

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра автоматизации обработки информации (АОИ)

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Методические указания к выполнению курсовой работы
для студентов направления
«Программная инженерия»
(уровень бакалавриата)

2018

Морозова Юлия Викторовна

Объектно-ориентированное программирование: Методические указания к выполнению курсовой работы для студентов направления «Программная инженерия» (уровень бакалавриата) / Ю.В. Морозова. – Томск, 2018. – 15 с.

© Томский государственный университет систем управления и радиоэлектроники, 2018
© Морозова Ю.В., 2018

Оглавление

1 Введение	4
2 Методические указания к выполнению курсовой работы	5
2.1 Общие требования к выполнению курсовой работы	5
2.2 Тематика курсовой работы	5
2.3 Порядок выполнения работы	6
2.4 Требования к содержанию и оформлению отчета.....	10
3 Рекомендуемые источники	12
Приложение А	14

1 Введение

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.

Курсовая работа по дисциплине «Объектно-ориентированное программирование» имеет целью получение навыков самостоятельной разработки программного продукта в соответствии с принципами объектно-ориентированного программирования.

В результате изучения дисциплины **студент должен:**

- **знать:** методы обработки и способы реализации основных структур данных в объектно-ориентированных программных средах.
- **уметь:** разрабатывать объектно-ориентированные программы в современных инструментальных средах.
- **владеть:** практическими приемами объектно-ориентированного программирования.

Курсовая работа позволяет сформировать способности будущего специалиста к самостоятельному решению практических задач и инженерных проблем с использованием теоретических положений, а также знаний и умений, полученных в ходе обучения объектно-ориентированному программированию.

Данные методические указания предназначены для выполнения курсовой работы по дисциплине «Объектно-ориентированное программирование» подготовки бакалавров направления «Программная инженерия».

2 Методические указания к выполнению курсовой работы

2.1 Общие требования к выполнению курсовой работы

Задачей курсовой работы является разработка программного продукта для заданной предметной области, включающей в себя:

- объектную декомпозицию предметной области;
- разработку иерархии классов и интерфейсов;
- программную реализацию;
- тестирование и отладку программы;
- подготовку и написание отчета;
- защиту работы.

Выполнение курсовой работы предполагает самостоятельное изучение дополнительных вопросов по объектно-ориентированным языкам программирования (в частности, C++ и C#, Java), средам проектирования Windows-приложений, а также получение практического опыта объектно-ориентированного анализа и программирования, оформления соответствующей документации на программную разработку.

Курсовая работа выполняется по типовым заданиям, студент может предложить собственный вариант задания для согласования с преподавателем.

К защите курсовой работы представляется своевременно выполненная студентом и отрецензированная преподавателем работа.

2.2 Тематика курсовой работы

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 объектно-ориентированных языков с графическим интерфейсом, но и протестирована.

Примерный перечень тем для курсовой работы

1. Генератор геометрических фракталов на основе стандартных средств, предоставляемых Java.

2. Визуализация генетического алгоритма.
3. Модель леса.
4. Система сбора данных для мониторинга погоды.
5. Система «антиплагиат».
6. Система мониторинга успеваемости студентов по группам.
7. Интерпретатор для разработки и компиляции программ на языке Java.
8. Астрономическая модель солнечной системы.
9. Игра «Умный муравей».
10. Игра «Глупый муравей».

2.3 Порядок выполнения работы

Курсовая работа представляет собой результат выполнения следующих взаимосвязанных этапов:

1. Этап постановки и формулирования задачи.
2. Этап проектирования.
3. Этап кодирования.
4. Этап отладки и тестирования.
5. Документирование созданного программного продукта и оформление отчета по курсовой работе.
6. Защита.

Работа выполняется самостоятельно в свободное время и сдается в строго оговоренные сроки.

В рамках *первого этапа* следует:

- 1) произвести изучение предметной области и осуществить сбор материала в проблемно-ориентированном контексте;
- 2) определить назначение программы, выработать требования к ней и представить их, если это возможно, в формализованном виде;
- 3) сформулировать требования к представлению исходных данных и выходных результатов;
- 4) определить структуры входных и выходных данных;
- 5) сформулировать ограничения и допущения на исходные и выходные данные.

В рамках *второго этапа* следует:

- 1) сформировать объектно-ориентированную модель задачи;
- 2) осуществить выбор метод реализации решения задачи;
- 3) разработать алгоритм реализации задачи.

