

Ravesli [Ravesli](https://ravesli.com)

- Уроки по C++
- [OpenGL](#)
- [SFML](#)
- [Qt5](#)
- [RegExr](#)
- [Ассемблер](#)
- [Купить .PDF](#)



# Уроки программирования на языке C++

Здесь представлены более 200 бесплатных уроков по программированию на языке C++. Онлайн курсы программирования с нуля для начинающих, где рассматриваются основы и тонкости языка программирования C++. Бесплатное обучение программированию, а именно учебник с практическими заданиями и тестами. Неважно, имеете ли Вы опыт или нет, эти уроки по программированию помогут Вам начать создавать, компилировать и отлаживать программы на языке C++ в разных средах разработки: Visual Studio, Code::Blocks, Xcode или Eclipse.

Множество примеров и подробных разъяснений. Отлично подойдут как для новичков (чайников), так и для более продвинутых. Объясняется всё с нуля и до самых деталей абсолютно бесплатно!

**Также рассматривается пошаговое создание игры на C++ и 70+ заданий для проверки своих навыков программирования.**

Вы также можете приобрести все «Уроки по C++», вместе с Бонусом в формате .PDF без рекламы и в едином файле.

Оглавление:

1. [Уроки по C++](#)
2. [Пошаговое создание игры на C++/MFC](#)
3. [Задания по C++](#)
4. [Уроки по графической библиотеке SFML](#)
5. [Уроки по OpenGL](#)
6. [Уроки по Qt5](#)

## Глава №0. Введение. Начало работы

[Урок №1. Введение в программирование](#)

[Урок №2. Введение в языки программирования C и C++](#)

[Урок №3. Введение в разработку программного обеспечения](#)

[Урок №4. Установка IDE \(Интегрированной Среды Разработки\)](#)

[Урок №5. Компиляция вашей первой программы](#)

[Урок №6. Режимы конфигурации «Debug» и «Release»](#)

[Конфигурация компилятора: Расширения компилятора](#)

[Конфигурация компилятора: Уровни предупреждений и ошибки](#)

[Конфигурация компилятора: Выбор стандарта языка C++](#)

[Урок №7. Решения самых распространенных проблем](#)

## Глава №1. Основы C++

[Урок №8. Структура программ](#)

[Урок №9. Комментарии](#)

[Урок №10. Переменные, Инициализация и Присваивание](#)

[Урок №11. cout, cin и endl](#)

[Урок №12. Функции и оператор возврата return](#)

[Урок №13. Параметры и аргументы функций](#)

[Урок №14. Почему функции — полезны, и как их эффективно использовать?](#)

[Урок №15. Локальная область видимости](#)

[Урок №16. Ключевые слова и идентификаторы](#)

[Урок №17. Операторы](#)

[Урок №18. Базовое форматирование кода](#)

[Урок №19. Прототип функции и Предварительное объявление](#)

[Урок №20. Многофайловые программы](#)

[Урок №21. Заголовочные файлы](#)

[Урок №22. Директивы препроцессора](#)

[Урок №23. Header guards и #pragma once](#)

[Урок №24. Конфликт имён и std namespace](#)

[Урок №25. Разработка ваших первых программ](#)

[Урок №26. Отладка программ: степпинг и точки останова](#)

[Урок №27. Отладка программ: стек вызовов и отслеживание переменных](#)

[Глава №1. Итоговый тест](#)

## Глава №2. Переменные и основные типы данных в C++

[Урок №28. Инициализация, присваивание и определение переменных](#)

[Урок №29. Тип данных void](#)

[Урок №30. Размер типов данных](#)

[Урок №31. Целочисленные типы данных: short, int и long](#)

[Урок №32. Фиксированный размер целочисленных типов данных](#)

[Урок №33. Типы данных с плавающей точкой: float, double и long double](#)

[Урок №34. Логический тип данных bool](#)

[Урок №35. Символьный тип данных char](#)

[Урок №36. Литералы и магические числа](#)

[Урок №37. const, constexpr и символьные константы](#)

[Глава №2. Итоговый тест](#)

## Глава №3. Операторы в C++

[Урок №38. Приоритет операций и правила ассоциативности](#)

[Урок №39. Арифметические операторы](#)

[Урок №40. Инкремент, декремент и побочные эффекты](#)

[Урок №41. Условный тернарный оператор, оператор sizeof и Запятая](#)

[Урок №42. Операторы сравнения](#)

[Урок №43. Логические операторы: И, ИЛИ, НЕ](#)

[Урок №44. Конвертация чисел из двоичной системы в десятичную и наоборот](#)

[Урок №45. Побитовые операторы](#)

[Урок №46. Битовые флаги и битовые маски](#)

[Глава №3. Итоговый тест](#)

