы, указанному в содержании, печатаются жирным шрифтом (размер шрифта 16); выравняются по центру текста без абзацного отступа, без подчеркивания, без точки в конце, межстрочный интервал двойной. Заголовки подразделов и пунктов печатаются жирным шрифтом (размер шрифта 16), выравняются по центру текста без абзацного отступа, без подчеркивания, без точки в конце, межстрочный интервал двойной. Если заголовок включает несколько предложений, их разделяют точками. Переносы слов в заголовках не допускаются. Длину строки заголовка не рекомендуется делать более 2/3 общей длины строки. Кроме этого, не заканчивают строку заголовка предлогом, союзом – их переносят на следующую строку.

Например:

Глава 2. Проектирование и разработка программного решения

2.1. Проектирование и разработка классов

Текст параграфа

Не разрешается помещать заголовки отдельно от последующего текста.

На странице, где приводится заголовок, должно быть не менее двух строк последующего текста. Нумерация разделов, подразделов, пунктов. Все разделы, подразделы, пункты нумеруются арабскими цифрами в начале заголовка. Разделы (главы) работы должны иметь порядковую нумерацию в пределах основной части работы и обозначаться арабскими цифрами с точкой, на- пример: 1., 2., 3., и т.д. Подразделы (параграфы) – часть раздела, нумеруются арабскими цифрами в пределах каждого раздела (главы) двумя цифрами, разделенными точкой. В конце номера подраздела также ставится

точка. Первая цифра обозначает номер раздела, вторая – номер подраздела. Например:

2.2. (второй подраздел, второй главы). Пункты – часть раздела или подраздела, обозначенные номером. Если текст не имеет подразделов, то нумерация пунктов должна быть в пределах раздела. Если текст имеет подразделы, то нумерация будет состоять из трех цифр. Первая цифра обозначает номер раздела, вторая – номер подраздела, третья - номер пункта.

Например:

2.4.3. (третий пункт четвертого подраздела второй главы).

Глава 1. Теоретические аспекты исследования конкурентной политики организации

2.2. Сущность, основные понятия и факторы конкурентной политики организации

Текст параграфа

Подпункт – часть пункта, имеющая порядковую нумерацию в пределах каждого пункта. Каждый элемент подпункта начинается с большой буквы и завершается точкой. Перечисления – структурный элемент текста работы, содержащий перечисления требований, указаний, положений. Перечисления, при необходимости, могут быть приведены внутри пунктов и подпунктов. Перечисления выделяются цифрами, буквами, абзацными отступлениями с тем, чтобы более сильно подчеркнуть, и тем самым обеспечить понимание сложного целого или упростить ссылки на тот или иной элемент. Элементы внутри абзацного перечисления выделяются арабскими цифрами или строчными буквами с закрывающимися скобками.

В курсовой работе повреждения листов работы, пометки, описки, ошибки и следы не полностью удаленного текста не допускаются. Иллюстрации – графики, чертежи, схемы, диаграммы, фотографии - именуются рисунками и обозначаются в тексте сокращением «Рис.». Каждая иллюстрация должна пояснять научный материал, то есть давать возможность наглядного восприятия рассматриваемых явлений, процессов и подытоживать

цифровые данные. В работе следует помещать лишь такие иллюстрации, которые дополняют, раскрывают содержание текста. Иллюстрация должна иметь наименование, а при необходимости – поясняющие данные в подрисуночном тексте. Наименование иллюстрации помещают под ней по центру, шрифт курсивом, а поясняющие данные – непосредственно под иллюстрацией. Иллюстрации в работе допускаются только черно- белые (в отличие от презентаций). Ссылки на иллюстрации в тексте работы делают по типу: На рис. 1 представлена диаграмма состояний автомата Мили.

Например:

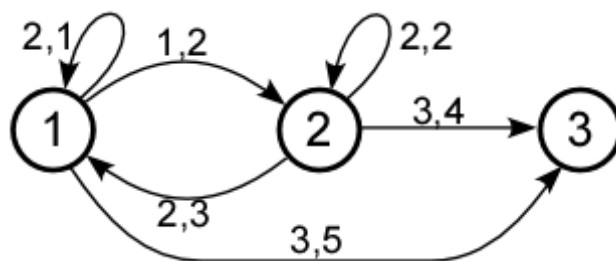


Рис 1. Диаграмма состояний автомата Мили.