При создании любого программного продукта всегда присутствует этап проектирования задачи. При написании реальных программ, одна из первых возникающих при этом проблем – определить, должна ли быть программа объектно-ориентированной или нет. ООП, примененное к задаче, в которой в нем нет необходимости, только увеличит объем программы и сложность ее написания.

Смешивать два подхода в одном проекте не рекомендуется, поскольку может оказаться, что готовый программный продукт обладает недостатками и структурного, и объектно-ориентированного принципов построения. Но следует также иметь в виду, что даже небольшие программы могут с успехом пользоваться уже существующими библиотеками классов.

При создании класса, то есть нового типа данных, следует хорошо продумать его интерфейс – средства работы с классом, доступные использующим его программистам. Интерфейс хорошо спроектированного класса должен быть интуитивно ясен, непротиворечив и обзорим. Как правило, он должен включать в себя только методы и свойства. Поля данных должны быть скрытыми. Это даст возможность впоследствии изменить реализацию класса без изменения в его интерфейсе, а также регулировать доступ к полям класса с помощью предоставляемых пользователю методов. Не нужно расширять интерфейс класса без необходимости, «на всякий случай», поскольку увеличение количества методов будет вести к трудности понимания класса другим программистом, который возможно будет использовать его в своей работе. В идеале интерфейс должен быть полным, т.е. предоставлять возможность выполнить любые разумные действия с классом, и минимальным – без дублирования и пересечения возможностей методов.

В рамках *третьего этапа* следует:

- 1) произвести уточнение структуры входных и выходных данных и определение формата их представления в объектно-ориентированной программе;
- 2) произвести программирование решения задачи;
- 3) произвести комментирование текста программы и составления предварительного описания программы.

После завершения проектирования библиотеки классов можно приступить к её программной реализации. Однако этот процесс невозможен без предварительного решения ряда технических вопросов: на каком языке следует писать программу, какая среда разработки должна использоваться, каких правил именования идентификаторов следует придержи-

ваться, как организовать проектную работу, как документировать процесс программирования, как отслеживать изменения в коде программы.

Рекомендованными в рамках дисциплины языками являются C++ и C#, Java.

Рекомендуемой средой разработки является система для языка Java – Eclipse. Тем не менее, студент имеет право остановиться на каком-либо другом объектно-ориентированном языке высокого уровня, позволяющем разрабатывать независимые оконные приложения. Выбор языка требует обязательного обоснования.

Обоснование строится на основе выполненного анализа предметной области, исходя из следующих определяющих факторов:

- функциональные требования к системе;
- наличие в языке возможностей для реализации функциональных требований;
- трудоёмкость разработки.

Обоснование должно быть оформлено в виде связного текста и содержательно являться сравнительной оценкой альтернативных вариантов выбора по указанным критериям. То есть в случае выбора языка программирования должна быть выполнена оценка каждой из альтернатив и, как следствие, указан сделанный выбор.

Далее необходимо выполнить генерацию иерархии классов на выбранном языке программирования с получением основных классов и структур данных, сформировать архитектуру программного модуля или модулей, определить алгоритмы методов, разработать при необходимости интерфейс программного продукта.

Грамотно спроектированная диаграмма классов позволяет очень легко написать программный код, содержащий общее описание классов (иерархия классов, свойства, прототипы методов). Однако полноценно использовать классы и работать с объектами этих классов можно только в том случае, когда полностью даны определения всем методам. Поэтому в ходе программной реализации системы классов основной решаемой задачей является алгоритмизация и программирование методов классов.

В рамках *четвертого этапа* следует:

- 1) составить тесты для проверки правильности работы программы;
- 2) произвести обнаружение, локализацию и устранение ошибок в программе, выявленных с помощью тестов;
- 3) скорректировать код программы и ее описание.

Тестирование программного обеспечения должно показать работоспособность программы, а также выявить ошибки и дефекты. Следует различать процессы тестирования и отладки программного кода.

Отладка выполняется программистом с помощью встроенных средств среды разработки и исходя из опыта написания программного кода. В основном она сводится к выявлению синтаксических и семантических ошибок в тексте программы.

Тестирование – это процесс, требующий планирования и выполнения ряда предварительных процедур, основной из которых является составление набора тестовых сценариев, образующих тест-сьют. Тестовые сценария в большинстве случаев основаны на функциональных требованиях к системе и могут затрагивать различные уровни разработки (модульное тестирование, интеграционное тестирование, системное тестирование).

