

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский ядерный университет «МИФИ»  
Озерский технологический институт -  
филиал федерального государственного автономного образовательного учреждения высшего  
образования «Национальный исследовательский ядерный университет «МИФИ»  
(ОТИ НИЯУ МИФИ)

*Кафедра прикладной математики*

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению курсовой работы по дисциплине  
«Программирование (Объектно-ориентированное программирование)»

---

специальность:	<u>230106 (09.05.01) Применение и эксплуатация автоматизированных систем специального назначения</u>
Профиль подготовки:	<u>Математическое, программное и информационное обеспечение вычислительной техники и автоматизированных систем</u>
Квалификация	<u>инженер</u>
Форма обучения:	<u>очная</u>

УТВЕРЖДЕНО  
Редакционно-издательским  
ОТИ НИЯУ МИФИ

г. Озерск, 2014 г.

Методические указания к выполнению курсовой работы по дисциплине «Программирование (Объектно-ориентированное программирование)». Подготовил Вл. Пономарев. Озерск: ОТИ НИЯУ МИФИ, 2014. — 28 с.

Методическое пособие предназначено для студентов направления подготовки 230106 (09.05.01) «Применение и эксплуатация автоматизированных систем специального назначения» и включает в себя требования к выполнению, содержанию и оформлению курсовой работы по дисциплине «Программирование (Объектно-ориентированное программирование)», а также рекомендации по ее защите.

Рецензенты:

- 1) Е.В. Синяков, ФГУП «ПО «Маяк».
- 2) А.Ф. Зубаиров, ст. преподаватель кафедры прикладной математики ОТИ НИЯУ МИФИ

## Содержание

1 Основные положения .....	4
1.1 Цели и задачи курсового проектирования .....	4
1.2 Тематика курсового проектирования .....	5
1.3 Руководство проектом .....	6
1.4 Структура и содержание курсовой работы .....	6
2 Выполнение курсовой работы .....	9
2.1 Планирование времени .....	9
2.2 Исследование предметной области .....	10
2.3 Проектирование иерархии классов .....	11
2.4 Разработка программного модуля .....	12
2.5 Разработка тестирующего приложения .....	12
3 Оформление курсовой работы .....	13
3.1 Пояснительная записка .....	13
3.2 Компьютерная презентация .....	18
4 Защита курсовой работы .....	20
5 Обязательные требования к курсовой работе .....	21
6 Критерии оценки курсовой работы .....	22
7 Примерные темы курсовых работ .....	23
Библиография .....	25
Приложение А (справочное) Титульный лист .....	26
Приложение Б (справочное) Лист аннотации .....	27
Приложение В (справочное) Лист содержания .....	28

# 1 Основные положения

## 1.1 Цели и задачи курсового проектирования

Целями курсового проектирования являются:

- закрепление знаний, полученных в ходе теоретического и практического изучения дисциплины «Программирование (Объектно-ориентированное программирование)»;

- приобретение навыков использования современных CASE-средств и языков моделирования, используемых при проектировании объектно-ориентированных программ и систем;

- приобретение навыков практического программирования с использованием объектно-ориентированной парадигмы;

- изучение современных систем программирования и сред для разработки объектно-ориентированных программ и систем;

- изучение отдельных разделов предметной области, не вошедших в программу теоретического обучения, формирование навыка поиска информации по конкретной теме, ее анализа и использования для решения задачи;

- подготовка к выполнению выпускной квалификационной работы.

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.

Задачей курсового проектирования является разработка иерархии типов в заданной предметной области, включающая в себя:

- разработку модели классов на языке UML;

- разработку диаграмм взаимодействия на языке UML;

- разработку абстрактных классов и интерфейсов;

- генерацию кода классов;

- разработку тестирующего приложения;

- разработку документации;

- подготовку презентации.

## 1.2 Тематика курсового проектирования

В качестве темы для курсового проектирования по дисциплине «Программирование (Объектно-ориентированное программирование)» студентам могут быть предложены для разработки разные модели иерархий типов. Примерная тематика курсовых работ:

1) проектирование классифицирующей иерархии типов в некоторой абстрактной предметной области, такой, как «Растительный мир»;

2) проектирование иерархии типов в некоторой реальной предметной области, такой, как «Почтовая служба»;

3) программы различного назначения, реализующие объектно-ориентированную парадигму программирования в той или иной мере;

4) исследовательские проекты; они включают в себя разработку различных программных компонентов, связанных с языками программирования и средами разработки. Примерами могут служить интерпретация языка UML, средства визуального программирования и т.п.

В состав проекта могут быть включены научно-исследовательские работы, в которых студент принимал участие по линии учебно-исследовательской работы студентов (УИРС), в научных учреждениях и на производстве при прохождении практики. Объем этих работ устанавливается руководителями проекта.

В случаях, когда сложность проекта достаточно велика и (или) необходима подробная разработка тем, кафедра имеет право выдавать комплексные задания на курсовое проектирование группе студентов. Такими заданиями могут быть проекты по созданию перспективных разработок в области языков программирования и сред разработки программного обеспечения.

