



**Филиал «Котельники» государственного
бюджетного образовательного учреждения
высшего образования Московской области
«Университет «Дубна»**

Артамонов Ю.Н.

Методические указания

**по выполнению курсовой работы по дисциплине
«Объектно-ориентированное программирование»**

**Для бакалавров по направлению
«Информатика и вычислительная техника»**

Москва – 2018

ОГЛАВЛЕНИЕ

1	Требования по оформлению курсовой работы	4
2	Варианты тем на курсовую работу	11
	Литература	14
A	Образец титульного листа	15

Введение

В ходе выполнения курсовой работы студенты должны практически освоить общий методологический подход, соответствующий объектно-ориентированной парадигме программирования. Создаваемая система классов описывает 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, но и протестирована. Для тестирования классов студент должен разработать оконное приложение, функциональные возможности которого должны демонстрировать различные варианты использования объектов созданных классов.

ГЛАВА 1

ТРЕБОВАНИЯ ПО ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

Пояснительная записка к курсовой работе выполняется в текстовом редакторе MS Word: формат А4, отступ слева 3 см., отступ справа 1 см., отступы сверху и снизу 2 см.; гарнитура шрифта Times New Roman; кегль 12 пунктов; интервал 1.5.

Структура пояснительной записки

Пояснительная записка курсовой работы должна включать следующие элементы:

1. Титульный лист
2. Содержание
3. Задание на курсовую работу
4. Анализ предметной области
5. Разработку библиотеки классов

6. Разработку тестового приложения
7. Выводы по работе
8. Список использованных источников
9. Приложения

Титульный лист оформляется в соответствии с приложением А.

Содержание оформляется, используя стандартные средства формирования оглавления в MS Word.

Анализ предметной области

Основой курсовой работы является некоторая предметная область со своими терминами, понятиями, объектами, отношениями между этими объектами. Залогом успешного решения поставленной 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.

2. Что входит в словарь предметной области, отдельно выделив список существительных и список глаголов, которые могут быть связаны с существительными.

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.

В рамках курсовой работы в роли заказчика выступает преподаватель, выдавший задание. Студент проводит анализ предметной области, основываясь на своих собственных знаниях, ли-

тературных источниках, а также общаясь с преподавателем. Результат должен быть оформлен в виде 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.

Разработка библиотеки классов

Проектирование системы классов начинается с обработки словаря предметной области. Эта обработка состоит в выявлении того, какие слова соответствуют объектам, классам, свойствам и методам. Список существительных служит основой для выделения классов и их свойств, а список глаголов – для определения методов. Например, для графических примитивов можно указать следующее соответствие: классы: точка, шар; свойства: координата, угол, цвет, длина, ширина; методы: нарисовать, повернуть, масштабировать, переместить.

Следующий шаг является, фактически, завершающим на этапе проектирования классов. Он состоит в том, чтобы определить, какой из классов какие свойства и методы содержит. Следует обратить внимание на то, что наборы свойств и методов у разных классов могут «пересекаться». Например, и для класса «точка», и для класса «шар» справедливо наличие методов «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 клас-

сов, 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 примере можно выделить абстрактный класс «фигура» со свойствами «абсцисса», «ордината», «аппликата», «цвет» и методом «нарисовать». Классы «точка» и «шар» будут являться наследниками класса «фигура», а метод «нарисовать» может являться виртуальным, что даёт нам полиморфический кластер, включающий три класса.

Результаты такого анализа должны быть оформлены в виде диаграммы классов. Предпочтительным является использование нотации языка UML.