В ходе курсовой работы необходимо выполнить упрощённый вариант модульного тестирования, сводящийся к тестированию всех методов разработанной библиотеки классов.

Тестирование библиотеки классов выполняется в соответствии с разработанной методикой. Если все предыдущие этапы работы завершены успешно, то данная процедура в основном сводится к выполнению тестов и фиксации результатов их выполнения. Документирование может быть выполнено в упрощённом режиме. Для каждого выполненного теста фиксируются:

1. Номер теста (по нумерации тестов в методике тестирования).
2. Входные данные.
3. Ожидаемый результат выполнения теста.
4. Результат выполнения теста.
5. Вывод: результат выполнения теста соответствует/не соответствует ожидаемому.

Если в ходе тестирования выявляются несоответствия, следует выполнить анализ его причин и привести результаты анализа в тексте отчета (например, указать, чем вызвано несоответствие – семантическими или алгоритмическими ошибками, сформулировать рекомендации по исправлению несоответствия, и т.п.).

В рамках *пятого этапа* следует:

- оформить курсовую работу в виде отчета согласно требованиям ОС ТУСУР 0.1-2013.

В рамках *шестого этапа* следует:

- подготовить доклад и презентацию к защите курсовой работы.

Для успешной защиты курсовой работы необходимо выполнить весь объём курсовой работы, продемонстрировать функционирование разработанного приложения, представить отчет и ответить на вопросы.

Студенту предоставляется слово для доклада (время доклада – 5 мин).

Приветствуется научный стиль изложения, лаконизм и содержательность выводов по работе.

В докладе должны быть отражены следующие основные моменты:

- цель и задачи работы;
- обоснование выбора языка и среды программирования;
- изложение основных результатов работы;
- краткие выводы по тем результатам работы, которые определяют ее практическую значимость, степень и характер новизны элементов.

Доклад может сопровождаться презентацией.

После доклада студенту-автору работы задаются вопросы. Докладчику может быть задан любой вопрос по содержанию работы. Общая длительность защиты одной работы – не более 10 минут.

Примерная тематика вопросов:

1. Описать этапы проектирования системы классов.
2. Обосновать принятые решения о выделении классов, их свойств и методов, назначении конкретных режимов доступа к членам классов.
3. Прокомментировать обоснование выбора языка и среды разработки.
4. Пояснить конкретные фрагменты исходного кода программы (включая тестовое приложение).
5. Предложить альтернативные варианты системы классов.
6. Обосновать предложенную методику тестирования.

2.4 Требования к содержанию и оформлению отчета по курсовой работе

По результатам курсовой работы оформляется отчет. Оформление должно соответствовать требованиям стандарта ОС ТУСУР 0.1-2013. Пример оформления титульного листа приведен в приложении А.

Для данной курсовой работы рекомендовано следующее содержание отчета:

1. Титульный лист
2. Задание на курсовую работу
3. Содержание
4. Введение
5. Разработка библиотеки классов
 - 5.1. Диаграмма классов
 - 5.2. Выбор языка программирования
 - 5.3. Реализация классов
6. Тестирование
7. Заключение
8. Список использованных источников
9. Приложение А. Диаграмма классов
10. Приложение Б. Листинг программы

Объем работы должен составлять не менее 25 страниц основной части. Изложение должно быть последовательным, логичным, конкретным.

Первая страница – титульный лист, вторая – задание, далее – аннотация, оглавление и текст (номера первых двух страниц не указываются). Оглавление создается автоматически средствами текстового редактора.

Документ может содержать таблицы, рисунки и формулы. На каждую таблицу, рисунок, формулу должна быть ссылка в тексте.

В текст отчета могут быть включены небольшие фрагменты программного кода, обязательно с комментариями. Рекомендуемый шрифт для выполнения фрагмента кода – Courier New, размер 12пт.

На материалы, взятые из литературы и других источников, должны быть даны ссылки с указанием номера источника по списку использованной литературы.

Список составляется в порядке появления ссылок.

В приложениях размещаются листинг основных классов, схемы программы, скриншоты интерфейса. Приложения нумеруются русскими буквами в порядке появления ссылок на них в основном тексте документа.

Для защиты курсовой работы рекомендуется использовать презентацию в виде последовательности слайдов.

