


**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. И.РАЗЗАКОВА**

Утверждено на заседании кафедры ПИ

Протокол № 5 от “25” декабря 2023г.

Зав. каф. ПИ



Орозобекова А.К

Методическое указание  
к выполнению курсовой работы  
по дисциплине «**Проектирование и архитектура  
программных систем 2**»

по направлению 710400 – Программная инженерия

Бишкек 2023

**Мукамбетова С.А., Шаршенбаева А.К., Жангазиева Г.А.**

Методические указания к выполнению курсовой работе по дисциплине «Проектирование и архитектура программных систем 2» для бакалавров по направлению 710400- Программная инженерия/Кырг.гос.тех.универ./сост.: Мукамбетова С.А., Шаршенбаева А.К., Жангазиева Г.А., Бишкек: Кырг. гос. тех. ун-т, 2023–16 с.: ил

Излагаются основные требования к выполнению курсовых работ по дисциплине «Проектирование и архитектура программных систем 2». Приводится краткое изложение ключевых понятий дисциплины и описание нотации языка UML, достаточной для изображения диаграмм классов, иллюстрирующих этапы разработки разрабатываемых программных систем.

Излагаются цели курсовой работы и требования к оформлению пояснительной записки.

Приводятся задания для курсовых работ и пример оформления пояснительной записки курсовой работы.

Используется среда IntelliJ IDEA с применением пакетов Java.

Для студентов вузов по направлению 710400 – Программная инженерия.

УДК 681.3.06

ББК 32.973.26 - 018

## **1. Общие положения**

Курсовая работа предусмотрена стандартом дисциплины «Проектирование и архитектура программных систем». Курсовая работа является самостоятельной работой студента, позволяет оценить качество знаний и отражает приобретенные студентом практические навыки.

Цель курсовой работы: расширить объем знаний студентов в области программирования и создать реальную основу использования своих знаний для решения на компьютере задач по другим дисциплинам и в своей дальнейшей практической деятельности.

Тематика курсовой работы: разработка системы классов для решения задач из некоторой предметной области и приложения для тестирования созданной системы классов. Список вариантов задач для конкретных предметных областей приведен в приложении А.

Тема назначается руководителем курсовой работы и утверждается на заседании учебно-методической комиссии кафедры. Задание на курсовую работу содержит сведения согласно приложению Б и оформляется на бланке формата А4.

## **2 Выполнение курсовой работы**

Выполнение курсовой работы состоит из четырех этапов:

1. Постановка задачи;
2. Разработка системы классов предметной области
3. Разработка тестовой программы и тестирование системы классов;
4. Оформление и защита курсовой работы.

Выполнение этапов курсовой работы должно соответствовать срокам, установленным в задании.

Все этапы разработки программы отражаются в пояснительной записке.

Содержание пояснительной записки:

- Титульный лист
- Задание на курсовую работу
- Содержание
- Введение
- Основная часть
- Заключение
- Список использованных источников
- Приложение А. Система классов
- Приложение Б. Листинг тестовой программы

Основная часть пояснительной записки должна состоять из следующих разделов:

1. Постановка задачи
2. Разработка системы классов предметной области
  - 2.1 Диаграмма классов
  - 2.2 Реализация классов
3. Разработка тестовой программы
  - 3.1 Методика тестирования
  - 3.2 Структура тестовой программы
4. Результаты тестирования

## **2.1 Постановка задачи**

Основой курсовой работы является некоторая предметная область со своими терминами, понятиями, субъектами и объектами, отношениями между ними. Необходимо провести анализ предметной области. В ходе анализа предметной области необходимо на основе знакомства с литературными источниками и общения с заказчиком выявить:

- Какие в предметной области есть термины и понятия, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности.