## Глава №4. Область видимости и другие типы переменных в C++

[Урок №47. Блоки стейтментов \(составные операторы\)](#)

[Урок №48. Локальные переменные, область видимости и продолжительность жизни](#)

[Урок №49. Глобальные переменные](#)

[Урок №50. Почему глобальные переменные – зло?](#)

[Урок №51. Статические переменные](#)

[Урок №52. Связи, область видимости и продолжительность жизни](#)

[Урок №53. Пространства имен](#)

[Урок №54. using-стейтменты](#)

[Урок №55. Неявное преобразование типов данных](#)

[Урок №56. Явное преобразование типов данных](#)

[Урок №57. Введение в std::string](#)

[Урок №58. Перечисления](#)

[Урок №59. Классы enum](#)

[Урок №60. Псевдонимы типов: typedef и type alias](#)

[Урок №61. Структуры](#)

[Урок №62. Вывод типов: ключевое слово auto](#)

[Глава №4. Итоговый тест](#)

## **Глава №5. Порядок выполнения кода в программе. Циклы, ветвления в C++**

[Урок №63. Операторы управления потоком выполнения программ](#)

[Урок №64. Операторы условного ветвления if/else](#)

[Урок №65. Оператор switch](#)

[Урок №66. Оператор goto](#)

[Урок №67. Цикл while](#)

[Урок №68. Цикл do while](#)

[Урок №69. Цикл for](#)

[Урок №70. Операторы break и continue](#)

[Урок №71. Генерация случайных чисел](#)

[Урок №72. Обработка некорректного пользовательского ввода](#)

[Урок №73. Введение в тестирование кода](#)

[Глава №5. Итоговый тест](#)**Глава №6. Массивы, Строки, Указатели и Ссылки в C++**[Урок №74. Массивы](#)[Урок №75. Фиксированные массивы](#)[Урок №76. Массивы и циклы](#)[Урок №77. Сортировка массивов методом выбора](#)[Урок №78. Многомерные массивы](#)[Урок №79. Строки C-style](#)[Введение в класс std::string\\_view в C++](#)[Урок №80. Указатели](#)[Урок №81. Нулевые указатели](#)[Урок №82. Указатели и массивы](#)[Урок №83. Адресная арифметика и индексация массивов](#)[Урок №84. Символьные константы строк C-style](#)[Урок №85. Динамическое выделение памяти](#)[Урок №86. Динамические массивы](#)[Урок №87. Указатели и const](#)[Урок №88. Ссылки](#)[Урок №89. Ссылки и const](#)[Урок №90. Оператор доступа к членам через указатель](#)[Урок №91. Цикл foreach](#)[Урок №92. Указатели типа void](#)[Урок №93. Указатели на указатели](#)[Урок №94. Введение в std::array.](#)[Урок №95. Введение в std::vector](#)[Введение в итераторы в C++](#)[Алгоритмы в Стандартной Библиотеке C++](#)[Глава №6. Итоговый тест](#)

## Глава №7. Функции в C++

[Урок №96. Параметры и аргументы функций](#)

[Урок №97. Передача по значению](#)

[Урок №98. Передача по ссылке](#)

[Урок №99. Передача по адресу](#)

[Урок №100. Возврат значений по ссылке, по адресу и по значению](#)

[Урок №101. Встроенные функции](#)

[Урок №102. Перегрузка функций](#)

[Урок №103. Параметры по умолчанию](#)

[Урок №104. Указатели на функции](#)

[Урок №105. Стек и Куча](#)

[Урок №106. Ёмкость вектора](#)

[Урок №107. Рекурсия и Числа Фибоначчи](#)

[Урок №108. Обработка ошибок, `cerr` и `exit\(\)`](#)

[Урок №109. `assert` и `static\_assert`](#)

[Урок №110. Аргументы командной строки](#)

[Урок №111. Эллипсис](#)

[Лямбда-выражения \(анонимные функции\) в C++](#)

[Лямбда-захваты в C++](#)

[Глава №7. Итоговый тест](#)

## Глава №8. Основы ООП в C++

[Урок №112. Введение в ООП](#)

[Урок №113. Классы, Объекты и Методы](#)

[Урок №114. Спецификаторы доступа `public` и `private`](#)

[Урок №115. Инкапсуляция, Геттеры и Сеттеры](#)

[Урок №116. Конструкторы](#)

[Урок №117. Список инициализации членов класса](#)

[Урок №118. Инициализация нестатических членов класса](#)

[Урок №119. Делегирующие конструкторы](#)

[Урок №120. Деструкторы](#)

[Урок №121. Скрытый указатель \\*this](#)

[Урок №122. Классы и заголовочные файлы](#)

[Урок №123. Классы и const](#)

[Урок №124. Статические переменные-члены класса](#)

[Урок №125. Статические методы класса](#)

[Урок №126. Дружественные функции и классы](#)