В частности, следует придерживаться следующих правил:

- класс обозначается прямоугольником;
- прямоугольник делится на три части, в каждой из которых, соответственно, указываются: имя класса, список свойств, список методов;
- имена классов, свойств и методов могут быть записаны на русском языке, но в соответствии с нормами написания стандартных идентификаторов (одно слово, включающее буквы, цифры, символ подчёркивания и не начинающееся с цифры);
- имена классов записываются с заглавной буквы, имена свойств и методов – со строчной;
- перед именем свойства или метода ставится символ, указывающий на режим доступности: закрытый (-), защищённый (#), открытый (+);
- после имени метода ставятся круглые скобки, в которых могут быть перечислены параметры метода;

- наследование классов обозначается стрелкой с треугольным незакрашенным наконечником;
- стрелка при наследовании направляется от класса - наследника к родительскому классу.

Пример диаграммы классов приведён на рисунке 1.1.

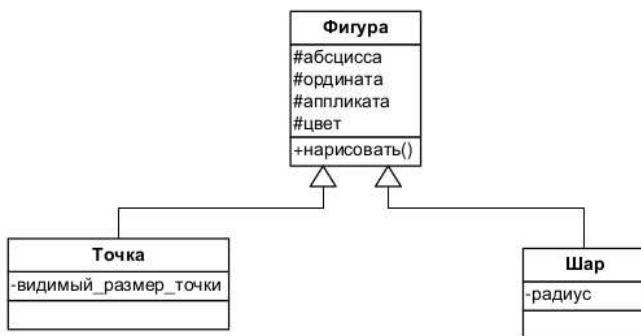


Рис. 1.1: Пример UML-диаграммы

Разработка тестового приложения

Тестирование – это процесс, требующий планирования и выполнения ряда предварительных процедур, основной из которых является составление набора тестовых примеров, образующих тест-план. Тестовые примеры 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.

Следует отметить те моменты и требования, которые присутствуют при разработке тестового приложения.

1. Тестовое приложение должно иметь оконный интерфейс, включающий меню и интерактивные интерфейсные средства, с помощью которых можно протестировать созданную систему классов.
2. Набор интерфейсных элементов (кнопки, списки, поля ввода, диалоговые окна, графические элементы и т.д.) должен быть определён на основе описанной методики тестирования.
3. В текст пояснительной записки следует обязательно включить структурную схему, описывающую компоненты приложения и связи между ними (с сопроводительным текстом) и экранные формы, наглядно демонстрирующие интерфейс программы. По собственной инициативе студент может включить в описание тестового приложения функциональные схемы, блок-схемы алгоритмов, таблицы, схемы классов, т.е. любые материалы, позволяющие лучше понять процесс разработки и функционирование тестового приложения.

Выводы по работе

При написании выводов по работе студент должен раскрыть следующий перечень вопросов:

- Описать этапы проектирования системы классов.
- Обосновать принятые решения о выделении классов, их свойств и методов, назначении конкретных режимов доступа к членам классов.
- Прокомментировать обоснование выбора языка и среды разработки.
- Предложить альтернативные варианты системы классов.
- Обосновать предложенную методику тестирования.

Литературные источники

При разработке программ, целесообразно использовать не менее 10 литературных источников. В пояснительной записке следует указывать ссылки на литературные источники. Литературные источники указываются в квадратных скобках. Например, [1]. Разрешается указывать сразу несколько источников: [1, 4, 7], [2-4, 6]. Соответственно в разделе «Список использованных источников» необходимо указать библиографические ссылки, которые следует оформлять по ГОСТ 7.1-2003 (по аналогии с библиографическими ссылками данного методического пособия).

Приложения

В приложения следует помещать программный код всего проекта Qt, скриншоты всех экранных форм.

Другие требования и рекомендации

К пояснительной записке должен быть приложен компакт-диск с исходным кодом приложения и исполняемым файлом, а также лист проверки курсовой работы в системе:

<https://www.antiplagiat.ru/>.

ГЛАВА 2

ВАРИАНТЫ ТЕМ НА КУРСОВУЮ РАБОТУ

Ниже представлены варианты тем на курсовую работу по вариантам. При реализации библиотеки классов следует проявить элемент творчества, не ограничиваясь лишь перечисленным функционалом.