Схемы являются графическим документами, на которых составные части объекта, их взаиморасположение и связи между ними изображены условно. Схемы выполняют без соблюдения масштаба, при этом действительное пространственное расположение составных частей можно не учитывать. Оформляются схемы по образцу оформления рисунков. Оформление таблиц При составлении и оформлении таблиц рекомендуется выполнять ряд требований (правил), основными из которых являются следующие: 1) слово «Таблица», порядковый номер и заголовок таблицы располагаются по центру, без абзацного отступа, без точки в конце непосредственно над таблицей; 2) заголовок печатается с прописной буквы курсивом, в единственном числе; 3) подчеркивать заголовок не допускается.

Таблица 1. Заголовок таблицы

Заголовок столбца	Подзаголовок столбцов		
	Заголовок	Заголовок	Заголовок

Таблицы в тексте работы имеют сквозную нумерацию. Номер таблицы печатается в правом верхнем углу над заголовком таблицы (например: Таблица 1). Знак «№» перед цифрой не ставится. Если в работе содержится одна таблица, ее не нумеруют. Заголовки строк и граф должны начинаться с прописной буквы; подзаголовки – со строчной, если они составляют одно предложение с заголовком, и с прописной, если они самостоятельные. Деление шапки таблицы по диагонали не допускается.

Рекомендуемый шрифт текста в таблице – 12, интервал 1,0 или 1,5, без каких-либо выделений. Если таблица по своим размерам большая и занимает не одну страницу или имеет большое количество столбцов, то допускается изменить в таблице шрифт на «11», интервал – «одинарный».

В таблице должны быть указаны единицы измерения всех показателей. Если размерность всех показателей, включенных в таблицу, одинаковая, то она указывается в круглых скобках сразу под названием таблицы. Если же показатели измеряются разными единицами, то в таблице после – первой графы «Наименование показателя» выделяется вторая графа «Единица измерения». Числовые данные в таблице приводятся с одинаковой степенью точности и выполнением правил округления. Для облегчения ссылок в тексте и при переносе таблицы на другую страницу делают нумерацию граф. Графу «№ п/п» в таблицу включать не допускается. При необходимости нумерации показателей, порядковые номера следует указывать в заголовках строк таблицы, непосредственно перед их наименованием. Если повторяющийся в графе таблицы текст состоит из одного слова, его допускается заменять кавычками; если из двух или более слов, то при первом повторении его заменяют словами «То же», а далее – кавычками. Нельзя ставить кавычки вместо повторяющихся цифр, марок, знаков математических и химических символов. Если цифровые или иные данные в какой-либо строке таблицы не приводятся, то в ней необходимо ставить прочерк.

Ссылки на таблицу в тексте работы делают по типу: В табл. 1 приведены диаграмма состояний автомата Мили. Не рекомендуется располагать две или несколько таблиц одну за другой, их надо разделять текстом. Таблицу следует размещать в тексте таким образом, чтобы ее можно было читать без поворота работы или с поворотом по часовой стрелке. Если таблица с большим количеством строк, допускается переносить на другую страницу. На странице, где приводится заголовок, должна помещаться шапка таблицы и не менее двух ее строк. При переносе части таблицы на следующую страницу над ней вверху справа помещают слова «Продолжение таблицы...» с указанием номера. Заголовок таблицы помещают только над первой ее частью.

Например:

На первой странице

Таблица 2. Объёмы продаж предприятия

Продукция	Количество, шт	Регион 1, %	Регион 2, %
1	2	3	4

На второй странице

Продолжение таблицы 2

Продукция	Количество, шт	Регион 1, %	Регион 2, %
1	2	3	4

В приложения 1, 2, 3 даны образцы оформления титульного листа, содержания, перечень примерных тем курсовых работ.

Курсовая работа подчиняется требованиям, предъявляемым к научному стилю. Как любой научный текст, курсовая работа должна удовлетворять

требованиям логического построения и максимальной объективности изложения. Синтаксическая структура должна быть стройной. Предложения должны быть развернутые, связи между элементами внутри предложения, между предложениями внутри абзацев и абзацами внутри глав выражены эксплицитно, что ведет к обилию и разнообразию союзов и союзных слов.