[Урок №127. Анонимные объекты](#)

[Урок №128. Вложенные типы данных в классах](#)

[Урок №129. Измерение времени выполнения \(тайминг\) кода](#)

[Глава №8. Итоговый тест](#)

## **Глава №9. Перегрузка операторов в C++**

[Урок №130. Введение в перегрузку операторов](#)

[Урок №131. Перегрузка операторов через дружественные функции](#)

[Урок №132. Перегрузка операторов через обычные функции](#)

[Урок №133. Перегрузка операторов ввода и вывода](#)

[Урок №134. Перегрузка операторов через методы класса](#)

[Урок №135. Перегрузка унарных операторов +, — и логического НЕ](#)

[Урок №136. Перегрузка операторов сравнения](#)

[Урок №137. Перегрузка операторов инкремента и декремента](#)

[Урок №138. Перегрузка оператора индексации \[\]](#)

[Урок №139. Перегрузка оператора \(\)](#)

[Урок №140. Перегрузка операций преобразования типов данных](#)

[Урок №141. Конструктор копирования](#)

[Урок №142. Копирующая инициализация](#)

[Урок №143. Конструкторы преобразования, ключевые слова `explicit` и `delete`](#)

[Урок №144. Перегрузка оператора присваивания](#)

[Урок №145. Поверхностное и глубокое копирование](#)

[Глава №9. Итоговый тест](#)

## Глава №10. Введение в связи между объектами в C++

[Урок №146. Типы связей между объектами](#)

[Урок №147. Композиция объектов](#)

[Урок №148. Агрегация](#)

[Урок №149. Ассоциация](#)

[Урок №150. Зависимость](#)

[Урок №151. Контейнерные классы](#)

[Урок №152. Список инициализации `std::initializer\_list`](#)

[Глава №10. Итоговый тест](#)

## Глава №11. Наследование в C++

[Урок №153. Введение в Наследование](#)

[Урок №154. Базовое наследование](#)

[Урок №155. Порядок построения дочерних классов](#)

[Урок №156. Конструкторы и инициализация дочерних классов](#)

[Урок №157. Наследование и спецификатор доступа `protected`](#)

[Урок №158. Добавление нового функционала в дочерний класс](#)

[Урок №159. Переопределение методов родительского класса](#)

[Урок №160. Соккрытие методов родительского класса](#)

[Урок №161. Множественное наследование](#)

[Глава №11. Итоговый тест](#)

## Глава №12. Виртуальные функции в C++



[Урок №162. Указатели/Ссылки и Наследование](#)

[Урок №163. Виртуальные функции и Полиморфизм](#)

[Урок №164. Модификаторы `override` и `final`](#)

[Урок №165. Виртуальные деструкторы и Виртуальное присваивание](#)

[Урок №166. Раннее и Позднее Связывания](#)

[Урок №167. Виртуальные таблицы](#)

[Урок №168. Чистые виртуальные функции, Интерфейсы и Абстрактные классы](#)

[Урок №169. Виртуальный базовый класс](#)

[Урок №170. Обрезка объектов](#)

[Урок №171. Динамическое приведение типов. Оператор `dynamic\_cast`](#)

[Урок №172. Вывод объектов классов через оператор вывода](#)

[Глава №12. Итоговый тест](#)

## Глава №13. Шаблоны в C++

[Урок №173. Шаблоны функций](#)

[Урок №174. Экземпляры шаблонов функций](#)

[Урок №175. Шаблоны классов](#)

[Урок №176. Параметр `non-type` в шаблоне](#)

[Урок №177. Явная специализация шаблона функции](#)

[Урок №178. Явная специализация шаблона класса](#)

[Урок №179. Частичная специализация шаблона](#)

[Урок №180. Частичная специализация шаблонов и Указатели](#)

[Глава №13. Итоговый тест](#)

## Глава №14. Исключения в C++

[Урок №181. Исключения. Зачем они нужны?](#)

[Урок №182. Обработка исключений. Операторы `throw`, `try` и `catch`](#)

[Урок №183. Исключения, Функции и Раскручивание стека](#)

[Урок №184. Непойманные исключения и обработчики catch-all](#)

[Урок №185. Классы-Исключения и Наследование](#)

[Урок №186. Повторная генерация исключений](#)

[Урок №187. Функциональный try-блок](#)

[Урок №188. Недостатки и опасности использования исключений](#)

[Глава №14. Итоговый тест](#)

## **Глава №15. Умные указатели и Семантика перемещения в C++**

[Урок №189. Умные указатели и Семантика перемещения](#)

[Урок №190. Ссылки r-value](#)

[Урок №191. Конструктор перемещения и Оператор присваивания перемещением](#)

[Урок №192. Функция std::move\(\)](#)

[Урок №193. Умный указатель std::unique\\_ptr](#)

