# МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО КУРСОВОМУ ПРОЕКТИРОВАНИ ДИСЦИПЛИНЫ «КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Оглавление

[МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО КУРСОВОМУ ПРОЕКТИРОВАНИ ДИСЦИПЛИНЫ «КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ» 1](#_Toc146381764)

[Средства выполнения 1](#_Toc146381765)

[Разделы задания по курсовому проектированию 2](#_Toc146381766)

[Раздел 1 Обзор и описание предметной области 2](#_Toc146381767)

[Раздел 2 Проектирование программного обеспечения 3](#_Toc146381768)

[Раздел 3 Реализация программного обеспечения 4](#_Toc146381769)

[Раздел 4 Тестирование программного обеспечения 5](#_Toc146381770)

[Заключение 5](#_Toc146381771)

[Приложение А 6](#_Toc146381772)

## Средства выполнения

1. Система контроля версий Git и т.п., в также удаленный репозиторий GitHub и т.п.
2. Объектно-ориентированный, компилируемый язык программирования высокого уровня, например: C#, Java, C++.
3. Интегрированные системы разработки программного обеспечения, например: Visual Studio, IntelliJ IDEA, Eclipse и т.п.
4. Инструменты для разработки UML-диаграмм и генерации исходного кода с помощью этих диаграмм, например: Диаграммы классов (как плагин Visual Studio), Umple (как плагин Eclipse) и т.п.

Все средства выполнения выбираются студентом самостоятельно по его усмотрению.

## Разделы задания по курсовому проектированию

**4. Содержание пояснительной записки (перечень вопросов, которые подлежат разработке)**

|  |
| --- |
| Лист задания |
| Содержание |
| Введение |
| 4.1 Обзор и описание предметной области |
| 4.2 Постановка задачи |
| 4.3 Проектирование программного обеспечения |
| 4.4 Реализация программного обеспечения |
| 4.5 Тестирование программного обеспечения |
| Заключение |
| Список использованной литературы |
| Приложение А Графическая часть |
| Приложение Б Код программы |

**5. Перечень графического материала (с точным указанием обязательных чертежей и графиков)**

|  |
| --- |
| 3-5 листов формата А4 |
| Интерфейс пользователя |
| Схемы алгоритмов |

## Раздел 1 Обзор и описание предметной области

Определите и опишите сущности предметной области.

Нужно разработать приложение с использованием методологии ООП. Для чего необходимо подобрать самостоятельно соответствующую проблемную (предметную/доменную) область, которая базируется на объектах и событиях реального мира (примеры соответствующих предметных областей приведены ниже в [Приложение А](#_Приложение_А).

Прежде чем начать проектирование и конструирование приложения исследуйте выбранную предметную область и определите основные требования к программе. Для этого используйте возможности интернета и посетите два, три сайта по соответствующей теме. Скриншоты сайтов и ссылки на них представить в записке.

Определите основные сущности предметной области и поведение программы. Кратко опишите это в отчете.

## Раздел 2 Проектирование программного обеспечения

Опишите постановку задачи. В этой части проекта укажите что должна делать ваша программа. Опишите требование к программе на основе анализа предметной области. Определите методы и модели, положенные в основу проекта.

Дайте краткое описание технологий, языка программирования и средств разработки, которые вы выбрали. Обоснуйте почему именно эти средства наиболее лучше всего подходят для реализации вашего приложения.

При проектировании и дальнейшей разработке ПО используйте основные принципы ООП (абстракция, инкапсуляция, наследование, полиморфизм).

Спроектируйте классы (собственные пользовательские типы данных) в языке программирования для программного представления данных объектов и основной логики будущей программной системы.

Система должна решать, полезные действия и иметь дополнительно следующие опции:

* не менее 3 разнообразных классов предметной области;
* не менее 5 атрибутов (состояния) и 3-х методов (поведения) в классе-сущности;
* не менее 3 методов, которые реализуют бизнес-логику программы, в соответствующих функциональных классах;
* при работе в команде с количеством более 3-х человек, количество классов и методов должно быть не менее численности команды.
* хранить глобальные характеристики системы или характеристики уровня отдельных классов.

Необходимо спроектировать и реализовать UML-диаграмму взаимодействия классов и объектов разрабатываемой программной системы с отображением всех связей (отношений) между классами и объектами.

Основные классы системы должны быть самодостаточными, т.е. не зависеть, к примеру, от консоли! Любые типы отношений между классами должны применяться обосновано и лишь тогда, когда это имеет смысл.

10. Рекомендуется придерживаться принципов SOLID (GRASP), особенно принципа единственной ответственности (Single Responsibility Principle, SRP): у каждого пакета, класса или метода должна быть только одна ответственность (цель), т.е. должна быть только одна причина изменить в дальнейшем соответствующий блок кода. Классы рекомендуется проектироваться и реализовываться таким образом, чтобы они были слабо зависимы от других классов.

Создаваемые классы необходимо грамотно разложить по соответствующим пакетам, которые должны иметь «*адекватные*» названия и быть вложены в указанные стартовые пакеты.

## Раздел 3 Реализация программного обеспечения

Приложение должно быть построено по *архитектурному шаблону* наиболее подходящей для выбранной вами технологии, например, Model-View-Controller (MVC).