Для проектирования диаграмм классов могут быть использованы как специализированные программные средства, так и средства, встроенные в современные системы программирования и проектирования.

В качестве языка программирования в курсовой работе допускается использование различных объектно-ориентированных языков, таких, как C++, C#, Java и других.

### 1.3 Руководство проектом

Общее методическое руководство курсовым проектированием осуществляется кафедрой прикладной математики, а непосредственно курсовой работой — его руководителем.

Кафедра определяет требования к содержанию курсовой работы, контролирует ход ее подготовки, обеспечивает студентов методическими материалами, информирует заместителя директора по учебной работе о выполнении графиков выполнения курсовых работ и степени их готовности и осуществляет подбор руководителей.

По предложению руководителей кафедра может пригласить консультантов по отдельным разделам курсовой работы. Консультантами могут быть преподаватели высших учебных заведений и высококвалифицированные инженеры и специалисты.

Руководитель курсовой работы выдает задание на проектирование, согласовывает план и график выполнения курсовой работы, оказывает методическую помощь в подборе литературы, справочных материалов, консультирует студента, дает письменный отзыв на курсовую работу.

При невыполнении студентом графика по представлению руководителя кафедра имеет право не допустить работу к защите. В этом случае заведующий кафедрой после ознакомления с отзывом руководителя и содержанием курсовой работы принимает решение о допуске работы к защите. При этом вопрос рассматривается коллегиально на заседании кафедры с участием студента и руководителя курсовой работы.

### 1.4 Структура и содержание курсовой работы

Курсовая работа включает в себя:

- 1) демонстрационную версию программного продукта;
- 2) пояснительную записку;
- 3) компьютерную презентацию.

Составными частями пояснительной записки являются:

- 1) титульный лист;
- 2) лист оценки курсовой работы;
- 3) рецензия;

- 4) задание на курсовое проектирование;
- 5) аннотация на русском языке (одна страница);
- 6) содержание;
- 7) список обозначений и сокращений (при необходимости);
- 8) основной текст;
- 9) список использованных источников;
- 10) приложения.

Примечание: дискета или компакт-диск с демо-версией помещается в специальный бумажный карман на задней обложке пояснительной записки.

Пояснительная записка выполняется в программе Microsoft Word с помощью шаблона, предоставляемого кафедрой. Объем пояснительной записки составляет от 30 до 60 страниц текста (без учета приложений), выполненного в соответствии с требованиями ЕСКД. Руководящим материалом к оформлению пояснительной записки служит методическое пособие [1].

Основной текст пояснительной записки состоит из введения, разделов основной части и заключения.

Во введении обосновывается актуальность проекта, раскрываются его цель и задачи, приводится краткое описание содержания последующей основной части пояснительной записки. Рекомендуемый объем введения 1-2 страницы.

Основная часть текста включает в себя описание предметной области и постановку задачи, проектную, практическую и экспериментальную части. Объем основной части составляет примерно 60% текста пояснительной записки. В ней приводится подробное описание выполненной работы, структурированное по разделам и подразделам. Каждый раздел и подраздел должен иметь свое наименование и отражаться в содержании.

Рекомендуемая структура разделов и подразделов основной части:

- 1 Предметная область
  - 1.1 Анализ задачи
  - 1.2 Требования к проекту

## 2 Проектная часть

### 2.1 Средства проектирования

### 2.2 Абстрагирование и выделение классов

### 2.3 Моделирование иерархии классов

### 2.4 Построение диаграмм использования

### 2.5 Построение диаграмм потоков данных и действий

## 3 Практическая часть

### 3.1 Структура программного проекта

### 3.2 Проектирование классов сущностей

### 3.3 Проектирование управляющих классов

### 3.4 Проектирование интерфейсных классов

## 4 Экспериментальная часть

### 4.1 Цели тестирования

### 4.2 Структура тестирующей программы

### 4.3 Тесты

В разделе «Предметная область» приводится анализ задачи на основании выданного задания, описывается исследование предметной области, выявляются требования к проекту, составляется техническое задание. Рекомендуемый объем этого раздела 3-6 страниц.

Проектная часть содержит описание процесса проектирования иерархии классов и интерфейсов на языке UML, разработку диаграмм использования и взаимодействия. Рекомендуемый объем этого раздела 8-10 страниц.

Практическая часть содержит описание процесса разработки иерархии классов и интерфейсов, с указанием свойств и методов классов и интерфейсов, содержит обоснование инкапсуляции, наследования и полиморфизма. Рекомендуемый объем этого раздела 15-20 страниц.

Заключение должно содержать выводы по выполнению задания на проект и соответствовать введению в смысле достижения указанных в нем поставленной цели и задач проектирования. Следует отметить преимущества, связанные с реализацией проектных предложений, отметить недостатки работы, дать практические рекомендации по совершенствованию

ванию объекта проектирования, охарактеризовать перспективы дальнейшего развития работы. Рекомендуемый объем раздела 1 страница.

Список использованных источников и количество приложений формально не ограничены. Рекомендуемый объем используемых литературных источников от 5 до 20 наименований. Сведения об использованных источниках должны быть выполнены в соответствии с требованиями ГОСТ 7.1-2003.

