

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования

Томский государственный университет систем управления и  
радиоэлектроники

Панов Сергей Аркадьевич

## **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

Методические указания по самостоятельной работе

2015

Панов С.А. Объектно-ориентированное программирование: Методические указания по самостоятельной работе / Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Томский государственный университет систем управления и радиоэлектроники. Кафедра моделирования и системного анализа. – Томск: ТУСУР, 2015. – 13 с.

## СОДЕРЖАНИЕ

Введение.....	4
Раздел 1. Структурное программирование на языке C++.....	6
Раздел 2. Модульное программирование .....	9
Раздел 3. Объектно-ориентированное программирование .....	11
Литература .....	13

## Введение

Целью курса «Объектно-ориентированное программирование» является изложение основных идей и принципов объектно-ориентированного программирования, а также ознакомление с современными технологиями программирования и привитие навыков программирования на объектно-ориентированном языке программирования C++.

В задачи курса входят:

- изучение теоретических основ и современных информационных технологий анализа, проектирования и разработки программного обеспечения;
- проектирование и разработка различных видов программного обеспечения на основе объектно-ориентированного подхода;
- представление о библиотеках классов и инструментальных средствах, применяемых при разработке программного обеспечения.

В результате изучения курса «Объектно-ориентированное программирование» студент должен:

**Знать:** основы объектно-ориентированного программирования на языке программирования высокого уровня C++; особенности создания классов и их использования в разрабатываемых приложениях, основы проектирования объектно-ориентированного программного обеспечения с использованием C++;

**Уметь:** разрабатывать программное обеспечение средствами объектно-ориентированного языка C++ и использовать его на практике;

**Владеть:** навыками программирования на языке C++.

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- способность к саморазвитию, повышению своей квалификации и мастерства (ОК-6);

– способность критически оценивать свои достоинства и недостатки, наметить пути и выбрать средства развития достоинств и устранения недостатков (ОК-7);

– способность применять методы, способы и средства получения, хранения, переработки информации и использовать компьютер как средство управления информацией (ОК-12);

– способность создавать программные комплексы для системного анализа и синтеза сложных систем (ПК-13);

– способность разрабатывать компоненты сложных систем управления, применять для разработки современные инструментальные средства и технологии программирования на основе профессиональной подготовки (ПК-14).

## Раздел 1. Структурное программирование на языке C++

### 1.1. *Содержание раздела*

Структура программы; операции над данными; выражения; базовые конструкции структурного программирования: условный оператор, оператор выбора, операторы цикла, операторы передачи управления; указатели и массивы: указатели на данные, функции и *void*, инициализация указателей, операции с указателями, одномерные массивы, динамические массивы, многомерные массивы, строки; типы данных, определяемые пользователем: переименование типов, перечисления, структуры, битовые поля, объединения.

### 1.2. *Методические указания по изучению раздела*

При изучении раздела следует обратить внимание на формирование структуры программы, принципы использования и приоритет операций, правила формирования выражений, правильность использования базовых конструкций структурного программирования, правильность определения, инициализации, использования указателей и массивов, а также освобождение занятой под них памяти, возможность ввода и использования новых типов данных.

### 1.3. *Вопросы для самопроверки*

1. Структура программы;
2. Применение операций инкремента и декремента;
3. Операция определения размера для типа данных и переменных;
4. Операции отрицания;
5. Применение операций целочисленного деления и остатка от деления;
6. Операции сдвига;

7. Операции отношения;
8. Поразрядные операции;
9. Логические операции;
10. Операция присваивания и условная операция;
11. Правила формирования выражений;
12. Базовые конструкции структурного программирования:  
следование, цикл, ветвление;
13. Операция «Выражение»;
14. Условный оператор;
15. Оператор выбора;
16. Цикл с предусловием;
17. Цикл с постусловием;
18. Цикл с параметром;
19. Оператор безусловного перехода;
20. Оператор выхода из цикла и оператора выбора;
21. Оператор продолжения;
22. Оператор возврата из функции;
23. Указатели на функции, на объект, на void;
24. Способы инициализации указателей;
25. Операции с указателями;
26. Одномерные статические массивы;
27. Динамические массивы;
28. Многомерные массивы;
29. Строки;
30. Переименование типов;
31. Перечисления;
32. Структуры;
33. Списки;
34. Массивы структур;

35. Битовые поля;
36. Объединения.

## Раздел 2. Модульное программирование

### 2.1 Содержание раздела

Функции: объявление и определение функции; глобальные переменные; возвращаемое значение; параметры функции; рекурсивные функции; перегрузка функций; шаблоны функций; главная функция; директивы препроцессора; предопределенные макросы; области действия идентификаторов; внешние объявления; поименованные области;

### 2.2 Методические указания по изучению раздела

При изучении данного раздела следует обратить внимание на правила выделения функций из общей структуры программы, их определение и объявление, правила выделения атрибутов и возвращения значений, а также на использование директив препроцессора.

