

## Программа курса «Основы программирования на C++»

**Цель курса:** познакомиться с языком программирования C++, изучить основные концепции программирования, заложить базовые навыки, необходимые для разработки программного обеспечения.

### Модуль 1. Знакомство с языком программирования C++

#### Занятие 1 Настройка рабочего окружения. «Hello world!»

- > Установка и настройка программы Visual Studio Community
- > Знакомство с ключевыми особенностями языка и областью применения
- > Знакомство с средой разработки
- > Создание первой программы «Hello world!»

**Результат занятия:** настроили рабочее окружение, познакомились с средой разработки (IDE), обсудили цели и задачи программирования.

**Практика:** создание первого приложения, настройка рабочего окружения

#### Занятие 2 Переменные и типы данных. Ввод и вывод информации

- > Знакомство с элементарными типами данных (целые, дробные числа, строка)
- > Основные операции с простыми типами
- > Ввод и вывод информации в консоль

**Результат занятия:** познакомились с элементарными типами данных, изучили основные действия с переменными, научились вводить и выводить данные из программы

**Практика:** создание приложения «Диалог»

#### Занятие 3 Условия и конструкции ветвления

- > Знакомство с базовыми логическими операциями
- > Изучения области применения конструкций ветвления
- > Изучение типичных конструкций ветвления

**Результат занятия:** изучили базовые логические операции, научились применять условные конструкции в решении алгоритмических задач.

**Практика:** создание приложения «Как мне одеться?»

#### Занятие 4 Циклические конструкции

- > Знакомство с основными видами циклов (for, while и do...while)
- > Создание приложения с использованием циклических конструкций

**Результат занятия:** изучили циклы и область их применения, создали приложение с использованием циклов

**Практика:** создание приложения для решения алгоритмических задач

## **Модуль 2. Динамическое программирование**

### **Занятие 5    Массивы**

- > Изучение структуры массивов
- > Освоение правил работы с массивами
- > Расположение массива в памяти компьютера
- > Изучение простейших алгоритмов сортировки

**Результат занятия:** изучили устройство массивов и правила работы с ними, создали приложение с использованием массивов, изучили «Сортировку пузырьком»

**Практика:** создание приложения для сортировки массива целых чисел

### **Занятие 6    Функции**

- > Зачем нужны функции?
- > Передача данных в функцию и из неё

**Результат занятия:** научились писать функции, передавать и возвращать данные, изучили главные правила работы с функциями

**Практика:** создание приложения «Калькулятор»

### **Занятие 7    Ссылки и указатели**

- > Что такое адрес ячейки памяти?
- > Изучение правил работы указателей и ссылок

**Результат занятия:** ознакомились с устройством памяти компьютера, научились работать с указателями и ссылками

**Практика:** модификация приложения «Калькулятор»

### **Занятие 8    Динамические массивы**

- > В чём отличие от обычного массива?
- > Правила работы с динамическими массивами
- > Изучение операторов работы с памятью new и delete

**Результат занятия:** закрепили правила работы с массивами, изучили операторы new и delete

**Практика:** решение алгоритмических задач

### **Модуль 3. Основы теории алгоритмов**

#### **Занятие 9    Алгоритмы сортировки и поиска данных**

- > Как отсортировать любые данные?
- > Как найти данные в неизвестном множестве?

**Результат занятия:** изучили сортировку и поиск данных, разработали два алгоритма поиска и два алгоритма поиска данных.

**Практика:** разработка алгоритмов

#### **Занятие 10    Стандартная библиотека. Векторы и списки**

- > Знакомство с структурами данных вектор и список
- > Изучение области применения этих типов
- > Использование типов в решении задач

**Результат занятия:** изучили устройство структур данных, на практике использовали их в написании кода

**Практика:** решение практических задач

#### **Занятие 11    Стандартная библиотека. Словари и множества**

- > Зачем нужны словари и множества?
- > Что такое хэш-функции?
- > Использование словарей для решения задач

**Результат занятия:** изучили устройство структур данных, познакомились с хэшированием и подкрепили знания решением практического задания

**Практика:** решение практических задач

#### **Занятие 12    Оценка сложности алгоритмов**

- > Правила оценки сложности алгоритмов
- > Изучение методов оптимизации кода

**Результат занятия:** практическое исследование сложности алгоритмов, освоение навыка оценки сложности до запуска программы

**Практика:** решение практических задач

## Модуль 4. Объектно-ориентированное программирование (Часть 1. Знакомство)

### Занятие 13 Структуры и классы

- > Изучение основных концепций ООП
- > Создание пользовательских типов данных
- > Модификаторы доступа полей класса

**Результат занятия:** сделали первые шаги в изучении ООП, познакомились с классами и структурами, написали свои первые типы данных

**Практика:** разработка простой структуры данных

### Занятие 14 Методы класса

- > Зачем классу нужны методы?
- > Модификаторы доступа методов класса

**Результат занятия:** сравнили функции и методы, изучили модификаторы доступа `public` и `private`

**Практика:** модификация структуры данных с прошлого занятия

### Занятие 15 Конструктор и деструктор

- > Создание объектов класса
- > Удаление объектов класса
- > Изучение работы памяти компьютера

**Результат занятия:** изучили конструктор и деструктор, освоили правила создания данных методов

**Практика:** финальная модификация структуры данных из прошлого занятия

### Занятие 16 Перегрузка методов и функций

- > Что такое перегрузка?
- > Изучение ситуаций, в которых лучше использовать перегрузку, а в каких избегать
- > Ограничения перегрузок