Демонстрационная версия программного продукта предоставляется на дискете или компакт-диске, который прилагается к пояснительной записке. Длительность демонстрации программного продукта при защите курсовой работы составляет 5-7 минут.

Для защиты курсовой работы рекомендуется использовать компьютерную презентацию в виде последовательности слайдов. Демонстрация графического и иллюстративного материала в этом случае выполняется на мультимедийном проекторе.

## 2 Выполнение курсовой работы

### 2.1 Планирование времени

Перед началом проектирования следует проанализировать, из каких этапов состоит проект, оценить сложность этапов и правильно распределить время, отводимое для выполнения курсовой работы, между отдельными этапами.

Можно выделить следующие основные отдельные составляющие части курсовой работы:

- а) исследование предметной области;
- б) проектирование иерархии классов на языке UML;
- в) разработка программного модуля, содержащего иерархию классов и интерфейсов;
- г) разработка тестирующего приложения и выполнение тестов;
- д) оформление пояснительной записки и подготовка презентации.

Учитывая, что на разработку курсовой работы отводится до 14 учебных недель, на выполнение одного этапа, при равном распределении времени между этапами, отводится около 20 календарных дней.

Рекомендуемое распределение времени на выполнение курсовой работы приведено в таблице 2.1.

Таблица 2.1

Этап	Время на выполнение
Исследование предметной области	2 недели
Проектирование иерархии классов	4 недели
Разработка программного модуля	4 недели
Разработка тестирующего приложения	2 недели
Пояснительная записка и презентация	2 недели

Приведенное в таблице 2.1 распределение времени является ориентировочным и соответствует заданию на проектирование программного модуля для использования в некоторой реальной предметной области. В других проектах может отсутствовать, например, разработка тестирующего приложения, если проект сам является приложением. В этих проектах большее время отводится на разработку программного модуля и проектирование интерфейса приложения, а пояснительная записка может содержать раздел «Разработка интерфейса» или подобный вместо раздела «Экспериментальная часть».

В любом случае распределение времени на выполнение этапов проектирования является предметом тщательного анализа, который выполняется студентом совместно с руководителем проекта во время выдачи задания на курсовое проектирование.

## 2.2 Исследование предметной области

В ходе анализа предметной области следует определить цели проектирования и круг решаемых задач. Используя литературные и иные источники информации по предметной области, нужно выявить:

- чем является предметная область, какие понятия и термины используются в ее рамках, какие объекты и субъекты составляют ее основу, как взаимодействуют субъекты, как используются объекты и т.п.

- из чего состоит словарь предметной области, выделив список существительных и связанных с ними глаголов. Существительные в конечном итоге определяют названия объектов или свойств, глаголы определяют возможные действия;

- функциональные требования к разрабатываемому программному продукту. В их основе лежат потребности условного заказчика, которые необходимо соотнести с техническими возможностями.

Результатом анализа является составление технического задания, подписываемого заказчиком и исполнителем. В рамках курсового проектирования такой документ не подписывается, и составляется как подраздел раздела «Предметная область» пояснительной записки. Этот подраздел должен описывать требования к конечному продукту, очерчивать границы реализуемых функций, определять состав конечного продукта.

### 2.3 Проектирование иерархии классов

Следующий этап работы соответствует разделу «Проектная часть» пояснительной записки. Используя принципы объектно-ориентированного анализа и проектирования, здесь следует с помощью объектно-ориентированной декомпозиции и абстрагирования выделить существующие классы и действующие объекты, свойства и методы классов, выполнить инкапсуляцию, обосновать наследование, выявить полиморфизм, определить существующие отношения между классами.

На этом этапе разрабатывается иерархия классов (или несколько иерархий классов) в виде диаграммы, определяются классы сущностей, управляющие классы и интерфейсные классы. Классы сущностей описывают сущности предметной области. Управляющие классы предназначены для организации взаимодействия классов сущностей и интерфейсных классов. Интерфейсные классы обеспечивают взаимодействие программного модуля с внешней средой, например, с пользователем.

Следующая часть этого этапа — построение диаграмм использования UseCase и диаграмм взаимодействия, например, диаграмм последовательности. Выбор тех или иных диаграмм определяется разработчи

ком самостоятельно, исходя из потребностей. Эта часть дает возможность выявить алгоритмические аспекты проектируемого модуля.

Результатом этапа проектирования является документированное описание разрабатываемого продукта на языке UML.

## 2.4 Разработка программного модуля

Этот этап работы соответствует разделу «Практическая часть» пояснительной записки. В рамках этапа следует 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.

Следует заметить, что раздел «Практическая часть» пояснительной записки является наиболее объемным и практически важным. Здесь описывается выполнение основной части работы, связанное непосредственно с программированием и формированием структуры программного продукта. Каждое действие в рамках разработки программного модуля (модулей) должно быть обосновано и задокументировано.

Все классы должны быть описаны с указанием свойств и методов, для сложных методов, если такие есть, должны быть приведены алгоритмы. Все действия разработчика в этой части должны выполняться в соответствии с результатами проектирования, а описание разработки должно показывать связь между элементами проекта на языке UML и элементами программного проекта.

