

ФГБОУ ВО Ижевский государственный технический университет
им. М.Т. Калашникова

Методические указания к курсовой работе
по дисциплине «Объектно-ориентированное программирование»
Чернышев К.С.

Ижевск, 2020

1. Организация курсовой работы

1.1. Цели курсового проектирования

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. 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.

1.2. 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 реализована в виде программы на одном из Си-подобных объектно-ориентированных языков.

1.3. Задание на курсовое проектирование

В ходе курсовой работы студентам необходимо разработать приложение, спроектированную и реализованное для решения конкретной задачи из некоторой предметной области. При этом подлежат разработке следующие вопросы:

- а) анализ задания;
- б) разработка библиотеки классов;
- в) оформление расчётно-пояснительной записки по результатам выполнения работы.

Открытый список вариантов задания на курсовое проектирование приведён в Приложении А. Каждый из вариантов определяет предметную область, для моделирования которой должна быть разработана система классов. Допустим выбор иных вариантов тем курсовой работы по предложениям преподавателя, работодателей, студентов. Объём расчётно-

пояснительной записки составляет 20-50 страниц машинописного текста, включая рисунки, таблицы и приложения. Её оформление должно соответствовать требованиям ГОСТ, ЕСКД, ЕСПД.

2. Рекомендации по выполнению работы

2.1. Анализ задания.

Основой курсовой работы 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 источников и в ходе общения 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.

2.2. Проектирование системы 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. Для приведённого выше примера 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 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 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.

Диаграмма классов с комментариями к ней является результатом данного этапа работы.

2.3. Выбор средств реализации библиотеки классов

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 придерживаться, как организовать проектную работу, как документировать процесс программирования, как отслеживать изменения в коде программы. В реальной ситуации данные вопросы решает руководство проекта, исходя из корпоративных стандартов, системных требований и многих других принципов. В курсовой работе набор перечисленных вопросов остаётся, однако от студента требуется обоснование ответа лишь на два из них – о выборе языка программирования и о выборе среды разработки. Рекомендованными в рамках дисциплины языками являются C++ и C#. Рекомендуемой средой разработки является система Microsoft Visual Studio.

Тем не менее, студент имеет право остановиться на каком-либо другом объектно-ориентированном языке высокого уровня, позволяющем разрабатывать независимые приложения. Выбор языка требует обязательного обоснования. В случае выбора одного из рекомендованных языков обоснование выбора среды разработки не требуется – достаточно лишь указать на используемый инструментарий. 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.

2.4. Программная реализация библиотеки классов

На этом этапе на основе диаграммы классов должны быть описаны спроектированные классы. 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 расчётно-пояснительной записке результаты данного этапа необходимо отразить в виде программного кода общего описания классов (объявления классов) и спецификаций методов. Полный исходный программный код с определениями методов классов должен быть помещён к расчётно-пояснительной записке.

3. Содержание пояснительной записки

1. Титульный лист
2. Описание задания
3. Анализ предметной области
4. Диаграмма классов
5. Выбор и обоснование языка программирования
6. Контрольный пример
7. Код программы

<p>1. Система управления местами гостиницы</p> <p>Система заказа номеров в гостинице.</p> <p>Ведение БД: Номера, Услуги, Клиенты</p> <p>Функции: Поселение и выселение клиентов, бронирование мест, учёт оказанных услуг</p> <p>Выходные документы: Счёт за проживание и услуги, Список проживавших на момент времени, Список номеров, Прейскурант услуг.</p>
<p>2. Система отслеживания успеваемости студентов</p> <p>Система ведения результатов успеваемости студентов.</p> <p>Ведение БД: кафедры, студенты, предметы</p> <p>Функции: занесение данных по студентам, формирование справочных документов,</p> <p>Выходные документы: Ведомость успеваемости по группе студентов.</p>
<p>3. Система учета пациентов больницы.</p> <p>Ведение справочников: Пациенты, Болезни, Палаты, Врачи, История болезни</p> <p>Функции: Ведение справочников, приём пациента, ведение истории болезни, выписка.</p> <p>Выходные документы: Список пациентов, Список врачей, Карточка больного.</p>
<p>4. Система управления складом аптеки</p> <p>Ведение справочников: Группы лекарств, Лекарства, Производители, Поставщики</p> <p>Функции: ведение справочников, учёт прихода и продаж лекарств</p> <p>Выходные документы: Отчёт по наличию лекарств на складе по группам, Отчёт по продажам по группам, Счёт-фактура.</p>
<p>5. Система учета заказов клиентов ресторана</p> <p>Ведение справочников: Продукты, Блюда, Заказы</p> <p>Функции: Ведение справочников, хранение рецептов, расчёт себестоимости блюда, приём заказов</p> <p>Выходные документы: Меню, Счёт заказа, Отчёт по продуктам на складе, Заказы за период.</p>