- Что входит в словарь предметной области, отдельно выделив список существительных и список глаголов, которые могут быть связаны с существительными.
- Каковы функциональные требования к разрабатываемой системе классов. 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. В рамках курсовой работы в роли заказчика выступает преподаватель, выдавший задание. Студент проводит анализ предметной области, основываясь на своих собственных знаниях, литературных источниках 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 Разработка системы классов предметной области**

Разработка 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 для выделения классов и их полей, а список глаголов – для определения методов. Следующий шаг состоит в том, чтобы определить, какой из классов какие поля и методы содержит. Далее необходимо выявить отношения между классами.

Результаты такого анализа должны быть оформлены в виде диаграммы классов в нотации UML. В UML между классами имеются четыре разновидности отношений: ассоциация (ее разновидности агрегация и

композиция), обобщение (наследование), зависимость и реализация. Следует обратить внимание, что понятие “наследование” чаще всего возникает тогда, когда разные классы обладают частично схожими наборами полей 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, полей, методов, их соответствие, и, возможно, выделить ряд новых классов, связанных с имеющимися отношениями наследования.

При оформлении диаграммы классов следует придерживаться следующих правил:

- класс обозначается прямоугольником;
- прямоугольник делится на три части, в каждой из которых, соответственно, указываются: имя класса, список полей, список методов;
- имена классов, полей и методов могут быть записаны на русском языке, но в соответствии с нормами написания идентификаторов (одно слово, включающее буквы, цифры, символ подчеркивания и не начинающееся с цифры);
- имена классов записываются с заглавной буквы, имена полей и методов – со строчной);
- перед именем поля или метода ставится символ, указывающий на режим доступности: закрытый (-), открытый (+), защищенный (#);
- после имени метода ставятся круглые скобки, в которых могут быть перечислены параметры метода;
- при обозначении отношений между классами нужно использовать синтаксис UML.

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

Далее на основе диаграммы классов должны быть описаны спроектированные классы на языке C++ (выполнена программная реализация

системы классов). Грамотно спроектированная диаграмма классов позволяет очень легко написать программный код, содержащий общее описание классов (объявление классов). 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 пояснительной записке результаты данного этапа необходимо отразить в виде программного кода общего описания классов (объявления классов) и описаний методов. Определения методов приводить в основном тексте пояснительной записки не следует. Полный исходный программный код с определениями методов классов должен быть помещен в приложение А к пояснительной записке. Следует обратить внимание, что речь идет только о программном коде, реализующем систему классов предметной области. Код тестовой программы должен быть помещен в приложение Б.

### **2.3 Разработка тестовой программы**

Перед разработкой программы тестирования системы классов предметной области, необходимо определиться с методикой тестирования.

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 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 упрощенный вариант модульного тестирования, сводящийся к тестированию всех методов разработанной системы классов. Под методикой тестирования в курсовой работе будем понимать список различных вариантов вызова методов классов с ожидаемыми результатами и порядок тестового выполнения этих вызовов.

Процесс проектирования и программной реализации тестового приложения очень похож на описанный выше процесс разработки системы классов. Однако, по своему наполнению он может сильно варьироваться для разных тем курсовых работ. Поэтому данный этап курсового проектирования предполагает больший объём самостоятельной работы, связанный с необходимостью изучения значительного количества справочной информации, не рассматриваемой в рамках основного курса.

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

- Тестовое приложение должно иметь оконный интерфейс, включающий меню и интерактивные интерфейсные средства, с помощью которых можно протестировать созданную систему классов.
- Набор интерфейсных элементов (кнопки, списки, поля ввода, диалоговые окна, графические элементы и т.д.) должен быть определён на основе описанной на предыдущем этапе методики тестирования.
- Жёсткие требования к качеству интерфейса тестового приложения (с точки зрения удобства использования и степени интерактивности) не предъявляются. Основное требование – возможность реализации методики тестирования.
- Жёсткие требования на документирование процесса разработки тестового приложения не накладываются. В текст расчётно-пояснительной записки следует обязательно включить лишь 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.

- Полный исходный программный код модулей тестового приложения (кроме описания системы классов предметной области) должен быть помещён в приложение Б к расчётно-пояснительной записке.

