

Где может быть определена структура или класс?

- В каких файлах определить класс или структуру
- Можно ли объявлять класс или структуру внутри других классов
- Какая область видимости
- Как получить доступ
- Можно ли объявлять класс внутри функций

Допустимо ли использование указателей/ссылок/массивов структур?

- Если можно, то каким образом это выглядит в коде
- Как создается массив структур
- Сколько памяти занимает, и.т.д.
- в чем особенность передачи по ссылке/указателю

Как осуществляется работа с динамической памятью (механизм или может конкретно calloc, malloc, new, delete)?

- что такое динамическая память
- как с ней работать (обращаться и изменять)
- в чем разница между calloc, malloc и new delete

Что такое exception-safe операция?

- Уровни гарантии
- Какие операции exception-safe
- Какие ключевые слова используются
- Что будет, если в exception-safe операции произойдет исключение
- move-семантику

В чем отличие синтаксической и логической константности методов?

- Синтаксическая константность это при использовании ключевого слово const
- Логическая конс-ть это гарантия от "программиста"

- ```
class A{
 int b;
public:
 int get(){return b;}
};
```

## Move-семантика

- Что это вообще такое.
- Кратко про операции `std::move` и `std::swap`
- В каких случаях move-семантика работает неявно
- Конструктор перемещения
- Кратко затронуть про lvalue и rvalue ссылки

- ```
int a = b + 5;
// a - lvalue
// b - lvalue, неявно преобразовывается к rvalue
// 5 - rvalue
```

Чем отличаются механизмы раннего и позднего связывания?

- Связывание - то, какой код нужно выполнить в том или ином месте. Какую функцию вызвать, иногда, какой класс использовать
- Раннее связывание - происходит на этапе компиляции. Например, перегрузка функций.
- Позднее связывание - происходит в runtime. Например, выбор виртуальной функции.
- `auto` -

- ```
auto it = v.begin();
```

## Как работает преобразование в Си-стиле на языке C++?

- `static_cast` и `dynamic_cast`

- ```
double a = 5.5;
int b = (int) a;
```

- в чем разница от `static_cast`
- Что нужно, чтобы мы могли приводить структуры и классы в другому типу в C-стиле

Странное поведение

```
int a = 7;

int& foo(){
    int a = 7;
    return a;
}

foo() = 5;
```

Для каких типов допустима перегрузка операторов?

```
class A{
public:
    A operator+(const A&, int){}
    A operator+(const A&, int, int){}

};

int operator+(int,int){}

enum B = {q,w,e,r};
```

Как происходит возбуждение исключения

- Как оно возникает?
- Как мы можем сами вызвать исключение
- Как происходит обработка
- Пояснить, что именно можно выдавать исключение
- Набор стандартных исключений

Что такое ключевое слово using и его связь с модификаторами доступа

- Как при наследовании изменить поле, которое в базовом классе было public, а в наследнике стало private
- Наследование конструкторов при помощи using

В чем особенности работы умных указателей с массивами?

- `shared_ptr<A*>`
- `shared_ptr<shared_ptr>`
- как это связано с move-семантикой
- `vector<shared_ptr>`
- `vector<unique_ptr>`