Программный текст должен выполняться в соответствии со стандартом образовательного учреждения [2]. Тексты программных модулей в пояснительной записке не приводятся. Вместо этого программный проект записывается на компакт-диск, который прилагается к пояснительной записке.

## 2.5 Разработка тестирующего приложения

Тестирующее приложение должно показать работоспособность разработанного программного продукта, а также выявить ошибки, недочеты и недостатки.

Перед разработкой тестирующего приложения необходимо сформировать цели тестирования, такие, как определение правильности функционирования и соответствия функциональной части техническому заданию, выявление ошибок, определение поведения программного продукта в нестандартных и (или) критических ситуациях.

В случае, если заданием на курсовое проектирование является разработка классифицирующей иерархии типов в некоторой абстрактной предметной области, тестирующее приложение должно описывать несколько коллекций разных типов, в том числе абстрактных, и должно выполнять операции с коллекциями, такие, как добавление, удаление и выбор элемента коллекции.

В случае, если разрабатываемый продукт является конечным приложением, взаимодействующим с пользователем, эта часть заменяется на разработку интерфейса приложения.

### 3 Оформление курсовой работы

#### 3.1 Пояснительная записка

Пояснительная записка выполняется с помощью компьютера и распечатывается на лазерном принтере. Оформление текста должно соответствовать стандарту ОТИ МИФИ [1].

Текст располагается на одной стороне белого листа формата А4 (210×297 мм). На каждом листе должна быть выполнена рамка по ГОСТ 2.104 (размер рамки 185×287 мм, толщина линии рамки 0,8-1,2 мм). Рамка имеет отступ от левого края листа 20 мм, от верхнего края листа - 5 мм. Размер и толщина линии рамки на всех листах должны быть одинаковыми. Лист аннотации содержит основную надпись по форме 2, остальные листы, за исключением титульного, содержат основную надпись по сокращенной (неполной) форме 2а.

Текст на странице со всех сторон должен иметь отступ от рамки и от основной надписи, равный 4-5 мм.

Текст выполняется шрифтом Times New Roman, размер шрифта 14 пунктов, междустрочный интервал - полуторный. Выравнивание основ

ного текста - «по ширине». Абзацы основного текста выполняются с отступом первой строки, равным 1,5 см (с «красной» строкой).

Не допускаются выделения текста при помощи полужирного и (или) курсивного начертания букв, а также подчеркивания.

Переносы слов в документе должны быть разрешены.

Заголовки разделов, подразделов, пунктов и подпунктов также выполняются с «красной» строкой и выравниваются по левому краю. Заголовки не должны содержать переносов слов. В конце заголовка не допускаются никакие знаки препинания.

Заголовки разделов и подразделов, пунктов и подпунктов должны быть по возможности краткими и раскрывать содержание изложенного материала. Разделы и подразделы должны иметь порядковые номера, обозначенные арабскими цифрами. После каждого номера должна стоять точка, за исключением последнего номера, после которого точка не ставится.

Разделы «Обозначения и сокращения», «Введение» и «Заключение» не нумеруются.

Каждый раздел следует начинать с новой страницы.

Расстояние между заголовком раздела, подраздела, пункта, подпункта и основным текстом визуально должно составлять одну пустую строку.

Перечисления оформляются так же, как основной текст.

Пример оформления перечислений:

- пример элемента перечисления первого уровня;

а) пример элемента перечисления второго уровня, который располагается на двух строчках;

1) пример элемента перечисления третьего уровня, который располагается на двух строчках;

2) пример элемента перечисления третьего уровня;

б) пример элемента перечисления второго уровня;

- пример элемента перечисления первого уровня, который располагается на двух строчках.

Документ может содержать таблицы, рисунки и формулы. На каждую таблицу, рисунок или формулу в документе непосредственно перед таблицей, рисунком или формулой должна быть ссылка, например, «рисунок 2.1», «(таблица 3.1)», «по формуле 3.2».

Рисунок размещается по центру страницы. Под рисунком размещается подпись, состоящая из слова «Рисунок» и порядкового номера. Если рисунок имеет название, оно размещается за номером рисунка через тире. Пример оформления рисунка приведен на рисунке 3.1.



Рисунок 3.1 - Пример оформления рисунка

Знаки препинания в конце названия рисунка недопустимы. Рисунок и подпись рисунка должны располагаться на одной странице. Расстояние между текстом и рисунком, между рисунком и подписью рисунка, а также между подписью рисунка и последующим текстом визуально должно составлять одну пустую строку.

Таблица выравнивается по левому краю. Перед таблицей должна находиться подпись, состоящая из слова «Таблица» и порядкового номера таблицы. Если таблица имеет название, оно размещается за номером рисунка через тире.

Знаки препинания в конце названия таблицы недопустимы. Расстояние между текстом и подписью таблицы, а также между таблицей и последующим текстом визуально должно составлять одну пустую строку. Расстояние между подписью таблицы и таблицей визуально должно составлять половину пустой строки. Подпись таблицы и начало таблицы должны располагаться на одной странице.