Важную роль в раскрытии логической структуры работы играет деление на абзацы. Каждый абзац желательно начинать с ключевого предложения, излагающего основную мысль. Для усиления логической связи между предложениями употребляются специальные устойчивые выражения и наречия, как: следовательно, таким образом, кроме того, наконец, итак; иными словами, точнее; по мнению..., по данным..., на ваш взгляд... и др.

Научный стиль характеризуется употреблением множественного числа личного местоимения первого лица вместо единственного: целью нашего исследования..., нами сформулировано следующее определение..., мы считаем...

Общая характеристика лексического состава научного текста имеет следующие черты: слова употребляются либо в основных прямых, либо в терминологических значениях, но не в экспрессивно-образных. Лексика литературная нейтральная или книжная. Научный текст характеризуется преобладанием именного стиля, что дает возможность большего обобщения. По этой же причине предпочтение отдается конструкциям в страдательном залоге и неличным формам глагола. Экспрессивность в научном тексте не исключается, но она специфична. Преобладает количественная экспрессивность: гораздо более ограничены..., наиболее важный..., чрезвычайно эффективный..., и т.п. Образная экспрессивность практически исключена.

При выполнении работы студент должен научиться составлять программы высокого качества, являющиеся легко модифицируемыми и простыми в обращении с использованием современных парадигм программирования. К этим парадигмам относятся: объектно-ориентированное

программирование, процедурное программирование, модульное программирование, событийно-ориентированное программирование и визуальное программирование. Также студент должен продемонстрировать, закрепить и улучшить свои практические навыки проектирования и разработки программного обеспечения с использованием принципов объектно-ориентированного анализа и программирования, а также современных информационных технологий и инструментов :

1) формируются навыки составления технического задания на проект (ТЗ);

2) формируется умение проектировать дизайн графических пользовательских интерфейсов (подготовка дизайн-макетов);

3) закрепляются навыки использования интегрированной среды разработки (Microsoft Visual Studio);

4) закрепляются и углубляются знания основ программирования на одном из современных высокоуровневых языков программирования (Microsoft Visual C#, C++, Java, PHP5) с использованием современных парадигм;

5) закрепляются и углубляются знания главных принципов объектноориентированного программирования;

6) формируется умение проектировать архитектуру приложения с использованием классов (подготовка UML-диаграмм классов);

7) формируется умение проектировать классы для хранения и представления данных и моделей (подготовка UML-диаграммвариантов-использования);

8) закрепляются и углубляются навыки разработки пользовательского интерфейса согласно дизайн-макетам;

9) закрепляются и углубляются навыки программирования алгоритмов обработки данных, взаимодействия с пользователем и др.;

10) формируются навыки составления отчета о проделанной работе и защиты результатов проекта.

Основой курсовой работы по дисциплине «Объектно-ориентированное программирование» является некоторая предметная область со своими терминами, понятиями, объектами, отношениями между этими объектами. Очевидно, что специалист в области информационных технологий далеко не всегда является специалистом в той сфере, для которой он выполняет разработку информационной системы. Поэтому залогом успешного решения поставленной перед ним задачи является подробный и качественный анализ всех аспектов той пользовательской среды, в которой будет функционировать создаваемое программное приложение или информационная система. В ходе анализа предметной области необходимо на основе знакомства с литературными источниками и общения с заказчиком выявить:

1. Чему посвящена предметная область, какие в ней есть термины и понятия, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности. Например, если речь идёт о графических примитивах в трёхмерном пространстве, то следует выявить список возможных примитивов (точка, линия, прямоугольник, параллелепипед, шар и т.п.), способы их описания (так, для точки достаточно указать её координаты, а для шара необходимо знать координаты центра и радиус), возможные способы преобразования (перемещение, масштабирование, поворот и т.п.).

2. Что входит в словарь предметной области, отдельно выделив список существительных и список глаголов, которые могут быть связаны с существительными. Для графических примитивов существительными могут быть: «точка», «координата», «шар», «угол», «цвет», «длина», «ширина» и др. А в качестве глаголов можно указать: «нарисовать», «повернуть», «масштабировать», «переместить».

3. Каковы функциональные требования к разрабатываемой информационной системе. Основой их служат потребности заказчика, однако разработчик должен оценить возможность реализации требований, исходя из технических возможностей и имеющихся ресурсов. Результат анализа должен

быть формализован. В реальной ситуации обычно оформляется протокол обсуждения, заключается договор, формулируется техническое задание. Все документы заверяют полномочные представители заказчика и разработчика. Во избежание конфликтных ситуаций следует задокументировать все решения, принятые по спорным моментам.

