

**ПРОГРАММА  
ДИСЦИПЛИНЫ  
“ПРОГРАММИРОВАНИЕ И  
ОСНОВЫ  
АЛГОРИТМИЗАЦИИ”**

**1. Цели изучения дисциплины**

*Знать основные конструкции языков C/C++, выбранного в качестве базового, современные требования к изобразительным средствам языка программирования в свете достижений технологий программирования, основные подходы к программированию, знать и уметь использовать технологии структурного, модульного и объектно-ориентированного программирования, инструментальные средства разработки программ, приобрести практические навыки алгоритмизации и программирования задач из различных проблемных областей, образующих фундамент науки программирования. Характерной особенностью этой дисциплины является рассмотрение технологии объектно-ориентированного программирования применительно к решению классических задач прикладного программирования.*

*В целях экономии учебного времени многие вопросы базового языка программирования изучаются студентами самостоятельно.*

**2. Место дисциплины в учебном плане**

*Дисциплина изучается в 1 и 2 семестрах и является базовой для всех последующих дисциплин и, в первую очередь, для курса "Технологии программирования", изучаемого в следующих семестрах.*

**3. Объем дисциплины по видам учебной работы и формы контроля**

Виды занятий и формы контроля	Объем по семестрам	
	1 семестр	2 семестр
Лекции, ч./нед.	<b>3</b>	<b>2</b>
Лабораторные занятия, ч./нед.	<b>1</b>	<b>1</b>
Самостоятельные занятия, ч./нед.	<b>1</b>	<b>1</b>
Экзамены, шт./сем.	<b>1</b>	<b>1</b>
Курсовые работы, шт./сем.	<b>1</b>	<b>1</b>

Общая трудоемкость дисциплины составляет 144 часа, в том числе 32 часа самостоятельных занятий.

**4. Содержание дисциплины**

**4.1. Разделы дисциплины и виды занятий**

Раздел программы	Объемы занятий, часов			
	ЛК	ПЗ	Лб	Сам
1. Введение. Системы счисления. Классификация языков программирования.				1
2. Язык программирования высокого уровня C++.				1

Алгоритм и способы его записи. Структурное и модульное программирование. Описание языка. Структура и конструкция программы.				
3. Типы данных и их атрибуты. Имена, типы. Ввод-вывод языка С.	4		2	
4. Класс хранения данного: область действия и время жизни. Внешние и внешние статические данные. Функции. Автоматические, регистровые и внутренние статические данные. Инициализация данных. Массивы и структуры.	4		3	
5. Операторы и управление их исполнением.	1		2	
6. Выражения и операции.	1			
7. Указатели.	3		1	
8. Поля битов и побитовые операции.	1			
9. Динамическое размещение объектов в памяти.	1			
10. Однонаправленный линейный список и операции с ним.	2		1	
11. Препроцессор.	1			
12. Объявление типа. Объединения и перечисления.	1			
13. Новые возможности языка С++, не связанные с ООП.	2		1	
14. Технология создания программ. Динамические структуры данных.				
15. Прикладное программирование. Сортировка массивов.	4			2
16. Графы. Транспортная задача.	3			
17. Поиск в таблице.	2			
18. ООП. Инкапсуляция.	9			
19. ООП. Наследование	6			
20. ООП. Полиморфизм	3			
<b>Курсовая работа (семестр 1)</b>			5	12
21. Области действия и пространство имен.	3			
22. Ввод-вывод в языке С++.	5			
23. Обработка исключительных ситуаций.	3			
24. Преобразование типов в языке С++.	3			
25. ООП: сортировка массивов.	3			
26. ООП: транспортная задача.	3			
27. ООП: поиск в таблице.	3			
28. ООП: списки, очереди и стеки.	3			
29. ООП: сортировка файлов.	3			
30. Стандартная библиотека языка С++. Строки.	3			
<b>Курсовая работа (семестр 2)</b>			16	16
Итого	80		32	32

Примечание: ЛК — лекции, ПЗ — практические занятия, Лб — лабораторные занятия, Сам — самостоятельные занятия.