**Результат занятия:** изучили важный механизм в устройстве языка, на практике исследовали перегрузку функций и методов

**Практика:** решение алгоритмических задач

## **Модуль 5. Объектно-ориентированное программирование (часть 2. Четыре важнейших принципа)**

### **Занятие 17 Абстракция**

- > Что такое абстракция в жизни?
- > Что такое абстракция в программировании?
- > Правила абстракции в ООП

**Результат занятия:** изучили абстракцию как важный механизм разработки программы, решили задачи для закрепления  
**Практика:** разработка класса «Утка»

### **Занятие 18 Инкапсуляция**

- > Что такое инкапсуляция?
- > Какие данные и методы необходимо прятать?
- > Изучение правил сохранения целостности данных

**Результат занятия:** освоили инкапсуляцию и правила применения данного принципа, на практике закрепили знания  
**Практика:** разработка класса «Озеро», настройка взаимодействия между классами

### **Занятие 19 Наследование**

- > Зачем нужно наследование?
- > Изучение главных принципов наследования
- > Изучение связи «родитель-ребенок» в программировании

**Результат занятия:** изучили наследование, разработали иерархию классов, освоили главные особенности наследования  
**Практика:** разработка иерархии классов

### **Занятие 20 Полиморфизм**

- > Что такое «абстрактный класс»?
- > Зачем нужны виртуальные методы?

**Результат занятия:** изучили абстрактные классы и виртуальные методы, на практике увидели, в каких случаях они применяются  
**Практика:** модификация иерархии классов

## **Модуль 6. Основной инструментарий программиста**

### **Занятие 21 Git и GitHub.com**

- > Что такое система контроля версий и зачем она нужна?
- > Изучение интерфейса программы
- > Изучение консольных методов программы
- > Изучение интеграции git в среду разработки

**Результат занятия:** освоили работу с системами контроля версий, создали свой первый репозиторий на github.com

**Практика:** создание git-репозитория

### **Занятие 22 Debugger и unit-тесты**

- > Что такое баг в коде?
- > Как найти в коде ошибку и избавиться от неё?
- > Изучение точек останова
- > Практика модульного тестирования

**Результат занятия:** на практике изучили создание и устранение ошибок в коде, научились отслеживать их в коде с помощью дебаггера, также разработали простую систему модульных тестов

**Практика:** решение практических задач

### **Занятие 23 Исключения, блоки try...catch...finally**

- > Зачем нужны исключения?
- > Правила отлова и обработки исключений

**Результат занятия:** изучили исключения, написали программу для обработки исключений

**Практика:** разработка программы с использованием исключений

### **Занятие 24 Создание и использование библиотек в коде**

- > Что такое API?
- > Правила разделения кода на заголовочные файлы и исходный код
- > Правила подключения файлов

**Результат занятия:** изучили подключение файлов и библиотек к программе, систематизировали предыдущие проекты

**Практика:** систематизация проектов

## **Модуль 7. Создание графических интерфейсов**

### **Занятие 25 Microsoft windows Forms**

- > Знакомство с интерфейсом и основными компонентами дизайнера форм
- > Создание прототипа программы
- > Основы дизайна интерфейсов

**Результат занятия:** познакомились с конструктором приложений, сделали первый набросок дизайна

**Практика:** первое графическое приложение

### **Занятие 26 Обработчики событий**

- > Как связать событие на экране с программным кодом?
- > Свойства элементов формы

**Результат занятия:** на практике рассмотрели интеграцию кода в логику графического приложения

**Практика:** продолжение работы с графическим приложением

### **Занятие 27 MessageBox и вызов дочерних форм**

- > Практика создания многооконных приложений
- > Создание всплывающих окон

**Результат занятия:** изучили методы разработки многооконных приложений, доработали собственное приложение

**Практика:** создание собственных предупреждений при неверных действиях пользователя

### **Занятие 28 Основы компьютерной графики, библиотека Graphics**

- > Рассмотрение основ компьютерной графики
- > 2D-координаты, математическое описание фигур
- > Хранение и обработка фигур

**Результат занятия:** познакомились с компьютерной графикой, на практике разработали параметрическую отрисовку объектов

**Практика:** разработка графического приложения с самостоятельной логикой графики

## **Модуль 8. Разработка собственного приложения**

### **Занятие 29 Проектирование логики собственного приложения**

- > Выбор идеи приложения
- > Создание плана разработки приложение, определение целей и задач каждого модуля
- > Начало разработки приложения

**Результат занятия:** разработали план-концепцию приложения, обсудили технологии и методы, которые будем использовать, начали разработку

**Практика:** план разработки и первые наброски проекта

### **Занятие 30 Разработка приложения**

- > Обсуждение результатов разработки
- > Разработка плана тестирования приложения
- > Помощь в разработке от опытного наставника

**Результат занятия:** разработали приложение, создали модульное тестирование, обсудили план дальнейших действий

**Практика:** продолжение разработки проекта

### **Занятие 31 Тестирование, исправление ошибок**

- > Продолжаем разрабатывать приложение, обсуждаем возникшие проблемы и находим решение
- > Тестируем приложение на ошибки

**Результат занятия:** доработали приложение, протестировали на типичные ошибки

**Практика:** продолжение разработки

### **Занятие 32 Окончательные доработки приложения, презентация проекта**

- > Завершение разработки приложения
- > Презентация проекта

**Результат занятия:** оценили полученные навыки и разработанное приложение, закрепили навыки презентации проекта

**Практика:** презентация проекта