

Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

Томский государственный университет систем управления  
и радиоэлектроники (ТУСУР)

Кафедра автоматизированных систем управления

**РОМАНЕНКО В.В.**

## **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

Методические указания по самостоятельной работе студентов

Уровень основной образовательной программы – бакалавриат

Специальность – Информатика и вычислительная техника (09.03.01)

Профиль – Программное обеспечение средств вычислительной техники и  
автоматизированных систем

Томск – 2018

**Романенко В.В.**

**Объектно-ориентированное программирование.** Методические указания по самостоятельной работе студентов по специальности 09.03.01 – «Информатика и вычислительная техника» (профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем») / В.В. Романенко. – Томск: ТУСУР, 2018. – 9 с.

Методические указания разработаны в соответствии с решением кафедры автоматизированных систем управления.

Составитель: к.т.н., доцент Романенко В.В.

Методические указания утверждены на заседании кафедры автоматизированных систем управления 28 августа 2013 г., протокол №1.

© ТУСУР, кафедра АСУ, 2018

© Романенко В.В., 2018

**СОДЕРЖАНИЕ**

1. Общие рекомендации	4
2. Требования к результатам освоения дисциплины	4
3. Содержание дисциплины	5
3.1 Теоретический материал	5
3.2 Лабораторные занятия	6
4. Темы для самостоятельного изучения	7
5. Методические рекомендации по самостоятельному изучению материала	7
6. Формы контроля	7
7. Интерактивные формы	7
8. Учебно-методическое и информационное обеспечение дисциплины	8

## 1. ОБЩИЕ РЕКОМЕНДАЦИИ

**Целью** курса является обучение студентов основам объектно-ориентированного проектирования и программирования в современных средах разработки ПО.

Основой **задачей** изучения курса является получение знаний и практических навыков в области проектирования и разработки объектно-ориентированных программ. В результате изучения курса студент должен иметь представление о предпосылках возникновения ООП и его месте в эволюции парадигм программирования, знать принципы объектно-ориентированного проектирования и программирования, а также уметь разрабатывать объектно-ориентированные программы на языках C++, C++ CLI и C# с применением библиотек классов STL и .NET.

Дисциплина «Объектно-ориентированное программирование» относится к вариативной части профессионального цикла дисциплин. Изучение объектно-ориентированного программирования необходимо для освоения на следующих курсах таких дисциплин, как «Операционные системы», «Теория языков программирования и методы трансляции», «Теория вычислительных процессов» и многих других дисциплин, связанных с разработкой и написанием программ, т.к. все современные языки программирования и программные библиотеки являются объектно-ориентированными.

## 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате изучения дисциплины студент должен:

***Знать:***

Принципы объектно-ориентированного анализа и проектирования, основы объектно-ориентированного подхода к программированию.

***Уметь:***

Проводить объектную декомпозицию предметной области, писать в современных средах разработки объектно-ориентированные программы, использовать объектно-ориентированные библиотеки классов при разработке ПО.

***Владеть:***

Основными приемами объектно-ориентированного программирования на языках C++, C++ CLI и C#, навыками использования библиотек классов STL, .NET.

### **3. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

#### **3.1 Теоретический материал**

##### **Тема 1. Введение**

Введение в предмет. История развития парадигм программирования. Причины возникновения ООП.

##### **Тема 2. Основные понятия**

Принципы ООП. Понятия объекта и класса. Понятия члена класса, поля, метода. Инкапсуляция, полиморфизм, наследование.

##### **Тема 3. Объектно-ориентированный анализ и проектирование**

Типы отношений между классами. Понятие предметной области решаемой задачи. Анализ предметной области. Программные средства для описания предметной области.

##### **Тема 4. Объектная декомпозиция**

Выделение в предметной области объектов и определение отношений между ними. Программные средства для описания объектной декомпозиции.

