



C++

## Разбор домашней работы



# Теория



20 минут



# Пример

```
MyVector a(n), b;
```

```
a.read();
```

```
b = a;
```

Что произойдет?

Запустится ли конструктор копирования?

**Почему конструктор  
копирования не будет  
запущен?**

# Конструктор копирования

**Конструктор копирования** - так или иначе конструктор, а значит запускается только в момент создания объекта.

Для уже существующего объекта запустить конструктор копирования нельзя.

Что в таком случае делать?

# **Перегрузка операции присваивания**

# Операция присваивания

**Операция присваивания** - как и любая другая операция в C++, это функция.

Как и любую другую функцию, операцию присваивания можно переопределить для нового набора параметров.





# Что является результатом присваивания?

`a = b = c;`

Какое действие выполняется первым?

Что должно быть результатом присваивания, чтобы второе действие также выполнилось?

# Что является результатом присваивания?

`a = b = c;`

Какое действие выполняется первым?

`b = c;`

Что должно быть результатом присваивания, чтобы второе действие также выполнилось?



# Что является результатом присваивания?

`a = b = c;`

Какое действие выполняется первым?

`b = c;`

Что должно быть результатом присваивания, чтобы второе действие также выполнилось?

Переменная `b`

# Операция присваивания

Имя функции, являющейся перегрузкой какой-либо операции, состоит из ключевого слова **operator** и значка перегружаемой операции.

Для класса `MyVector` заголовок функции будет следующим:

```
MyVector& operator=(const MyVector& a);
```

Присваивание реализуется как метод класса, т.е. внутри класса.

# Перегрузка присваивания

```
class MyVector {  
    int *data, len;  
public:  
    MyVector(int len = 0, int value = 0);  
    ~MyVector();  
  
    MyVector& operator=(const MyVector& a) {  
        delete[] data;  
        // В отличие от копирования, при присваивании мы работаем с существующим  
        // объектом, а значит там уже есть свой динамический массив  
        len = a.len;  
        data = new int[len];  
        for (int i = 0; i < len; i++)  
            data[i] = a.data[i];  
        return *this;  
    }  
};
```

**Что произойдет, если объекту  
присвоить его самого?**

**`a = a;`**

**Как проверить, что это другой объект?**

# Перегрузка присваивания

```
class MyVector {  
    int *data, len;  
public:  
    MyVector(int len = 0, int value = 0);  
    ~MyVector();  
  
    MyVector& operator=(const MyVector& a) {  
        if (&a == this) return *this;  
        // Если возникает самоприсваивание - завершаем функцию  
        delete[] data;  
        len = a.len;  
        data = new int[len];  
        for (int i = 0; i < len; i++)  
            data[i] = a.data[i];  
        return *this;  
    }  
};
```



# Практика



15 минут



# Разбор



10 минут



# Практика



25 минут



# Итоги урока

- 1) Повторили работу конструктором копирования
- 2) Узнали, что можно перегружать в том числе и операции
- 3) Узнали, как перегрузить операцию присваивания