

Программа экзамена по дисциплине

Программирование на C++

2023-2024 уч. год

сетевой бакалавриат МФТИ

1. Базовые конструкции

- (a) Алфавит языка. Ключевые слова. Идентификаторы. Литералы. Знаки операций.
- (b) Структура программы на C++. Объявления и определения. Операторы. Комментарии.
- (c) Концепция типа данных. Фундаментальные типы данных: целые типы, символьный тип, логический тип, типы с плавающей точкой, `void`. Операции над фундаментальными типами.
- (d) Переменные. Классы памяти и области действия. Область видимости. Инициализация. CV-квалификаторы.
- (e) Выражения. Особенности работы операций. Категории значения. `static_cast`, `sizeof`.

2. Операторы управления

- (a) Условная операция (`?:`). Условный оператор (`if – else`).
- (b) Циклы: `while`, `do – while`, `for`.
- (c) Оператор `switch`.
- (d) Операторы `break` и `continue`. Оператор безусловного перехода.

3. Массивы и указатели

- (a) Операция взятия адреса. Указатели. Разыменовывание указателя. Арифметика указателей. Константные указатели и указатели на константы. Нулевой указатель. Провисший указатель.
- (b) Ссылки. Правила работы со ссылками. Ссылки на константу.
- (c) Массивы в стиле C. Операции над массивами. Инициализация массивов. Связь массивов и указателей. Указатель на массив. Многомерные массивы.
- (d) Динамическая память. Выражения `new` и `delete`. Динамические массивы.
- (e) Строковые литералы. Строки в стиле C. Функции для работы со строками (`<cstring>`).

4. Функции

- (a) Функции. Объявление и определение. Возвращаемое значение, аргументы, аргументы по умолчанию.
- (b) Перегрузка функций. Статические переменные функции.
- (c) Передача и возврат указателей, ссылок и массивов в/из функции.
- (d) Рекурсия: прямая, косвенная. Примеры рекурсивных вычислений.

5. Пользовательские типы

- (a) Перечисления (`enum`)
- (b) объединения (`union`)
- (c) структуры (`struct`).

6. Шаблоны

- (a) Шаблоны функций. Шаблоны классов. Шаблоны переменных. Шаблоны псевдонимов типов (`using`).

- (b) Вывод типа аргумента шаблона. Параметры шаблона по умолчанию. Перегрузка шаблонов функций. Объявление *auto* переменных.
- (c) Специализация шаблонов. Инстанцирование шаблонов.

7. Классы. Начало

- (a) Парадигмы программирования. Объектно-ориентированное программирование. Абстракция, инкапсуляция, полиморфизм, наследование.
- (b) Классы и объекты. Поля и методы. Модификаторы доступа. Друзья класса.
- (c) Константные и статические поля и методы. Логическая и физическая константность. Ключевое слово *mutable*.

8. Конструкторы и деструктор

- (a) Агрегатная инициализация. Конструктор. Виды конструкторов. Списки инициализации. Делегирующие конструкторы. Ключевое слово *explicit*.
- (b) Конструкции *= default* и *= delete*.
- (c) Деструктор. Идиома RAII. Порядок вызова конструкторов и деструкторов. Правило трех.

9. Сборка программ на C++ и перегрузка операций

- (a) Перегрузка операций. Правила перегрузки операций. Перегрузка побитового сдвига для потокового ввода/вывода.
- (b) One Definition Rule. Директива *#include*, проблема двойного включения, стражи включения.
- (c) Сборка программ на C++. Раздельная компиляция, модули (до C++20). Встраиваемые функции и переменные. Внутреннее и внешнее связывание (*static*, *extern*).
- (d) Пространства имен, директивы *using* и *using namespace*.

10. Функциональные объекты и алгоритмы

- (a) Функторы. Классы с перегруженной операцией функционального вызова. Лямбда-выражения
- (b) Алгоритмы. Модифицирующие алгоритмы. Немодифицирующие алгоритмы. Алгоритмы сортировки. Алгоритмы упорядоченных последовательностей. Алгоритмы для работы с пирамидой.

11. Контейнеры

- (a) Последовательные контейнеры: *std :: array*, *std :: vector*, *std :: deque*, *std :: list*, *std :: forward_list*, *std :: initializer_list*.
- (b) Ассоциативные контейнеры: *std :: map*, *std :: set.unordered*— и *multi*— версии. *Structured bindings*.
- (c) Контейнерные адаптеры: *std :: stack*, *std :: queue*, *std :: priority_queue*.

12. Наследование и Статический полиморфизм

- (a) Наследование. Модификаторы доступа при наследовании. Конструкторы и деструкторы при наследовании.
- (b) Срезка. Затенение методов базового класса. Работа с производными классами через указатель или ссылку на базовый.

13. Динамический полиморфизм

- (a) Динамический полиморфизм. Виртуальные функции. Идентификаторы *override* и *final*. Ковариантные возвращаемые типы.

- (b) Виртуальный деструктор. Позднее и раннее связывание, таблица виртуальных функций.
- (c) Чисто виртуальные функции. Абстрактные классы.

14. Исключения

- (a) Исключения. Ключевое слово *throw*. Блок *catch*.
- (b) Статическая спецификация исключений. Ключевое слово *noexcept* (спецификатор, условный спецификатор, операция). Исключения в конструкторах и деструкторах.
- (c) Гарантии безопасности исключений. Иерархия исключений в C++.

15. Итераторы

- (a) Итераторы. Константные итераторы. Категории итераторов. Класс *std::iterator_traits*. Обобщенные функции для работы с итераторами (*std::advance*, *std::next*, *std::prev*, *std::distance*). Инвалидация итераторов. *Range – based for*.
- (b) Итераторные адаптеры. Итераторы вставки. Обратные итераторы. Итераторы перемещения. Поточные итераторы.

16. Категории значений и конструктор перемещения

- (a) Категории значений (lvalue, rvalue). lvalue- и rvalue-ссылки. Перегрузка функций по типу ссылки.
- (b) Конструктор перемещения и перемещающее присваивание. Правило пяти.

17. Семантика перемещения

- (a) Функция *std::move*. Примеры использования.
- (b) Функция *std::forward*.
- (c) *Copy elision*. Оптимизация возвращаемого значения (*RVO*, *NRVO*).
- (d) Универсальные ссылки (*forwarding references*). Правила вывода шаблонных параметров для универсальных ссылок. Примеры применения.