3 Рекомендуемые источники

1. Павловская Т. А. С/С++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. – СПб.: Питер, 2013. – 461 с.

2. Лаптев В. В. С++. Объектно-ориентированное программирование : учеб. пособие / В. В. Лаптев. – СПб. : Питер, 2008. – 464 с.

3. Катаев М. Ю. Объектно-ориентированное программирование : Учебное пособие / М. Ю. Катаев, А. Я. Суханов ; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра автоматизированных систем управления. – Томск : ТМЦДО, 2007. – 160 с.

4. Объектно-ориентированное программирование: Учебное пособие / Романенко В. В. – 2014. 475 с. – [Электронный ресурс]. – <https://edu.tusur.ru/training/publications/4872> (дата обращения: 4.05.2018).

5. Объектно-ориентированное программирование: Курс лекций / Ганджа Т. В., Панов С. А. – 2015. 110 с. [Электронный ресурс]. – <https://edu.tusur.ru/training/publications/5009> (дата обращения: 4.05.2018).

6. Лаптев В. В. С++. Объектно-ориентированное программирование. Задачи и упражнения : учебное пособие для вузов / В. В. Лаптев, А. В. Морозов, А. В. Бокова. – СПб. : Питер, 2007. – 288 с.

7. Ашарина И. В. Язык С++ и объектно-ориентированное программирование в С++. Лабораторный практикум. Учебное пособие / И.В. Ашарина. – М.: Горячая Линия – Телеком, 2015. – 232 с.

8. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, Р.А. Максимчук, М.У. Энгл и др. – М.: Вильямс, 2010. – 720 с.

9. Васильев А. Н. С#. Объектно-ориентированное программирование / А.Н. Васильев. – СПб.: Питер, 2012. – 320 с.

10. Лафоре Р. Объектно-ориентированное программирование в С++ / Р. Лафоре. – СПб.: Питер, 2015. – 928 с.

11. Лесневский А. С. Объектно-ориентированное программирование для начинающих / А.С. Лесневский. – М.: Бином. Лаборатория знаний, 2010. – 232 с.

12. Пол, А. Объектно-ориентированное программирование в С++ / А. Пол. – М.: Бином, 2001. – 464 с.

13. Рассел Дж. Объектно-ориентированное программирование / Дж. Рассел. – М.: Книга по требованию, 2012. – 74 с.

14. Хорев П. Б. Объектно-ориентированное программирование / П. Б. Хорев. – М.: Академия, 2012. – 448 с.

15. Шлеер С. Объектно-ориентированный анализ: моделирование мира в состояниях / С. Шлеер, С. Меллор. – М.: Диалектика, 1993.

16. Морозова Ю. В. Объектно-ориентированный анализ и программирование для студентов направлений «Программной инженерии» [Электронный ресурс]. – Режим доступа: <https://sdo.tusur.ru/course/view.php?id=40> (дата обращения: 24.04.2018).

17. Образовательный стандарт вуза ОС ТУСУР 01-2013. Работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления. [Электронный ресурс]. – Режим доступа: https://regulations.tusur.ru/storage/58109/obrazovatelnyy_standart_vuza_os_tusur_01-2013_raboty_studencheskie_po_napravleniyam_podgotovki_i_spetsialnostyam_tekhnicheskogo_profilya_obschie_trebovaniya_i_pravila_oformleniya.pdf?1490862364 (дата обращения: 4.05.2018).

Приложение А

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра автоматизации обработки информации (АОИ)

ТЕМА РАБОТЫ

Курсовая работа по дисциплине
«Объектно-ориентированное программирование»

Студент гр. ____
____ И. О. Фамилия
« ____ » _____ 201_ г.

Руководитель
доцент каф. АОИ,
канд. техн. наук
____ Ю. В. Морозова
« ____ » _____ 201_ г.

Томск 201_

Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра автоматизации обработки информации (АОИ)

УТВЕРЖДАЮ

Зав. каф. АОИ

_____ Ю.П. Ехлаков

«__» _____ 2018 г.

ЗАДАНИЕ

на курсовую работу по дисциплине
«Объектно-ориентированное программирование»

- 1) Тема работы: «_____».
- 2) Срок сдачи работы: «__» _____ 20__ г.
- 3) Цель работы: _____.
- 4) Формулировка задания: _____.
- 5) Содержание отчета: _____.

Задание выдал:

«__» _____ 20__

Принял к выполнению:

«__» _____ 20__