## 4.2 Содержание разделов дисциплины

**1. Введение.** Системы счисления. Классификация языков программирования и их краткая характеристика.

### **Часть 1. Основы базового языка программирования**

**2. Язык программирования высокого уровня C++.** Структурное и модульное программирование. Синтаксис и семантика языка: металингвистические формулы и синтаксические диаграммы. Структура и конструкция программы: комментарии, идентификаторы, служебные слова, константы, структура программы.

**3, 4. Типы данных и их атрибуты. Функции. Массивы и структуры. Имена.** Типы данных.

Простой ввод-вывод: ввод-вывод потока, ввод с использованием семейства функций `scanf()`, вывод с использованием семейства функций `printf()`.

Класс хранения: область действия и время жизни. Внешние и внутренние статические данные.

Средства поддержки иерархического проектирования программ. Основы методологии модульного программирования. Нисходящее программирование. Иерархическая декомпозиция. Требования к модулям. Функция как реализация модуля. Аргументы и параметры, передача аргументов по значению и по ссылке. Возвращаемое значение функций.

Автоматические, регистровые и внутренние статические данные. Инициализация данных.

Использование описателей класса хранения объектов для обмена данными с функциями вместо аргументов и параметров.

Производные типы данных. Массивы. Массивы как аргументы функций.

Структуры. Структуры в качестве аргументов функций.

Почему массивы и структуры следует передавать в функцию по ссылке.

**5. Операторы и управление их исполнением.** Основы методологии структурного программирования. Конструкции структурного программирования - следование, ветвление и повторение.

Пустой оператор. Операторы-выражения. Операторы `break` и `continue`. Блок операторов. Оператор `return`. Операторы ветвлений (`if`, `switch`). Циклические операторы (`for`, `while`, `do-while`). Оператор `goto` и метки операторов.

**6. Выражения и операции.** Приоритеты и ассоциативность операций. Операции ссылки. Унарные операции. Бинарные операции. Правила преобразования типов операндов для арифметических операций и операций отношений. Тернарные операции. Операции отношений. Операция запятая.

**7. Указатели.** Зачем нужны указатели? Связь указателей с массивами и строками. Определение и применение указателей. Указатели на структуры. Использование указателей в качестве аргументов функций. Массивы указателей: запуск программы с помощью командной строки. Замена типов указателей.

**8. Поля битов и побитовые операции.**

**9. Динамическое размещение объектов в памяти.** Динамическое размещение объектов в памяти средствами языков C и C++. Операции `new` и `delete`.

**10. Однонаправленный не кольцевой линейный список и операции с ним.** Понятие об однонаправленном линейном списке (ЛС), зачем нужен ЛС? Операции над ЛС.

**11. Препроцессор.** Препроцессор как инструмент проектирования программ. Директивы препроцессора. Подстановка имен. Включение файлов. Условная компиляция. Указания по работе с препроцессором.

**12. Объявление типа. Объединения и перечисления.** Объявление имени типа typedef. Объекты перечислимого типа. Объединения.

**13. Новые возможности языка C++, не связанные с объектно-ориентированным программированием (ООП).** Прототипы функций. Аргументы по умолчанию. Доступ к глобальным переменным, скрытым локальными переменными с тем же именем. Модификаторы const и volatile. Ссылки. Подставляемые функции. Операции динамического распределения памяти. Перегрузка функций. Шаблоны функций. Перегрузка операций.

## **Часть 2. Прикладное программирование с использованием технологий структурного и модульного программирования**

**14. Технология создания программ. Динамические структуры данных.** Кодирование и документирование программы. Проектирование и тестирование программы.

Динамические структуры данных — линейные списки, бинарные деревья, очереди и их разновидности. Реализация динамических структур с помощью массивов.