## **2.4 Тестирование системы 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 пояснительной записки (например, указать, чем вызвано несоответствие – семантическими или алгоритмическими ошибками, сформулировать рекомендации по исправлению несоответствия, и т.п.).

### **3. Защита курсовой работы**

Курсовая работа представляется на проверку руководителю поэтапно или полностью выполненная. При обнаружении принципиальных ошибок, незаконченном или небрежно оформленной работе она возвращается на доработку в назначенные сроки.

Возвраты на доработку или увеличение периода разработки без уважительных причин снижают оценку при защите курсовой работы.

Для защиты предъявляется пояснительная записка, выполненная в соответствии с требованиями и отлаженная программа.

Защищая курсовую работу, студент коротко докладывает руководителю о проделанной работе и ее результатах, демонстрирует работу приложения на компьютере и отвечает на вопросы.

## ПРИЛОЖЕНИЕ А

### **1. «Библиотека»**

В системе должны поддерживаться режимы поиска книги по заданному критерию (автор, название), заказа книги, учета клиентов и книг в книгохранилище, выдачи информации по запросам (местонахождение книги в архиве или ее отсутствие), выдачи информации о должниках.

### **2. «Магазин»**

В системе должны поддерживаться режимы заказа товара (продовольственных товаров), покупки и учета товаров, анализа покупаемости товара, анализа покупаемости продуктов в зависимости времени дня и дня недели.

### **3. «Дом»**

В системе должны поддерживаться режимы учета жильцов и учета доходов и расходов, связанных с проживанием жильцов в доме (аренда, электроэнергия, ком. услуги и т. д.), выдачи аналитической информации, выдачи списка жильцов.

### **4. «Гостиница»**

В системе должны поддерживаться режимы учета и распределения по номерам приезжих гостей в зависимости от требований и пожеланий проживающих.

### **5. «Театр»**

В системе должны поддерживаться режимы учета спектаклей, актеров, играющих в спектаклях, концертов, распределения мест и стоимости билетов, анализа популярности спектаклей по различным критериям.

### **6. «Успеваемость студентов на факультете»**

В системе должны поддерживаться режимы учета учащихся и результатов сдачи экзаменов, анализа сессии по семестрам, по факультетам, специальностям, генерации отчетов отличников и двоечников.

## **7. «Водительские курсы»**

В системе должны поддерживаться режимы учета учащихся на курсах в зависимости от категории, посещаемости, тренировок вождения, учета ошибок вождения, предварительных результатов тестов и результатов сдачи экзаменов, анализа популярности курсов в зависимости от стоимости, продолжительности и качества курсов.

## **8. «Магазин (аудио-видео продукции)»**

В системе должны поддерживаться режимы учета товара в магазине по категориям (аудио, видео, CD или DVD диски, кассеты), по темам (фильмы, музыка по категориям), заказа товара и анализа популярности продукции и в зависимости от этого дополнительного заказа.

## **9. «Конкурс поэтов»**

В системе должны поддерживаться режимы учета конкурсантов, их творческого багажа, их опубликованные работы, порядка проведения, сценария выступления каждого и в целом в зависимости от темы и анализа конкурсов по годам, учета посещаемости и участия в конкурсе.

## **10. «Конференция»**

В системе должны поддерживаться режимы учета выступающих студентов, аспирантов и преподавателей по тематикам, по специальностям, регистрации участников и гостей на конференции, учета длительности и новизны тем, анализа конференции.

## **11. «Фестиваль фильмов»**

В системе должны поддерживаться режимы учета гостей, участников и жюри, фильмов и номинаций по категориям, регистрации участников и проведения голосования с учетом зрительских симпатий и рейтинга популярности фильмов. Оценка по различным критериям должна зависеть от категории (сложность, глубина и т. д.).

## **12. «Центр занятости»**

В системе должны поддерживаться режимы учета безработных, их стажа, квалификации, желания работать по определенной специальности, места расположения и заработной платы, учета уже стоящих на учете в центре занятости, анализа занятости от времени, специальности и т. д.