В рамках курсовой работы в роли заказчика выступает преподаватель, выдавший задание (либо представитель работодателя, если задание было сформулировано им). Студент проводит анализ предметной области, основываясь на своих собственных знаниях, литературных источниках и в ходе общения с преподавателем. Результат должен быть оформлен в виде небольшого реферативного описания предметной области. Из этого описания должен логически следовать словарь предметной области, состоящий из списка существительных и глаголов. Именно он послужит основой следующего этапа работы. Следующий этап посвящается описанию проектирования и разработки программного решения, в соответствии с темой курсовой работы (проекта). Обязательным элементом является “Проектирование и разработка классов”. Входной информацией являются определенные на этапе анализа задач объекты, общая характеристика программы. Проектирование системы классов начинается с обработки словаря предметной области. Эта обработка состоит в выявлении того, какие слова соответствуют объектам, классам, свойствам и операциям.

Список существительных служит основой для выделения классов и их свойств, а список глаголов – для определения операций. В качестве примера для графических примитивов примера можно указать следующее соответствие:

- классы: точка, шар;
- свойства: координата, угол, цвет, длина, ширина;
- методы: нарисовать, повернуть, масштабировать, переместить.

Следующий шаг состоит в том, чтобы определить, какой из классов какие свойства и функции содержит.

Следует обратить внимание на то, что эти наборы у разных классов могут «пересекаться». Например, и для класса «точка», и для класса «шар» справедливо наличие операций «нарисовать», «масштабировать», «переместить».

В то же время, метод «повернуть» не имеет смысла по отношению к объектам данных классов, зато может присутствовать у класса «параллелепипед». Ещё одним вопросом, требующим решения на данном шаге, является выявление отношений между классами. Речь идёт об отношениях наследования и включения. Следует обратить внимание, что понятие «наследование» чаще всего возникает тогда, когда разные классы обладают частично схожими наборами свойств и методов. При составлении словаря предметной области далеко не всегда в список могут попасть понятия, которым можно сопоставить базовые классы в иерархии наследования. Поэтому следует внимательно проанализировать список классов, свойств, методов, их соответствие, и, возможно, выделить ряд новых классов, связанных с имеющимися отношениями наследования и включения.

На данном этапе можно уже учитывать не только законы предметной области, но и такие принципы объектно-ориентированного подхода как абстракция, инкапсуляция, полиморфизм. В рассматриваемом выше примере можно выделить абстрактный класс «фигура» со свойствами «абсцисса», «ордината», «аппликата», «цвет» и операцией «нарисовать». Классы «точка» и «шар» будут являться наследниками класса «фигура», а операция «нарисовать» может быть являться виртуальной, что даёт нам полиморфический кластер, включающий три класса. Завершается этап тем, что разрабатываются полные спецификации базовых классов системы. При этом характеристики объектов вписываются в поля классов, а на основании операций разрабатываются методы и определяются способы доступа (public, private, protected).

Следует иметь в виду, что все имена классов, полей и методов должны быть осмысленными, начинаться с заглавной буквы. На этапе проектирования

для именованя классов, полей и методов рекомендуется использовать русские слова, однако, на этапе разработки предпочтительным является английский язык. Большим плюсом к проекту будет являться использование известных паттернов (шаблонов) проектирования, таких как: одиночка (singleton), прототип (prototype), строитель (builder), абстрактная фабрика (abstract factory), адаптер (adapter), декоратор (decorator), заместитель (proxy), мост (bridge), компоновщик (composite), фасад (facade), итератор (iterator), команда (command), наблюдатель (observer), стратегия (strategy), посетитель (visitor), посредник (mediator), состояние (state). Завершается этот этап подготовкой и представлением в этом разделе UML-диаграмм классов.

После проектирования и разработки классов в тексте необходимо представить проектирование пользовательского интерфейса информационной системы. На этом этапе разрабатываются:

- 1) структурная схема пользовательского интерфейса программы;
- 2) детали управления системой (пользовательские истории);
- 3) дизайн-макеты основных окон графического интерфейса.