Также в архитектуре приложения должна быть модульность, для упрощения его дальнейшего сопровождения и масштабирования. С этой целью используйте статические или динамические *пользовательские библиотеки*.

Для взаимодействия с пользователем в программе должен быть дружелюбный и интуитивно понятный интерфейс. Интерфейс программы и комментарии желательно, должны быть на английском языке.

Программа должна быть интерактивна. В приложении разработайте меню выбора действий пользователя. Должен быть предусмотрен *корректный выход* из программы.

В соответствующих компонентах бизнес-логики необходимо предусмотреть «защиту от дурака». Предусмотрите проверку корректного ввода данных.

При разработке программ придерживайтесь соглашений по написанию кода на языке программирования разрабатываемого проекта.

В приложении обязательно должны быть следующий функционал и технологии:

ПОТОКИ ВВОДА-ВЫВОДА

Приложение должно импортировать и экспортировать данные в текстовые файлы и электронные таблицы, например, таблицы EXEL.

ЖУРНАЛИРОВАНИЕ

Приложение должно сохранять результирующие данные в лог-файлы, а также вести журналирование любых исключительных ситуации, которые могут возникнуть в процессе работы программы.

СЕРИАЛИЗАЦИЯ И ДЕСЕРИАЛИЗАЦИЯ ОБЪЕКТОВ

Приложение должно упаковывать и распаковывать данные в бинарные файлы и файлы формата разметки с использованием стандартной и пользовательской («кастомной») сериализации и десериализации.

ЛОКАЛИЗАЦИЯ ИНТЕРНАЦИОНАЛЬНЫХ ДАННЫХ

Приложение должно поддерживать нескольких языков для локализации приложения.

ОБРАБОТКА ТЕКСТА И РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

Интерактивное приложение должно иметь функцию для анализа текста.

Для разбиения тестовой информации необходимо использовать регулярные выражения.

Предусмотрите восстановление текста в первоначальный (оригинальный) вид из программных объектов-сущностей. Оригинальный текст и текст, который получается в результате восстановления, должны полностью совпадать. Это и будет показателем высокого качества разработанного класса-парсера.

РАЗВЕРТЫВАНИЕ ПРИЛОЖЕНИЙ

В заключительной части раздела по реализации программного обеспечения приложение должно быть развернуто на компьютере конечного пользователя. Для этого используйте технологии *создания пакета установки программ*.

## Раздел 4 Тестирование программного обеспечения

Весь код модели системы должен быть покрыт модульными тестами, подтверждающие качество и работоспособность ядра системы. Необходимо проверить все тестовые случаи. Для реализации модульных тестов рекомендуется использовать jUnit фреймворк.

Необходимо:

* создать отдельный тестовый подпроект в рамках текущего программного решения;
* максимально покрыть основную бизнес-логику модульными тестами;
* реализовать тестовые планы, которые будут включать как методы, которые будут вызываться перед и после каждого тестового случая, так и методы, которые должны вызываться на уровне всего класса;
* предусмотреть также тестирование методов на исключительные ситуации и временные интервалы.

## Заключение

Дайте краткое описание результатов выполнения предыдущих разделов.

# Приложение А

**Примеры предметных областей**

1. **Цветочница или магазин цветов** (Flower Shop).
2. **Продуктовый магазин** (Grocery Store).
3. **Новогодняя ёлка** (Christmas tree).
4. **Кредиты** (Credits).
5. **Жилищно-Коммунальное Хозяйство, ЖКХ** (Housing and Communal Services).
6. **Турагентство и Туристические путевки** (Tourist trips).
7. **Ресторан** (Restaurant).
8. **Шеф-повар** (Chef).
9. **Налоги** (Taxes).
10. **Игровая комната** (Game room).
11. **Камни** (Stones).
12. **Звукозапись** (Sound Recording).
13. **Компьютерный герой** (Game Hero).
14. **Новогодний подарок** (New Year gift).
15. **Страховое агентство** (Insurance Company).
16. **Видео/Компьютерная игра** (Video/Computer Game).
17. **Авиакомпания (Airline).**
18. **Железнодорожный транспорт** (Railway Transport).
19. **Автосалон** (Car Center).
20. **Таксопарк** (Taxi Station).
21. **Автостоянка** (Parking).
22. **Грузоперевозки** (Сargo Transportation).
23. **Пассажирские перевозки** (Passenger Operations).
24. **Фургон кофе** (Coffee Car).
25. **Морской порт** (Sea Port).
26. **Рыцарь** (Knight).
27. **Мобильная связь** (Mobile Communication).
28. **Футбольный Менеджер** (Football Manager System).
29. **Вклады или кредиты** (Deposits or Credits).
30. **Зоопарк** (Zoo).
31. **Спортивная рыбалка** (Sports Fishing).
32. **Компания по разработки программного обеспечения, IT-компания** (Software Company).
33. **Домашние электроприборы** (House Equipments).
34. **Пчелиная пасека** (Bee Apiary).
35. **Поликлиника** (Clinic/Hospital).
36. **Библиотека** (Library).
37. **Тюрьма** (Jail/Prison).
38. **Суд или судебное делопроизводство** (Court).
39. \* **Придумайте** свою предметную область, которая вам интересна.