**15. Прикладное программирование. Сортировка массивов.** Показатели эффективности. Сортировка "на месте". Устойчивость сортировки. Сортировки выбором, включением и обменом. Простые и сложные (Шелла, бинарная, Хоора) сортировки.

**16. Графы. Транспортная задача.** Терминология. Формы задания графа. Почему подходит рекурсивный алгоритм? Представление кратчайшего пути до каждой вершины. Как найти минимальный путь? Рекурсия.

**17. Поиск в таблице.** Постановка задачи поиска. Последовательный, бинарный поиск и хэш-поиск.

**18. ООП. Инкапсуляция.** Парадигма объектно-ориентированного программирования и средства поддержки ООП в базовом языке. Класс = данные + функции для работы с ними. Инициализация и разрушение объектов. Конструкторы и деструкторы. Статические члены классов. Друзья класса. Перегрузка операций (операторов) для классов. Шаблоны классов. Конструктор копирования. Поверхностное и глубинное копирование.

## **Часть 3. Объектно-ориентированная технология программирования (ООП)**

**19. ООП. Наследование.** Иерархия наследования классов. Доступ к членам базовых классов. Виртуальные базовые классы.

**20. ООП. Полиморфизм.** Виртуальные функции-члены классов. Виртуальные деструкторы. Абстрактные функции и классы.

**21. Области действия и пространство имен.** Области действия и время жизни. Пространства имен. Пространство имен стандартной библиотеки.

**22. Ввод-вывод в языке C++. Потокные классы.** Ввод-вывод стандартных типов. Состояния потоков и их ошибки. Ввод-вывод для типов, определенных пользователем.

Форматированный ввод-вывод. Манипуляторы. Методы обмена с потоками. Файловый ввод-вывод.

**23. Обработка исключительных ситуаций.** Общий механизм обработки исключений. Синтаксис исключений. Перехват исключений. Примеры обработки исключений.

**24. Преобразование типов в языке C++.** Операция `const_cast`. Преобразование типов во время компиляции (операция `static_cast`). Преобразование типов во время выполнения программы (операция `dynamic_cast`). Преобразование "на свой страх и риск".

#### **Часть 4. Прикладное программирование с использованием технологии ООП**

**25. ООП. Сортировка массивов.** Спецификация класса для сортировки массива. Сортировка массивов с использованием шаблонных классов.

**26. ООП. Транспортная задача.** Спецификация класса для решения транспортной задачи. Решение транспортной задачи с использованием иерархии обычных классов.

**27. ООП. Поиск в таблице.** Спецификация класса для решения транспортной задачи. Поиск в таблице с использованием шаблонного класса.

**28. ООП. Списки, очереди и стеки.** Бинарные деревья. Очереди и стеки. Универсальная очередь неограниченного и ограниченного размеров. Динамический стек.

**29. ООП. Сортировка файлов.** Сортировка файлов простым слиянием. Сортировка файлов естественным слиянием (спецификация класса, шаблон классов).

#### **Часть 5. Стандартная библиотека языка C++. Обобщенное программирование**

**30. Строки.** Создание строк. Конструкторы и деструкторы строк. Операции над строками (присваивание и добавление частей строк, преобразования строк, поиск подстрок, сравнение частей строк, получение характеристик строк).

### **5. Лабораторный практикум**

В процессе лабораторного практикума осваивается интегрированная среда разработки программ и выполняются две курсовых работы – по одной в первом и втором семестрах.

#### **6. Курсовой проект (курсовая работа)**

□ В курсовой работе, выполняемой в первом семестре, по индивидуальным заданиям разрабатываются три простых программных проекта, направленных на освоение методологии структурного программирования - с линейным следованием операторов, с ветвлениями и циклическими операторами - и программный проект, направленный на освоение технологии модульного программирования. Последний программный проект обеспечивает освоение методики декомпозиции решаемой задачи, получения навыков нисходящего проектирования программ и применения средств модульного программирования.