Если таблица не помещается на одном листе, она разбивается на две или более частей. Перед таблицей, которая является продолжением таблицы, начало которой находится на предыдущей странице, размещается подпись «Продолжение таблицы» или «Окончание таблицы» с указанием номера.

Рамки таблицы выполняются толщиной 0,75 пункта. Между рамкой таблицы и текстом внутри таблицы должно быть расстояние не менее 2 мм. Числовые данные внутри таблиц выравниваются по правому краю так, чтобы одноименные разряды чисел находились на одной вертикали. Текст внутри таблиц выравнивается по левому краю. Если текст имеет вид предложения, размещенного на нескольких строчках, то он должен иметь отступ «красной» строки, равный 1-1,5 см.

В заголовках граф таблицы не допускаются переносы.

В текст пояснительной записки могут быть включены небольшие фрагменты кода, если они помогают раскрыть содержание. При этом фрагменты кода должны оформляться как рисунок, и текст документа должен иметь ссылку на фрагмент кода, как на рисунок. Рекомендуемый шрифт для выполнения фрагмента кода - Courier New, размер шрифта 10 пунктов, межстрочное расстояние 1,2, начертание полужирное. Каждая строка кода должна иметь отступ 1,5 см от левого края. Пример выполнения фрагмента кода приведен на рисунке 3.2.

```
/* комментарий */
int some_func() {
    /* комментарий */
    while (1) {
        char c = next_char();
        if (c == END_OF_TEXT) {
            /* комментарий */
            return 1;
        }
    }
    return 0;
}
```

Рисунок 3.2 - Пример оформления фрагмента кода

Вместо фрагментов кода предпочтительнее вставлять в текст документа алгоритмы, выполненные на виде алгоритмической записи или в виде блок-схемы.

Алгоритмы, выполненные в виде алгоритмической записи, оформляются так же, как и фрагменты кода. Блок-схемы оформляются в соответствии с ГОСТ 19.701 90. Непосредственно в тексте допускается размещать блок-схемы, целиком уместающиеся на одной странице. При этом блок-схема помечается, как рисунок, и текст документа ссылается на нее, как на рисунок. Объемные блок-схемы следует размещать в приложениях, помеченных как обязательные, если они необходимы для правильного понимания текста.

В основном тексте нецелесообразно использовать длинные цитаты. Запрещается воспроизведение фрагментов текста, фактов, данных публикаций тех или иных авторов без указания заимствованных источников. На материалы, взятые из литературы и других источников (утверждения, формулы, цитаты и т.п.) должны быть даны ссылки с указанием номера источника по списку использованной литературы. Номер ссылки проставляется арабскими цифрами в квадратных скобках.

Список использованных источников должен содержать только те источники, которые непосредственно использованы студентом и на которую имеются ссылки в тексте. Список составляется в порядке появления ссылок.

В качестве приложений к курсовой работе может быть иллюстрированный фактический материал, служащий для подтверждения тех или иных положений автора и занимающий определенный объем: схемы данных, схемы программ, схемы взаимодействия программ, схемы ресурсов системы, фрагменты листингов программ, графики, таблицы, диаграммы.

Каждое приложение начинается с нового листа.

Приложения нумеруются буквами русского алфавита в порядке появления ссылок на них в основном тексте документа. Заголовок приложения состоит из трех строк, составляющих один абзац, выравнивание текста по центру. Первая строка содержит слово «Приложение» и про

писную букву, обозначающую номер приложения, например, «Приложение А». Вторая строка содержит заключенное в круглые скобки слово «обязательное», «справочное» или «рекомендуемое». Третья строка содержит название приложения с прописной первой буквы.

Пояснительная записка представляется к защите переплетённой или сброшюрованной.

Пример оформления титульного листа приведен в приложении А на рисунке А.1. Пример оформления листа аннотации приведен в приложении Б на рисунке Б.1. Пример оформления листа содержания приведен в приложении В на рисунке В.1.

### 3.2 Компьютерная презентация

Компьютерная презентация выполняется в Microsoft PowerPoint версии не старше 2003. Она состоит из последовательности слайдов. Рекомендуемое количество слайдов 10-15.

Презентация сопровождает выступление студента и поэтому строится в соответствии с его докладом. Выступление необходимо продумать таким образом, чтобы сформировать 10-15 фрагментов длительностью 30-40 секунд (ориентировочно), на которых будет обращено внимание комиссии. Каждой такой фрагмент выступления должен быть связан соответствующим слайдом презентации.

Каждый слайд должен иметь заголовок из одной строки.

На слайдах следует размещать либо пункты, которые подчеркивают главные моменты фрагмента выступления, либо графический материал в виде рисунков, диаграмм, схем, графиков, формул и т.п. Недопустимо размещать на слайдах текст выступления. Следует помнить о том, что если все слайды презентации содержат только текстовый материал, дублирующий выступление, надобность в презентации отпадает и эффект, который от нее можно было бы получить, утрачивается.

Если слайд содержит пункты, они не должны дословно цитировать выступление. Вместо этого каждый пункт должен являться кратким содержанием предложения, которое выступающий проговаривает полностью. Как правило, пункт слайда содержит не более одной строки, а в идеале - одно слово, являющееся ключевым.

