

Программа по курсу «Программирование на C++» часть 1
для студентов 2-го курса направления Математика
2026-2027

1. Язык C++: назначение, основные принципы, история развития и современные стандарты. Сравнение C++ и Python. Компиляция и исполнение программы.
2. Структура простейшей программы. Точка входа main. Подключение заголовочных файлов. Пространства имен. Консольный ввод-вывод: `std::cin`, `std::cout`, `std::print` и `std::format`.
3. Фундаментальные типы данных. Размеры и предельные значения типов. Литералы. Объявление и инициализация переменных. Вывод типа `auto`. Константы `const` и `constexpr`. Явные и неявные преобразования типов. Переполнение. Сравнение вещественных чисел.
4. Основные операции и их приоритет. Условные конструкции `if-else`, тернарная операция и `switch`. Циклы `for`, `while` и `do-while`. Операторы `break` и `continue`. Вложенные циклы. Математические функции. Проверки `assert`.
5. Функции. Объявление и определение функции. Прототип функции. Возвращаемое значение и вывод возвращаемого типа. `void`-функции и досрочный выход.
6. Передача параметров по значению, ссылке и константной ссылке. Перегрузка функций. Аргументы по умолчанию. Инлайн-функции. Накладные расходы при вызове функции.
7. Многофайловая программа. Заголовочные файлы и файлы реализации. Директивы препроцессора. Условная компиляция. Стражи включения. Этапы сборки проекта. Ошибки компиляции и компоновки.
8. Объявления и определения. Глобальные и локальные объекты. `extern`. Пространства имен, вложенные и анонимные пространства имен, псевдонимы и `using`-директивы. Модули C++20: назначение и сравнение с заголовочными файлами.
9. Модульное тестирование. Организация основного и тестового проектов. Оформление, запуск и анализ тестов. Тестирование целых и вещественных значений, статических и динамических массивов, исключений, `std::vector` и `std::string`.
10. Структуры. Объявление, инициализация и доступ к полям. Передача структуры по значению, ссылке и константной ссылке. Возврат структуры и ссылки на структуру. Вложенные структуры, массивы структур и структурированные привязки.
11. Перечисления `enum` и `enum class`. Базовый тип перечисления. Использование перечислений в `switch`. `std::to_underlying`. Кортежи `std::tuple` и пары `std::pair`. Доступ и распаковка элементов. `std::tie`, `std::apply` и структурированные привязки. `std::initializer_list`. Псевдонимы типов `using` и `typedef`.
12. Системы счисления. Двоичная, восьмеричная и шестнадцатеричная запись. Представление целых и вещественных чисел в памяти.
13. Побитовые операции. Побитовое И, ИЛИ, исключающее ИЛИ и инверсия. Побитовые сдвиги. Битовые маски: установка, сброс, проверка и переключение битов. Упаковка данных.
14. Генерация псевдослучайных чисел. `std::random_device`, `std::mt19937`, `std::default_random_engine` и `std::seed_seq`. Равномерные и другие распределения. Сравнение библиотеки `<random>` с `rand()`.
15. Статические одномерные массивы. Размещение в памяти, объявление, инициализация и определение размера. Обход массива. Передача массива в функцию. Защита массива от изменения.
16. Указатели. Объявление и разыменование. `nullptr`. Указатели и ссылки. Указатель `void*`. Указатели на структуры. Массивы указателей и указатели на указатели. Указатели и константность. Связь массивов и указателей. Адресная арифметика.