На практических занятиях и в процессе самостоятельной работы во втором семестре студенты выполняют курсовую работу, целью которой является освоение технологии объектно-ориентированного программирования. В рамках курсовой работы проектируются базовый и производный классы или их шаблоны и выполняется их тестирование. Для ввода-вывода применяются средства языка C++, обработка ошибок ведется путем обработки исключений.

*В процессе выполнения курсовых работ студенты знакомятся со следующими аспектами проектирования программных продуктов:*

□ *Программные продукты. Основные подходы (структурное, модульное и объектно-ориентированное программирование) и их реализация средствами языка программирования высокого уровня.*

□ *Документирование. Единая система программной документации и ее характеристика.*

□ *Технологический цикл разработки программных систем. Коллективная работа по созданию программ.*

*Тестирование и отладка. Требования к контрольным (тестовым) примерам. Обоснование контрольных примеров и их анализ. Планирование отладки: выбор точек останова, анализ ожидаемых в них результатов, поэтапное (пошаговое) выполнение программ. Средства отладки интегрированных сред разработки.*

*Примерный объем и трудоемкость курсовых работ характеризуют данные, приведенные ранее в таблице в разделе 4.*

## **7. Учебно-методическое обеспечение дисциплины**

### **7.1. Рекомендуемая литература**

*Основная*

*1. Давыдов В.Г. Программирование и основы алгоритмизации: Учеб. пособие / В.Г. Давыдов. – 2-е изд., стер. – М.: Высш. шк., 2005. – 447 с.: ил.*

*2. Давыдов В.Г. Технологии программирования. С++. –СПб.: БХВ-Петербург, 2005. – 672 с.: ил.*

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

*Дополнительная*

*4. С/С++. Программирование на языке высокого уровня / Т.А. Павловская. СПб: Питер, 2001. – 464 с.: ил.*

*5. Павловская Т., Щупак Ю. С/С++. Структурное программирование: Практикум. – СПб.: Питер, 2002. – 240 с.*

*6. Павловская Т.А., Щупак Ю.А. С++. Объектно-ориентированное программирование: Практикум. – СПб.: Питер, 2005. – 265 с.: ил.*

### **7.2. Технические средства освоения дисциплины**

*Любая интегрированная среда разработки (ИСР) программ на языке С++, например, наиболее широко распространенная ИСР Microsoft Visual Studio С++ . NET 2005.*

## **8. Материально-техническое обеспечение дисциплины**

*Компьютерный класс, оснащенный персональными ЭВМ типа Pentium4 и принтером (принтерами), на 12-15 посадочных мест. При чтении лекций рекомендуется использовать компьютер с мультимедийным проектором.*

## **9. Методические рекомендации по организации изучения дисциплины**

Для организации текущего контроля, наряду с курсовыми работами и экзаменами в 1 и 2 семестрах выполняются тестовые работы по следующим темам [1, 2]:

- ввод-вывод средствами языка C (1 семестр);
- ветвления и циклы (1 семестр);
- структуры (1 семестр);
- функции (1 семестр);
- области действия и время жизни (1 семестр);
- указатели (1 семестр);
- динамическая память и работа с ЛС (1 семестр);
- препроцессор, перечисления, функции со значениями параметров по умолчанию, перегрузка функций, шаблоны функций, перегрузка операций (1 семестр);
- классы, инкапсуляция (2 семестр);
- классы, наследование и полиморфизм (2 семестр);
- пространства имен, ввод-вывод в языке C++, обработка исключений (2 семестр);
- прикладное программирование, сортировка массивов и файлов, транспортная задача, поиск в таблице, списки, очереди и стеки (2 семестр).

Утвержденно на заседании кафедры , 20.12.2012

Декан факультета

Зав. Кафедрой

Составитель

**Phan Nguyên Hải**

**Phan Nguyên Hải**