Если слайд содержит графический материал, количество отдельных элементов этого материала должно совпадать с тем, что намеревается показать выступающий. Если докладчик рассказывает о трех составляющих системы, которая отображена на слайде, рисунок при этом должен содержать три элемента, и количество подписей к ним также должно быть равно трем.

Следует помнить о том, что во время выступления докладчик стоит спиной к экрану (и лицом к комиссии). Если во время выступления необходимо показать на какой-то элемент графического материала, следует показать указкой в его сторону, не сходя при этом с места. Чтобы приемная комиссия могла понять, на какой элемент указывает докладчик, этот элемент при необходимости должен быть каким-либо образом выделен. При необходимости выделения нескольких элементов изображения следует подготовить несколько подряд идущих слайдов.

При размещении текста и иллюстративного материала следует помнить о том, что изображение на экране монитора и на экране проектора могут сильно различаться. Поэтому следует избегать мелких надписей, затрудняющих их прочтение, а также сочетания цвета плана и фона, имеющих один цветовой тон. Необходимо обеспечить достаточный контраст так, чтобы все надписи, а также мелкие элементы рисунков, графиков, диаграмм и т.п. были хорошо видны (различимы).

Не рекомендуется использовать для написания текста экзотические шрифты - их может не оказаться на том компьютере, с помощью которого презентация будет демонстрироваться.

При использовании в качестве иллюстративного материала скриншотов экранных форм необходимо помнить о том, что на формах используются мелкие шрифты. Поэтому полученные рисунки при размещении на слайдах лучше увеличить до масштаба 120-150%. А для этого, вероятно, реальные размеры форм во время получения скриншота нужно уменьшить.

На слайдах недопустимо размещать рекламные элементы изображения, такие, как логотипы.

Анимация слайдов должна быть отключена.

Первый слайд содержит тему курсовой работы, фамилию руководителя и фамилию студента.

Следующий слайд раскрывает предметную область и подводит повествование к цели курсового проектирования. Для представления целей и задач курсовой работы отводится один слайд. Далее следует разместить слайд, на котором описываются средства разработки. Оставшиеся слайды посвящаются разработке проекта. В конце могут быть приведены слайды, посвященные результатам проектирования и перспективам проекта. Не следует заключать последовательность слайдом «Спасибо за внимание» или подобным.

Во время выступления следует избегать ситуаций, когда слайд отображается на экране время, недостаточное для его полного прочтения, или отображается дольше необходимого. Время отображения слайда должно в точности соответствовать времени, необходимому для рассказа о его содержимом. Недопустимо рассказывать во время показа слайда о предметах, которые не нашли отображение на слайде, равно как и рассказывать о содержимом слишком долго.

#### 4 Защита курсовой работы

К защите допускаются курсовые работы, выполненные в установленные сроки и имеющие положительный отзыв руководителя о ходе проектирования. В связи со сложностью проекта к защите могут быть допущены курсовые проекты, выполненные частично (не имеющие части диаграмм, тестирующего приложения и т.п.). Решение о возможности допуска проекта к защите принимает руководитель проекта.

Защита (публичная защита) курсовых работ проводится на открытых заседаниях приемной комиссии, состоящей из трех-четырех преподавателей кафедры в сроки, регламентируемые учебным планом специальности и установленные кафедрой.

Организацией защиты руководит заведующий кафедрой, а в его отсутствие — помощник заведующего кафедрой или один из руководителей курсовых работ.

Защита включает в себя доклад студента на основе компьютерной презентации и демонстрацию разработанного программного продукта.

В выступлении следует сформулировать цели и задачи курсовой работы, раскрыть его структуру, показать используемые при проектировании решения. Следует уделить внимание выводам, предложениям, рекомендациям, сделанным автором на основе проведенной работы. Длительность выступления составляет 5-7 минут.

После доклада члены комиссии и присутствующие на защите лица задают студенту вопросы, связанные с проектированием. Далее студент демонстрирует работу разработанного программного продукта.

По окончании защиты члены комиссии на закрытом заседании коллективно обсуждают итоги защиты каждого проекта и оценивают ее большинством голосов по сто-балльной шкале в соответствии с критериями оценки курсовой работы, изложенными в 6. При равном количестве голосов приоритетное право решения предоставляется руководителю проекта, а в его отсутствие — заведующему кафедрой или председателю комиссии.

Итоги обсуждения объявляются открыто.

В тех случаях, когда защита курсовой работы признается комиссией неудовлетворительной, студент может представить к повторной защите тот же проект с доработкой, определяемой комиссией, или же обязан разработать новую тему, которая устанавливается кафедрой. Сроки и условия защиты проекта в этом случае устанавливаются кафедрой по согласованию с заместителем директора по учебной работе.

## 5 Обязательные требования к курсовой работе

5.1. Использование CASE-средств для проектирования UML диаграмм иерархии классов, диаграмм использования (UseCase), диаграмм взаимодействия, таких, как Rational Rose, IntelliJIDEa, NetBeans и т.п.

5.2. Использование полностью объектно-ориентированных сред разработки объектно-ориентированных систем.