Структурная схема позволяет выверить все детали проекта, определить взаимоотношения между отдельными частями программы, а также определяет содержание программных сообщений. На основании этой схемы в дальнейшем можно построить схему движения информационных потоков, диаграмму взаимодействия классов и т.д. Начало данного раздела может быть, например, следующим: «В соответствии с проведенным выше анализом задачи разрабатываемая программа должна содержать: последовательное задание исходных данных для эксперимента, которые определяют условия для решения задачи; выбор вида решения и вида результатов (в соответствии с ранее определенными функциями).

В результате может быть сформирована структурная схема функционирования интерфейса которая представлена на рис.1. ... После проектирования пользовательского интерфейса описывается разработка модульных элементов программы, приводится иерархическая структура

приложения и схема связности модулей. Также в этом разделе нужно привести описание наиболее важных (сложных) алгоритмов, разработанных для данной информационной системы. Алгоритмы можно описать с использованием классического языка блок-схем или (более предпочтительный вариант) – с использованием нотации UML-диаграмм (activity, sequence). Завершается раздел кратким руководством пользователя по эксплуатации разработанной системы.

Исходный код разработанных программ рекомендуется включать в приложения к курсовой работе, а также в электронном виде приложить к работе на цифровом носителе информации (компакт-диск).

4. Оформление списка использованной литературы

Список использованной литературы составляют непосредственно по произведению печати или выписывают из каталогов и библиографических указателей полностью, без пропусков каких-либо элементов, сокращений заглавий и т.п. После общего количества страниц источника печатается строчная буква «с». Например:

Математические методы педагогических измерений : моногр. / Г.А. Жаркова., А.В. Жарков. – Ульяновск : УлГУ, 2018. – 154 с. ISBN 978-5-88866-689-0

Если ссылка дается на статью из журнала или сборника – количественная характеристика представляет собой номера страниц, на которых опубликована эта статья. Перед номерами страниц ставится прописная буква «С» с точкой.

Например:

Жаркова, Г.А., Митина И.Д., Чуканов И.А. Информационное обеспечение компьютерных игровых моделей как средство оптимизации образовательного процесса вуза / Г.А. Жаркова, И.Д. Митина, И.А. Чуканов // Экономические и гуманитарные исследования регионов. 2018. № 6. – С.66-72.

В библиографическом списке должно быть указано полное название акта, дата его принятия, номер, а также официальный источник опубликования.

Например:

Федеральный закон от 26 февраля 1995 г. № 208-ФЗ «Об акционерных обществах» //Собрание законодательства Российской Федерации, 1996. - №1, ст.1.

Оформляют список в алфавитном порядке. Количество источников – 15-20.

5. Оценивание курсовой работы

Для защиты курсовой работы на кафедре необходимо подготовить презентацию, которая сопровождается докладом студента. Предварительно работа сдается преподавателю для проверки и получения допуска к защите (не менее чем за 14 дней до назначенного срока защиты). Результаты курсовой работы определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно»:

- оценку «отлично» заслуживает работа, в которой дано всестороннее и глубокое освещение избранной темы в тесной взаимосвязи с практикой, а её автор показал умение работать с различными видами источников, систематизировать, классифицировать, обобщать материал, формулируя выводы, соответствующие поставленным целям;

- оценкой «хорошо» оценивается работа, отвечающая основным, предъявляемым к ней, требованиям. Студент обнаруживает глубокие знания по предмету и владеет навыками научного исследования, но при этом имеются незначительные замечания по содержанию работы, по процедуре защиты (студент не может дать аргументировано ответы на вопросы);

- курсовая работа оценивается на «удовлетворительно», если в ней, в основном, соблюдены общие требования, но неполно раскрыты разделы плана, работа носит реферативный характер, отсутствуют аргументированные

выводы. Автор курсовой работы посредственно владеет материалом, поверхностно отвечает на вопросы, в процессе защиты курсовой работы;

- «неудовлетворительно» оценивается курсовая работа, если установлено, что работа выполнена несамостоятельно, имеются принципиальные замечания по многим параметрам, содержание не соответствует теме, допущены грубые теоретические ошибки.

Защищенная или зачтенная курсовая работа хранится на кафедре в течение года.



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет Математики, Информационных и Авиационных технологий

Кафедра Информационных технологий

**КУРСОВАЯ РАБОТА
по дисциплине «название дисциплины»**

(название темы)

020403 - «Математическое обеспечение и администрирование информационных систем»
(магистратура),
(профиль/специализация) Технология программирования

(наименование и номер специальности(направления))

Проект выполнил студент

группа

подпись, дата

Ф.И.О.