##### **Тема 5. Объектно-ориентированное программирование на языке C++**

Особенности объектной реализации в языке C++. Переход от языка C к языку C++. Программирование на смешанных объектно-ориентированных языках. Указатели на функции и процедуры. Написание динамических библиотек (DLL).

##### **Тема 6. Классы и структуры. Члены классов. Дружественность**

Синтаксис описания класса. Отличия классов и структур в языке C++. Члены класса. Конструкторы и деструкторы. Поля. Методы. Объявление вложенных типов. Объявление дружественности. Статические поля и методы. Указатели на методы классов.

##### **Тема 7. Перегрузка стандартных операторов**

Правила перегрузки операций в языке C++. Перегрузка унарных и бинарных операций. Операторы-члены класса и внешние операторы. Перегрузка операторов приведения типа.

##### **Тема 8. Виртуальный и абстрактный полиморфизм. Наследование**

Таблицы виртуальных функций. Виртуальные и абстрактные методы. Абстрактные классы. Статический и виртуальный полиморфизм. Наследование. Поведение классов при наследовании.

##### **Тема 9. Шаблоны функций и классов**

Шаблоны процедур и функций. Шаблоны структур и классов. Шаблоны констант и типов.

**Тема 10.** Объектно-ориентированное программирование на языке C#  
 Основы программирования на языке C#. Безопасность кода. Управляемые ресурсы. Типы данных по значению и ссылочные типы данных. Полностью объектно-ориентированные языки программирования. Язык C++ CLI.

**Тема 11.** Библиотека .NET. Основы языка C#  
 Операторы языка C#. Операторы выражений. Идентификаторы. Форматирование и разбор строк. Консольный ввод-вывод. Файловый ввод-вывод. Сериализация и десериализация.

**Тема 12.** Классы, структуры и интерфейсы  
 Пространства имен. Сборки. Синтаксис описания классов, структур и интерфейсов. Наследование и полиморфизм в языке C#. Вложенные типы.

**Тема 13.** Свойства и индексы  
 Поля класса. Синтаксис описания свойств и индексов в классах и интерфейсах. Статические поля.

**Тема 14.** Делегаты. События  
 Методы класса. Указатели на методы классов. Синтаксис описания делегатов и событий. Статические и анонимные методы.

**Тема 15.** Универсальные типы  
 Параметры типа. Ограничения параметров типа. Универсальные методы, классы и интерфейсы. Наследование универсальных типов.

**Тема 16.** Документирование кода  
 Расстановка в коде тегов документирования. Сборка XML-документации. Генерация файлов документации. Разработка проектной документации.

### **3.2. Лабораторные занятия**

#### **Темы лабораторных работ:**

**Тема 1.** Анализ предметной области и проектирование класса на языке C++.

**Тема 2.** Инкапсуляция объектов линейной алгебры в классе, и перегрузка стандартных операций для них на языке C++.

**Тема 3.** Наследование на примере создания иерархии объектов линейной алгебры на языке C++.

**Тема 4.** Инкапсуляция математических объектов в классе, и перегрузка стандартных операций для них на языке C++.

**Тема 5.** Создание классов-шаблонов на языке C++.

**Тема 6.** Инкапсуляция объектов линейной алгебры в классе, и перегрузка стандартных операций для них на языке C#.

**Тема 7.** Наследование на примере создания иерархии объектов линейной алгебры на языке C#.

**Тема 8.** Создание универсальных классов на языке C# и документирование кода проекта.

#### **4. ТЕМЫ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ**

1. Изучение библиотеки STL.

2. Основы консольного и файлового ввода-вывода на языке C#.

Сериализация.

#### **5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ МАТЕРИАЛА**

Указанные в п. 4 темы изучаются студентами самостоятельно. Литература по этим темам дана в п. 8.1, 8.2. Контроль знаний по этим темам осуществляется на лекционных занятиях, а также на экзамене.

Темы лабораторных занятий студенты изучают следующим образом. Перед лабораторной работой студенты самостоятельно повторяют теоретический материал. Далее проводят объектную декомпозицию предметной области, разрабатывают проектную документацию в формате СНМ или HTML. Затем пишут программы и защищают их.