Таковыми средами являются, например, Microsoft Visual Studio C#, IntelliJIDEa, NetBeans, Eclipse, Delphi и другие.

5.3. Программный продукт будет удовлетворять принципам объектно-ориентированной парадигмы программирования при выполнении следующих условий:

- базовыми элементами программного продукта являются объекты, а не алгоритмы:

- каждый объект является экземпляром определенного класса;
- классы используют инкапсуляцию;
- классы образуют иерархическую структуру;
- классы создают условия возникновения полиморфизма.

## 6 Критерии оценки курсовой работы

Курсовая работа оценивается по сто-балльной шкале с выставлением оценок двух видов.

Первый вид оценки формируется словами «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно». Второй вид оценки выставляется по системе ECTS в виде букв A, B, C, D, E или F. Перевод оценки из сто-балльной шкалы в оценки первого и второго вида производится в соответствии с таблицей 6.1.

Таблица 6.1

Сумма баллов	Оценка	Оценка ECTS
90-100	отлично	A
85-89	хорошо	B
75-84		C
70-74		D
65-69	удовлетворительно	D
60-64		E
Ниже 60	неудовлетворительно	F

Критериями, влияющими на результирующую оценку, являются:

- 6.1) полнота реализации требований к программе;
- 6.2) тщательность тестирования программных модулей;
- 6.3) удобство программного интерфейса;
- 6.4) стиль написания программного кода;
- 6.5) качество оформления пояснительной записки;
- 6.6) полнота и правильность ответов на вопросы;

6.7) соблюдение календарного плана выполнения работы.

Оценка «отлично» выставляется за работу, выполненную в установленные сроки, полностью отвечающую требованиям 5.1-5.3 и критериям оценки 6.1-6.6.

Оценка «хорошо» выставляется за работу, выполненную в установленные сроки, в случае, если не выполняется максимум одно или два из требований 5.1 или 5.2, или качество программного продукта не удовлетворяет критериям оценки 6.1-6.6. Эта оценка уточняется в зависимости от качества работы в соответствии с критериями 6.1-6.6.

Оценка «удовлетворительно» выставляется в случае, если программный продукт не удовлетворяет требованиям 5.3 и критерию 6.1, или если по сумме всех требований и критериев выполненная работа не может быть оценена как хорошая. Эта оценка уточняется в зависимости от качества работы в соответствии с критериями 6.1-6.6.

Оценка «неудовлетворительно» выставляется в случае, если сроки выполнения этапов не соблюдены, работа выполнена частично, объектно-ориентированная парадигма в части требований 5.3 реализована неправильно, неточно или не реализована.

## 7 Примерные темы курсовых работ

### 7.1. Примеры классифицирующих иерархий типов.

Требования к классифицирующим иерархиям типов:

- количество классов сущностей предметной области не менее 50;
- между классами существуют отношения обобщения и реализации;
- классы сущностей обладают поведением.

Студент может выбрать любую из примерных тем, а также предложить свою предметную область, если она удовлетворяет требованиям.

7.1.1. Иерархия типов: машиностроительные механизмы.

7.1.2. Иерархия типов: строительные машины и механизмы.

7.1.3. Иерархия типов: строительный инструмент.

7.1.4. Иерархия типов: музыкальные инструменты.

7.1.5. Иерархия типов: минералы.

7.1.6. Иерархия типов: бабочки.

7.1.7. Иерархия типов: насекомые.

7.1.7. Иерархия типов: млекопитающие.

7.1.9. Иерархия типов: измерительные приборы.

7.1.10. Иерархия типов: лекарственные растения.

7.2. Примеры иерархий типов реальной предметной области.

Требования к иерархии типов:

- количество классов сущностей предметной области не менее 10;
- между классами существуют отношения обобщения и реализации;

7.2.1. Иерархия типов: АСУ «Библиотека».

7.2.2. Иерархия типов: АСУ «Складское хозяйство».

7.2.3. Иерархия типов: АСУ «Почтовая служба».

7.2.4. Иерархия типов: АСУ «Такси».

7.2.5. Иерархия типов: АСУ «Гостиница».

7.2.6. Иерархия типов: АСУ «Заказ товаров».

7.2.7. Иерархия типов: АСУ «Туристическое агентство».

7.2.8. Иерархия типов: АСУ «Портфолио студента».

7.2.9. Иерархия типов: АСУ «Успеваемость».

7.2.10. Иерархия типов: Банкомат.

7.3. Примеры программ различного назначения, реализующих объектно-ориентированную парадигму программирования.

7.3.1. Стратегическая или ролевая игровая программа.

7.3.2. Программа формирования и отображения 3D-объектов.

7.3.3. Программа, моделирующая физические, химические, иные процессы и (или) явления.

## Библиография

### Список использованных источников

1. Комаров А.А., Ларьков Н.С., Нуржанова И.А., Пономарев В.В., Сосюрко В.Г. Оформление текстов учебных студенческих работ (общие требования). Методические указания — Озерск: ОТИ МИФИ, 2007. — 44 с.