### **13. «Больница»**

В системе должны поддерживаться режимы учета больных по отделениям, заболеваниям, сложности заболевания и количеству заболеваний у одного человека, продолжительности болезни, количеству койко-мест и анализа заболеваемости по районам и категориям.

### **14. «Автобусный парк»**

В системе должны поддерживаться режимы учета транспортных средств в автопарке, маршрутов, водителей, учета доходов и расходов (оплата за проезд, ремонт и т. п.), выдачи отчетов по запросам.

### **15. «Расписание занятий в университете»**

В системе должны поддерживаться режимы поиска занятия по заданному критерию (время, преподаватель), регистрации занятий, учета занятий по типу, генерации расписаний.

### **16. «Школа»**

В системе должны поддерживаться режимы учета классов и учеников в них, регистрации нового ученика, учета посещаемости занятий и оценок учащихся, генерации отчетов по успеваемости учеников.

### **17. «Поликлиника»**

В системе должны поддерживаться режимы учета докторов и пациентов, поиска пациента по заданному критерию, анализа заболеваемости по районам, по месяцам и выдачи справок о болезни.

### **18. «Аптека»**

В системе должны поддерживаться режимы поиска лекарства по заданному критерию (название, болезнь, цена), заказа, покупки и учета лекарств, анализа

спроса на лекарства в зависимости от стоимости, времени года и т. п., выдачи соответствующих отчетов.

### **19. «Музей»**

В системе должны поддерживаться режимы учета посетителей и экспонатов музея, регистрации новых экспонатов, поиска экспонатов по заданному критерию (название, эпоха и т. д.), учета доходов и расходов, связанных с проведением экскурсий, выдачи отчетов по запросам.

### **20. «Ресторан»**

В системе должны поддерживаться режимы заказа и поиска блюда по заданному критерию (название, цена), анализа спроса на различные блюда в зависимости от цены, времени дня, дня недели, времени года и т. п., выдачи соответствующих отчетов, генерации меню.

### **21. «Туристическая фирма»**

В системе должны поддерживаться режимы учёта клиентов и продаваемых путевок, поиска «горячих» путевок, выдачу информации самых популярных маршрутов, выдачу информации о маршрутах и путевках по критериям.

### **22. «Станция техобслуживания»**

В системе должны поддерживаться режимы учета работников и клиентов станции, оказываемых услугах по ремонту автомобилей, выдачи клиенту счета в котором содержится перечень устраненных неисправностей и стоимости работы, а так же сведения о исполнителях.

### **23. «Почтовое отделение»**

В системе должны поддерживаться режимы учета почтальонов и обслуживаемых ими участков, подписчиков газеты журналов, поиск информации о подписных изданиях по различным критериям, прием и увольнение почтальонов.

### **24. «Хлебокомбинат»**

В системе должны поддерживаться режимы учета производимых изделий их продаж, а так же учет закупок необходимых для производства продуктов,

анализ спроса на производимую продукцию в зависимости от стоимости, получение суммарной прибыли от продаж, поиск информации о продукции по различным критериям.

## **25. «Соревнования по футболу»**

В системе должны поддерживаться режимы учета команд, участвующих в соревновании, игроках команд, расписание встреч и их результатах, цены билетов на игры, получение отчетов с результатами игр, поиска информации о командах и играх по различным критериям.

ПРИЛОЖЕНИЕ Б  
ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

По дисциплине Проектирование и архитектура программных систем 2

Студенту группы \_\_\_\_\_

\_\_\_\_\_

фамилия, имя, отчество

Тема: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Этап разработки	Срок Выполнения
1. Постановка задачи	4 — 7 недели семестра
2. Разработка системы классов предметной области	8 — 11 недели семестра
3. Разработка тестовой программы и тестирование системы классов	12 — 15 недели семестра
4. Оформление и защита курсовой работы.	16,17 неделя семестра

Руководитель работы \_\_\_\_\_

фамилия, имя, отчество

Подпись руководителя

Подпись студента

\_\_\_\_\_

\_\_\_\_\_

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.