[Урок №194. Умный указатель std::shared\\_ptr](#)

[Урок №195. Умный указатель std::weak\\_ptr](#)

[Глава №15. Итоговый тест](#)

## **Глава №16. Стандартная библиотека шаблонов (STL) в C++**

[Урок №196. Стандартная библиотека шаблонов \(STL\)](#)

[Урок №197. Контейнеры STL](#)

[Урок №198. Итераторы STL](#)

[Урок №199. Алгоритмы STL](#)

## **Глава №17. std::string в C++**

[Урок №200. Строковые классы std::string и std::wstring](#)

[Урок №201. Создание, уничтожение и конвертация std::string](#)

[Урок №202. Длина и ёмкость std::string](#)

[Урок №203. Доступ к символам std::string. Конвертация std::string в строки C-style](#)

[Урок №204. Присваивание и перестановка значений с std::string](#)

[Урок №205. Добавление к std::string](#)

## [Урок №206. Вставка символов и строк в std::string](#)

## **Глава №18. Ввод/Вывод в C++**

### [Урок №207. Потоки ввода и вывода](#)

### [Урок №208. Функционал класса istream](#)

### [Урок №209. Функционал классов ostream и ios. Форматирование вывода](#)

### [Урок №210. Потокотые классы и Строки](#)

### [Урок №211. Состояния потока и валидация пользовательского ввода](#)

### [Урок №212. Базовый файловый ввод и вывод](#)

### [Урок №213. Рандомный файловый ввод и вывод](#)

## **Дополнительные уроки**

### [Статические и динамические библиотеки](#)

### [Подключение и использование библиотек в Visual Studio](#)

### [C++11. Нововведения](#)

### [C++14. Нововведения](#)

### [C++17. Нововведения](#)

### [Спецификации исключений и спецификатор noexcept в C++](#)

### [Функция std::move\\_if\\_noexcept\(\) в C++](#)

## **Финал**

### [Конец? Что дальше?](#)

Адаптированный перевод с LearnCpp.com.

## **Пошаговое создание игры на C++/MFC**

[Урок №1: Введение в создание игры «Same Game» на C++/MFC](#)

[Урок №2: Архитектура и хранение данных в игре «Same Game» на C++/MFC](#)

[Урок №3: Отрисовка игры «Same Game» на C++/MFC](#)

[Урок №4: Обработка событий в игре «Same Game» на C++/MFC](#)

[Урок №5: Работа с алгоритмом в игре «Same Game» на C++/MFC](#)

[Урок №6: Работа с меню в игре «Same Game» на C++/MFC](#)

[Урок №7: Добавление уровней сложности в игре «Same Game» на C++/MFC](#)

[Урок №8: Размеры и количество блоков в игре «Same Game» на C++/MFC](#)

[Урок №9: Финальные штрихи в создании игры «Same Game» на C++/MFC](#)

Отдельное спасибо Дмитрию Бушуеву за помощь в переводе с [cprogramming.com](https://cprogramming.com).

## Задания по C++

[Часть №1: Практические задания по C++](#)

[Часть №2: Практические задания по C++](#)

[Часть №3: Практические задания по C++](#)

[Часть №4: Практические задания по C++](#)

[Часть №5: Практические задания по C++](#)

[Часть №6: Практические задания по C++](#)

[Часть №7: Практические задания по C++](#)

[Часть №8: Практические задания по C++](#)

[Часть №9: Практические задания по C++](#)

[Часть №10: Практические задания по C++](#)

[Часть №11: Практические задания по C++](#)

[Часть №12: Практические задания по C++](#)

[Часть №13: Практические задания по C++](#)

[Часть №14: Практические задания по C++](#)

[Часть №15: Практические задания по C++](#)

[Часть №16: Практические задания по C++](#)

[Часть №17: Практические задания по C++](#)

[Часть №18: Практические задания по C++](#)

[Часть №19: Практические задания по C++](#)

[Часть №20: Практические задания по C++](#)

[Часть №21: Практические задания по C++](#)

[Часть №22: Практические задания по C++](#)

[Часть №23: Практические задания по C++](#)

[Часть №24: Практические задания по C++](#)






TELEGRAM  КАНАЛ

Электронная почта



ПАБЛИК 

## ТОП СТАТЬИ

-  [Словарь программиста. Сленг, который должен знать каждый кодер](#)
-  [Урок №1. Введение в программирование](#)
-  [70+ бесплатных ресурсов для изучения программирования](#)
-  [Урок №1: Введение в создание игры «Same Game»](#)
-  [Урок №4. Установка IDE \(Интегрированной Среды Разработки\)](#)
- [Ravesli](#)
- - [О проекте](#) -
- - [Пользовательское Соглашение](#) -
- - [Все статьи](#) -
- Copyright © 2015 - 2020