2. Вл. Пономарев. ТПМ. Требования к программным модулям. Методические указания. Озерск: ОТИ МИФИ, 2006. — 68 с.

### Рекомендуемая литература

3. Пышкин Е.В. Основные концепции и механизмы объектно-ориентированного программирования. — СПб.: БХВ-Петербург, 2005. — 640 с.: ил.

4. Васильев А.Н. Объектно-ориентированное программирование: Учебное пособие. — СПб: Питер: 2014. — 400 с. — (Серия «Учебное пособие»).

5. Лафоре Р. Объектно-ориентированное программирование в C++/ Р. Лафоре. — 4-е изд. — СПб; М; Нижний Новгород [и др.]: Питер, 2013. — 923 с. — (Классика Computer Science).

6. Пономарев В.В. Объектно-ориентированное программирование на C++. Учебно-методическое пособие. Озерск: ОТИ НИЯУ МИФИ, 2014. — 60 с., ил.

### Рекомендуемые источники в сети интернет

7. (<http://www.intuit.ru/studies/courses/71/71/info>). Национальный открытый университет «ИНТУИТ», курс «Основы объектно-ориентированного программирования». Проверено 06.06.2015.

8. (<http://www.intuit.ru/studies/courses/72/72/info>). Национальный открытый университет «ИНТУИТ», курс «Основы объектно-ориентированного проектирования». Проверено 06.06.2015.

---

Пономарев Владимир Вадимович  
Программирование (Объектно-ориентированное программирование)  
Методические указания по курсовому проектированию  
Озерск: ОТИ НИЯУ МИФИ, 2014

Приложение А  
(справочное)  
Титульный лист

<p>МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «Национальный исследовательский ядерный университет «МИФИ» Озерский технологический институт - филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ» (ОТИ НИЯУ МИФИ)</p>	
<p><i>Кафедра прикладной математики</i></p>	
<p>Допускается к защите</p>	
<p>Зав. кафедрой</p>	
<p>к.ф.-м.н. _____ Р.Р. Акопян</p>	
<p>« ____ » _____ 20 ____ г.</p>	
<h2>КУРСОВАЯ РАБОТА</h2>	
по дисциплине:	<u>Объектно-ориентированное программирование</u>
Тема работы:	<u>Иерархия типов АСУ почтовой службы</u>
<p>Пояснительная записка МИФИ.091211.101ПЗ</p>	
Выполнил студент гр. 1ПО-ХХД	_____ И.С. Петров
Руководитель	_____ Е.Л. Шершнев
Рецензент	_____ А.Ф. Зубаиров
Н. контролер	_____ В.В. Пономарев
<p>г. Озерск, 2015 г.</p>	

Рисунок А.1 - Титульный лист

Приложение Б  
(справочное)  
Лист аннотации

<p><b>А н н о т а ц и я</b></p> <p>И.С. Петров. Иерархия типов АСУ почтовой службы: Курсовая работа. ОТИ НИЯУ МИФИ, 2015, 34 стр., 12 ил. Библиография - 7 наименований. Вложения - CD-диск с демо-версией.</p> <p>В курсовом проекте разработана иерархия типов АСУ почтовой службы. Иерархия типов включает в себя:</p> <ul style="list-style-type: none"> <li>- классов всего: 39;</li> <li>- классов сущностей: 20;</li> <li>- управляющих классов: 9;</li> <li>- интерфейсных классов: 10</li> </ul> <p>Иерархия типов образует один библиотечный модуль на языке С#.</p> <p>Разработанное тестовое приложение позволяет проверить правильность функционирования классов библиотечного модуля.</p>									
<i>МИФИ.091211.101ПЗ</i>									
Изм.	Лист	На докум.	Подпись	Дата	Иерархия типов АСУ почтовой службы	К		Лист	Листов
	Петров			24.05.15		5		34	
	Шершнев					<i>ОТИ 1ПО-ХХД</i>			
	Пономарев								
	Утв.	Акопян							

Рисунок Б.1 - Лист аннотации

Приложение В  
(справочное)  
Лист содержания

Содержание				
Обозначения и сокращения.....				7
Введение.....				8
1 Предметная область.....				9
1.1 Анализ задачи.....				9
1.2 Требования к проекту.....				10
2 Проектная часть.....				11
2.1 Средства проектирования.....				11
2.2 Абстрагирование и выделение объектов.....				11
2.3 Моделирование иерархии классов.....				12
2.4 Построение диаграмм использования.....				14
2.5 Построение диаграмм взаимодействия.....				16
3 Практическая часть.....				18
3.1 Структура программного проекта.....				18
3.2 Проектирование классов сущностей.....				19
3.3 Проектирование управляющих классов.....				21
3.4 Проектирование интерфейсных классов.....				23
4 Экспериментальная часть.....				25
4.1 Цели тестирования.....				25
4.2 Структура тестирующей программы.....				25
4.3 Тесты.....				26
Заключение.....				29
Список использованных источников.....				30
Приложение А (обязательное) Иерархия классов.....				31
				Лист
				7
Изм.	Лист	№ докум.	Подпись	Дата
			МИФИ.091211.101ПЗ	

Рисунок В.1 - Лист содержания