Научный руководитель

должность

подпись, дата

Ф.И.О.

оценка

У Л Ь Я Н О В С К
2020 г.

Содержание

Введение	3
Глава 1. Анализ предметной области	4
1.1. Описание предметной области	4
1.2. Выделение словаря предметной области	11
1.3. Анализ существующих программных решений	14
1.4. Техническое задание	17
Выводы по первой главе	19
Глава 2. Проектирование и разработка программного решения	20
2.1. Проектирование и разработка классов	20
2.2. Проектирование пользовательского интерфейса	24
2.3. Описание разработанных алгоритмов и модулей системы	27
2.4. Краткое руководство пользователя по эксплуатации системы	30
Выводы по второй главе	34
Заключение	35
Библиографический список	36
Приложения	37
Приложение 1. UML-диаграммы	38
Приложение 2. Исходный код модуля	39

Примерные темы курсовых работ

1. Объектная реализация контейнера на основе комбинированной структуры «Двунаправленный список массивов-стеков»
2. Объектная реализация контейнера на основе комбинированной структуры "Упорядоченный массив динамических очередей"
3. Разработка объектной программы для задачи управления новостным Internet-порталом
4. Разработка объектной программы для задачи учета компьютерной техники в организации
5. Основы объектно-ориентированного визуального программирования
6. Проектирование библиотеки классов для реализации оконного интерфейса на платформе MS Windows
7. Методология объектно-ориентированного программирования. Свойства экторной модели параллельных вычислений Хьюита
8. Разработка механизма сборки мусора объектов на языке программирования C++
9. Исследование графических возможностей Windows. Создание приложения (на любом языке программирования), которое использует возможности интерфейса Aero.
10. Программирование гаджетов (Gadgets) в Windows. Создание гаджета для отображения картинки с произвольным текстом поверх нее.
11. Использование возможностей видеокарты для выполнения вычислений (CUDA от nVidia, ATI Stream и OpenCL). Написание программы шифрации и дешифрации с использованием возможностей видеокарты.
12. Исследование возможностей стандарта web-разработки HTML5 и CSS3 (каскадные таблицы стилей 3-й версии). Создание web-странички отображающей

графики и диаграммы средствами HTML5 (без применения JavaScript или с его минимальным использованием).

- 13.Использование хранимых процедур CLR в SQL Server 2008 R2. Исследование возможностей и сравнение с конкурирующими технологиями (обычные хранимые процедуры в TransactSQL, PL/SQL в Oracle и в PostgreSQL,...). Создание программы, у которой вся бизнес-логика реализована средствами хранимых процедур CLR (т.е. в программе имеются только вызовы этих процедур).
- 14.Библиотека классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований.
- 15.Библиотека классов, реализующих графические примитивы на плоскости с реализацией операций над множествами.
- 16.Библиотека классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований.
- 17.Классы для описания оконного графического интерфейса, аналогичного интерфейсу MS Windows. Должны быть реализованы обработчики событий с использованием виртуальных функций.
- 18.Библиотека классов, реализующих комплексную арифметику.
- 19.Библиотека классов, реализующих векторы в n-мерном пространстве.
- 20.Реализация строк и операций над ними, включая работу с регулярными выражениями.
- 21.«Обобщённый массив» (позволяющий хранить данные произвольных типов).
- 22.Ассоциативный массив (хэш-массив) (с возможностью хранения данных произвольных типов).
- 23.Реализация различных типов графов и операций над ними.

24. Система классов для обеспечения работы с абонентами телефонной компании.
25. Система классов для обеспечения работы деканата.
26. Система классов, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...).
27. Система классов, описывающих различные транспортные средства.
28. Моделирование замкнутой биологической системы (корм, травоядное, хищник).
29. Моделирование муравейника (несколько типов муравьёв, источники питания, внешние раздражители, ...).
30. Моделирование дорожного движения на заданной карте дорог.
31. Моделирование компьютерной сети (стационарной).
32. Моделирование компьютерной сети с изменяющейся топологией.
33. Моделирование планетарной системы.
34. Система классов для реализации матричной арифметики.
35. Библиотека классов для реализации игрового приложения.