#### **6. ФОРМЫ КОНТРОЛЯ**

Формой контроля являются тесты, конспекты лекционного материала и самостоятельной работы, защита лабораторных работ и экзамен.

#### **7. ИНТЕРАКТИВНЫЕ ФОРМЫ**

Важным элементом успешного освоения материала являются интерактивные формы проведения занятий. По данной дисциплине предполагаются следующие интерактивные формы: IT-методы, работа в команде, игра:

1) Интерактивные IT-методы используются в демонстрации на лекции (с использованием проектора) процесса компиляции и работы приложений в Microsoft Visual Studio, демонстрирующих различные аспекты ООП (виртуальный полиморфизм, перегрузка операторов и т.д.). При этом студенты могут вносить свои предложения по модификации кода.

2) Работу в команде студенты используют при выполнении лабораторной работы №8, в которой выполняется документирование кода

класса. При этом они совместно разрабатывают один проект (координатор команды разрабатывает ядро программы, остальные участники – дополнительные модули в виде DLL-сборок). Каждый участник генерирует XML-документацию для своей части проекта.

3) Обучение в виде игры используется при демонстрации работы динамических структур данных. При этом каждый студент является одним из элементов такой структуры, связанный с одним или несколькими другими элементами. В таком формате демонстрируется работа с динамическими списками и деревьями, цепочками вызова полиморфных функций и т.п.

## **8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **8.1. Основная литература**

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня: учебник для вузов. – СПб: Питер, 2013. – 461 с. (35 экз.).

2. Объектно-ориентированное программирование: Учебное пособие / Романенко В.В. 2016. 475 с. [Электронный ресурс] – Режим доступа: <https://edu.tusur.ru/publications/6300>, дата обращения: 21.06.2018.

### **8.2. Дополнительная литература**

1. Павловская Т.А. С#. Программирование на языке высокого уровня: учебник для вузов. – СПб: Питер, 2013 – 432 с. (16 экз.).

2. Орлов С.А. Технологии разработки программного обеспечения: современный курс по программной инженерии. – СПб: Питер, 2012. – 608 с. (15 экз.).

### **8.3. Учебно-методические пособия и требуемое программное обеспечение**

Учебно-методическое пособие по выполнению лабораторных работ:

1. Романенко В.В. Объектно-ориентированное программирование: Методические указания по выполнению лабораторных работ для студентов специальности 09.03.01 «Информатика и вычислительная техника» [Электронный ресурс] / В. В. Романенко. – Томск: ТУСУР, 2018. – 44 с. – Режим доступа: <https://edu.tusur.ru/publications/8018>, дата обращения: 21.06.2018.

Программное обеспечение для выполнения лабораторных работ:

1. Среда разработки Code::Blocks (распространяется бесплатно, установлена в компьютерных классах).

2. Среда разработки Microsoft Visual Studio 2013 Professional (установлена в компьютерных классах).

4. Файловый менеджер Far Manager (распространяется бесплатно, установлен в компьютерных классах).

5. Текстовый редактор Notepad++ (распространяется бесплатно, установлен в компьютерных классах).

6. Среда разработки Microsoft Visual C++ 2017 Community (распространяется бесплатно).

7. Среда разработки Microsoft Visual C# 2017 Community (распространяется бесплатно).

Программное обеспечение для оформления отчетов:

1. Офисный пакет LibreOffice (распространяется бесплатно, установлен в компьютерных классах).

2. Офисный пакет Microsoft Office 2003 (установлен в компьютерных классах).

#### **8.4. Базы данных, информационно-справочные и поисковые системы**

1. Библиотека Microsoft Developer Network (<http://msdn.microsoft.com/ru-ru>).

2. Виртуальная академия Microsoft Virtual Academy (<https://mva.microsoft.com/?lang=ru-ru>).

3. Интернет-университет ИНТУИТ (<http://www.intuit.ru/>).