1. Библиотека классов, реализующая движение различных графических примитивов на плоскости с отражением их от краев прямоугольной формы (предусмотреть примитивы разной формы, цвета, поведения, времени жизни, реакции на соударение.)
2. Библиотека классов, моделирующая муравейник: предусмотреть несколько типов муравьёв, источники питания, внешние раздражители, варианты размножения и гибели.
3. Библиотека классов, реализующая работу компьютерной сети, используя понятия коммутатор, роутер, повторитель,

линия связи, сервер, пользовательский компьютер, пакет. Смоделировать работу компьютерной сети при разных настройках ее параметров.

4. Библиотека классов, реализующая работу со строками. Ввести понятия символа, слова, строки, текста, ввести операции над ними: поиск, замена, статистические вычисления, реализовать работу с регулярными выражениями.
5. Библиотека классов «обобщённый массив» (позволяющий хранить данные произвольных типов). Предусмотреть проверку диапазона индексов, присвоение массивов, сравнение массивов, ввод, вывод массивов, знание массивами своих размеров.
6. Библиотека классов классов, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, клиент, директор и т.д.). Смоделировать взаимодействие объектов фирмы, учитывая факторы заработной платы, трудоемкости и т.д.
7. Библиотека классов, описывающая функционирование замкнутой биологической системы (корм, травоядное, хищник, со взаимосвязями: чем больше хищников, тем меньше травоядных, становится больше корма (травы), меньше хищников, что опять приводит к росту числа травоядных и т.д.).
8. Библиотека классов для реализации матричной арифметики. Предусмотреть операции транспонирования матрицы, нахождения обратной (для прямоугольной матрицы), нахождения ранга, а также операции сложения, умножения матриц, присвоения матриц и т.д.
9. Библиотека классов, моделирующая улей пчел: предусмотреть несколько типов пчел, сбор меда, внешние раздражители (наличие цветов, дождь, солнце и т.д.), варианты размножения и гибели.

10. Библиотека классов, реализующая работу с большими числами (произвольной длины): сложение, вычитание, умножение, деление, возведение в степень, нахождение остатка от деления, возведение в степень по модулю.

Распределение заданий по вариантам

Номер варианта:	1	2	3	4	5	6	7	8	9	10	11
Номер задания:	1	2	3	4	5	6	7	8	9	10	1

Номер варианта:	12	13	14	15	16	17	18	19	20
Номер задания:	2	3	4	5	6	7	8	9	10

Номер варианта:	21	22	23	24	25	26	27	28	29
Номер задания:	9	3	4	5	1	7	8	2	10

Допускается командная работа в рамках одного варианта курсовой работы, при этом каждый студент пишет свою пояснительную записку. Для совместной работы над кодом программного приложения целесообразно использовать сервисы github.com, gitlab.com на выбор команды.

Команда оценивает участие каждого в процентном отношении, о чем сообщает преподавателю.

ЛИТЕРАТУРА

- [1] *Лафоре Р.* Объектно-ориентированное программирование в C++. – СПб.: Питер, 2007.
- [2] *Павловская Т.А.* C++. Объектно-ориентированное программирование: практикум : учебное пособие. — Санкт-Петербург : Питер, 2006.
- [3] *Дубов И.Р.* Объектно-ориентированное программирование: учебное пособие. — Владимир: Владимирский государственный университет, 2003.
- [4] *Лантез В.В., Морозов А.В., Бокова А.В.* C++. Объектно-ориентированное программирование. Задачи и упражнения. – СПб.: Питер, 2007.

ПРИЛОЖЕНИЕ А _____

_____ ОБРАЗЕЦ ТИТУЛЬНОГО ЛИСТА

Филиал «Котельники» государственного бюджетного
образовательного учреждения высшего образования
Московской области «Университет «Дубна»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
на курсовую работу по дисциплине
«Объектно-ориентированное программирование»

ВАРИАНТ №1

Выполнил: _____

студент группы ИВТ-21 Иванов И.И.

Проверил: _____

доцент, к.т.н. Артамонов Ю.Н.

Котельники – 2018