17. Статические двумерные массивы. Организация и инициализация матрицы. Передача двумерного массива в функцию. Связь матриц и указателей.
18. `std::array` как контейнер фиксированного размера. Создание, инициализация, методы и итераторы. Передача `std::array` в функцию. `std::span` как невладеющее представление непрерывного диапазона. Время жизни и ограничения `std::span`. Метод `std::span::at` в C++26.
19. Указатели на функции. Объявление, инициализация и вызов. Передача указателя на функцию в качестве параметра. Callback-вызовы, предикаты и компараторы. Перегруженные функции и указатели на них. `std::function`. Чтение сложных объявлений.
20. Сортировки массивов. Сортировка пузырьком, выбором и вставками. Быстрая сортировка. Разбиение по схеме Хоара. Выбор опорного элемента и оптимизация быстрой сортировки.
21. О-нотация оценки сложности алгоритмов. Основные классы сложности. Вычисление временной и пространственной сложности. Лучший, средний и худший случаи. Устойчивость сортировки и сортировка на месте.
22. Ошибки времени выполнения и способы их обработки. Функции с побочным эффектом. Исключения: `throw`, `try` и `catch`. Перехват нескольких типов и всех исключений. Стандартные и пользовательские исключения. `noexcept`. Гарантии безопасности исключений. Идиома `copy-and-swap`. `std::expected`.
23. Области памяти программы. Стек и динамическая память. Операторы `new`, `delete`, `new[]` и `delete[]`. Утечки памяти, висячие указатели, двойное удаление, несоответствие `new` и `delete`, выход за границы и утечки при исключениях. Концепция умных указателей.
24. Динамические одномерные массивы. Выделение и освобождение памяти. Доступ к элементам и передача массива в функцию. Задание диапазона элементов. Изменение размера и копирование динамического массива.
25. Динамические двумерные массивы. Выделение и освобождение памяти. Передача матрицы в функцию. Ступенчатые динамические массивы.
26. Контейнер `std::vector`. Создание, инициализация и доступ к элементам. Размер и емкость. `reserve`, `shrink_to_fit` и `resize`. Добавление, вставка, удаление и очистка элементов. Сложность операций.
27. Итераторы. `begin`, `end`, `rbegin` и `rend`. Константные и обратные итераторы. Цикл range-based `for`. Недействительные итераторы. `assign`, `assign_range`, `insert_range` и `append_range`. `std::erase` и `std::erase_if`. Создание `std::span` из `std::vector`.
28. C-строки. Статические и динамические строки. Строковые литералы и нуль-терминатор. Обход строки с помощью индексов и указателей. Ввод C-строк. Передача C-строк в функции. Переполнение буфера, висячие указатели и ошибки управления памятью.
29. Стандартные функции работы с C-строками: `strlen`, `strcpy`, `strncpy`, `strcat`, `strncat`, `strcmp` и `strstr`. Реализация аналогичных функций. Вставка и удаление символов. Аргументы командной строки.
30. `std::string`. Создание и инициализация. Размер и емкость. Итераторы. Конкатенация, вставка, удаление, замена, поиск, получение подстроки и сравнение. Ввод с помощью `std::getline`. Передача строки в функцию. Преобразование строк и чисел.
31. `std::string_view` как невладеющее представление строки. Создание, методы, преимущества и опасности. Время жизни данных. Сравнение C-строк, `std::string` и `std::string_view`. Строковые возможности C++20 и C++23.
32. Рекурсия. Рекурсивное определение алгоритмов и структур данных. Базовый случай и рекурсивный шаг. Прямая, косвенная, вложенная и концевая рекурсия. Стек вызовов. Бесконечная рекурсия. Сравнение рекурсии и итерации. Рекурсивный бинарный поиск.

33. Шаблоны функций. Синтаксис, параметры и инстанцирование шаблона. Два этапа компиляции. Размещение шаблонов. Перегрузка и специализация шаблонов. Сравнение шаблонов с перегруженными функциями и макросами.
34. Шаблоны с переменным числом аргументов. Fold expressions. constexpr-функции и рекурсия. static_assert. Нетиповые параметры и шаблоны переменных. Ограничения requires и концепты C++20.
35. Динамические структуры данных. Размерная и ссылочная модели объектов. Списочные структуры и их классификация. Представление списка в памяти. Сравнение массивов и списков.
36. Линейный односвязный список. Структура узла. Создание, обход и очистка списка. Вставка в начало, конец и произвольное место. Удаление первого, последнего и заданного узла. Поиск и обработка с предикатом. Обработка соседних элементов.
37. Двусвязный двунаправленный список. Структура узла и согласование связей prev и next. Вставка и удаление в начале, конце и середине. Обход в обратном порядке. Сравнение односвязного и двусвязного списков. std::list.
38. Деревья. Рекурсивное определение и основные термины. Уровень, глубина и высота. Бинарные, полные, идеально сбалансированные, AVL-сбалансированные и несбалансированные деревья. Структура узла. Создание и удаление дерева. Префиксный, инфиксный и постфиксный обходы. Печать дерева с отступами. Рекурсивные алгоритмы поиска и обработки.
39. Бинарное дерево поиска. Свойство дерева поиска. Работа с повторяющимися ключами. Поиск, вставка и удаление узлов. Удаление листа, узла с одним потомком и узла с двумя потомками. Сортировка с помощью дерева поиска. Сложность операций.
40. Абстрактный тип данных. Интерфейс, инкапсуляция и модульность. Стек и его основные операции. Реализация стека на статическом и динамическом массивах и на односвязном списке.
41. Очередь и ее основные операции. Реализация очереди на массиве и односвязном списке. Кольцевая очередь. Дек и его реализация на двусвязном списке. Очередь с приоритетом и варианты ее реализации.
42. Множество и ассоциативный массив. Хеш-таблица. Хеш-функция, бакеты и коллизии. Требования к хеш-функции. Способы разрешения коллизий и применения хеш-таблиц.
43. Файлы и потоки. Классы файловых потоков. Открытие и закрытие файла. Ошибки открытия и состояние потока. Режимы открытия файла. Пути. std::filesystem: обход каталогов, создание и удаление каталогов, переименование и удаление файлов.
44. Текстовые файлы. Форматированный и построчный ввод. Запись и чтение текстовых данных. Обработка строк и слов. Форматирование вывода классическими средствами, std::format и std::print.
45. Бинарные файлы. Открытие в бинарном режиме. Функции read и write. Запись и чтение структур. Ограничения прямой записи объектов. Сериализация. Произвольный доступ и работа с массивом записей в файле.
46. Приведение типов. Приведение в стиле C. Операторы static_cast, const_cast, reinterpret_cast и dynamic_cast. Назначение, ограничения и опасности операторов приведения C++. std::to_underlying.