

Теория к КР #1 (2-й семестр):

1. Ссылки, пространство имён и другое.

Что такое ссылки. Различие ссылок и указателей. Ссылки на константу. 3 типа передачи аргументов в функцию. Передача по значению, передача по ссылке и передача по ссылке на константу. Преимущества каждого метода. Возвращение ссылки из функции. Перегрузка функций. Функции с аргументами по умолчанию. Пространство имён: что такое и зачем нужно. Строка `std::string`. Различие строки в стиле C и строки `std::string`.

2. Перегрузка операторов

Перегрузка операторов в языке C++. Перегрузка арифметических операторов. Перегрузка операторов с использованием методов класса. Перегрузка унарных операторов. Перегрузка операторов ввода вывода `<<` и `>>` с `cin` и `cout`. Преимущества перегрузки операторов.

3. ООП: Инкапсуляция

Что такое объектно-ориентированное программирование. Основные принципы ООП: инкапсуляция, наследование и полиморфизм. Классы. Поля и методы класса. Указатель `this`. Модификаторы доступа `private` и `public`. Различие ключевых слов `struct` и `class` в языке C++. Друзья. Ключевое слово `friend`. Конструкторы и деструкторы. Список инициализации членов класса. Перегрузка конструкторов. Создание экземпляров класса в стеке и куче в языке C++. Использование операторов `new` и `delete`. Основные отличия `new` и `delete` от `malloc` и `free`.

4. ООП: Наследование и полиморфизм.

Наследование в языке C++. Добавление новых полей и методов в наследуемый класс. Вызов конструкторов наследуемого класса. Модификатор доступа `protected`. Указатели на базовый класс, хранящие адрес объекта наследуемого класса. Полиморфизм в C++. Виртуальные функции. Таблица виртуальных функций. Абстрактные классы и интерфейсы.

5. Шаблоны.

Шаблонные функции. Использование шаблонных функций в языке C++. Шаблоны классов. Стандартная библиотека шаблонов.

6. Стандартная библиотека шаблонов(STL).

Динамический массив `std::vector` и его реализация. Размер и вместимость вектора. Использование `std::vector`. Методы `push_back`, `reserve` и `resize`. Связный список `std::list` и его реализация. Множества `std::set` и `std::unordered_set` и их реализация(с помощью какой структуры данных они реализованы). Методы `insert`, `erase` и `find`. Обход стандартных контейнеров. Итераторы.

7. Раздельная компиляция

Что такое файл исходного кода и исполняемый файл. Этап сборки программы: препроцессинг, ассемблирование, компиляция и линковка. Директивы препроцессора `#include` и `#define`. Компиляция программы с помощью `gcc`. Header-файлы. Раздельная компиляция. Преимущества раздельной компиляции. Статические библиотеки и их подключение с помощью компилятора `gcc`. Динамические библиотеки и их подключение. Скрипты `bash`. Make-файлы.