### 2.3. Вопросы для самопроверки

1. Что такое функция?
2. Что называется прототипом функции?
3. Какие области содержит определение функции?
4. По какому признаку производится классификация переменных на глобальные и локальные?
5. Как правильно возвращать значения на локальные переменные?
6. Какие параметры называются формальными по отношению к функции?
7. Что называется фактическими параметрами?
8. Как осуществляется передача по адресу?
9. Как осуществляется передача по ссылке?

10. Как правильно передать массив в качестве параметра функции?
11. Принципы вызова функций с переменным числом параметров?
12. Правила применения рекурсивных функций.
13. Каковы правила и назначения перегружаемых функций?
14. Правила формирования и использования шаблонов функций.
15. Примеры использования функции main с параметрами.
16. Правила реализации многомодульных проектов с помощью директивы include?
17. Назначение директивы define?
18. Принципы использования директив условной компиляции
19. Перечислите и приведите назначения каждой области действия идентификаторов: блок, файл, функция, прототип функции, поименованная область.
20. Приведите правила внешнего объявления.
21. Назначение и правила использования поименованных областей

## Раздел 3. Объектно-ориентированное программирование

### *3.1 Содержание раздела*

Основные принципы объектно-ориентированного программирования:

наследование, инкапсуляция, полиморфизм; классы: описание класса, описание объектов, указатель this, конструкторы, статические элементы класса, дружественные функции и классы, деструкторы, перегрузка операций; наследование: ключи доступа, простое наследование, виртуальные методы, механизм позднего связывания, абстрактные классы; шаблоны классов: обработка исключительных ситуаций: общий механизм обработки исключений, синтаксис исключений, перехват исключений, список исключений функции, исключения в конструкторах и деструкторах, иерархии исключений.

### *3.2. Методические указания при изучении раздела*

При изучении раздела важным является понимание механизмов реализации в языке программирования C++ основных положений объектно-ориентированного программирования: наследования, полиморфизма и инкапсуляции. Основное внимание также должно быть уделено возможности классификации исследуемой предметной области и правильному построению и реализации иерархии классов.

### *3.3. Вопросы для самопроверки*

1. Три способа повышения абстракции программы;
2. Назначение наследования, полиморфизма и инкапсуляции;
3. Класс – как тип данных;
4. Правила описания класса;
5. Поля и методы класса;

6. Правила описания объектов класса и указателей на них;
7. Назначение указателя `this`;
8. Назначение и правила определения конструкторов класса;
9. Для чего применяются конструкторы копирования?
10. Назначение статических полей класса;
11. Правила реализации статических методов класса;
12. Для чего применяются дружественные функции и классы?
13. Что такое деструктор и для чего он реализуется в классе?
14. Приведите правила применения перегрузки операций;
15. Правила перегрузки унарных операций;
16. Правила перегрузки бинарных операций;
17. В чем суть применение механизма наследования классов?
18. Правила применения ключей доступа;
19. Что называется прямым наследованием?
20. Для чего применяются виртуальные методы?
21. В чем суть механизма позднего связывания?
22. Приведите назначение и правила применения абстрактных классов;
23. Назначение шаблонов классов;
24. Правила создания и применения шаблонов классов;
25. Что называется исключительной ситуацией или исключением?
26. В чем заключается общий механизм обработки исключений?
27. Приведите синтаксис исключения;
28. Правила перехвата исключений;
29. Как создается список исключений функции?
30. Как используются исключения в конструкторах и деструкторах?
31. Как создается иерархия исключений?

## Литература

1. Ганджа Т.В., Панов С.А. Объектно-ориентированное программирование: Курс лекций по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс]. – Режим доступа: [http://vkiem.ru/to\\_student](http://vkiem.ru/to_student) свободный.
2. C/C++. Программирование на языке высокого уровня / Т.А. Павловская. - СПб.: Питер, 2010. - 460 с. (30 экз.).
3. Язык Си ++ / В. В. Подбельский. - 5-е изд. - М.: Финансы и статистика, 2006. – 559 с. (20 экз.).
4. Шелупанов, Александр Александрович. Информатика. Базовый курс [Текст]: учебник / А. А. Шелупанов, В. Н. Кирнос; Министерство образования и науки Российской Федерации, Томский государственный университет систем управления и радиоэлектроники, Кафедра комплексной информационной безопасности электронно-вычислительных систем. - Томск: В-Спектр, 2010. - 380 с. (1 экз.).
5. Культин, Н. Б. Microsoft© Visual C++ в задачах и примерах: / Н. Б. Культин. - СПб.: БХВ-Петербург, 2010. – 264 с. (1 экз.).
6. Алексеев Е. Р. MS Visual C ++ и Turbo C++ Explorer: / Е. Р. Алексеев; ред. О. В. Чеснокова. - М.: НТ Пресс, 2007. – 351 с. (1 экз.).