

Конструктор копирования

29 мая 2017 г.

Виды конструкторов

1. Конструктор по умолчанию
2. Конструктор с параметрами
 - явный (неконвертирующий)
 - конвертирующий
3. Делегирующий конструктор (C++11)
4. Копирующий конструктор
5. Конструктор перемещения (C++11)

Копирующий конструктор

- имеет один аргумент (ссылка на объект класса)
- нужен для копирования объектов класса
- может генерироваться автоматически
- копирующий конструктор по умолчанию делает **побитовую копию** объекта
- вызывается всегда, когда создается копия

`X(const X& myObj);`

где X – имя класса

Когда нужен копирующий конструктор?

1. Передача объекта в функцию

- при передаче объекта в функцию создается новый объект
- при выходе из функции временный объект разрушается (вызывается деструктор копии)

```
void f (X obj) { ... }
```

```
...
```

```
X myObj;
```

```
f(myObj);
```

Когда нужен копирующий конструктор?

2. Возврат объекта из функции

- при возвращении объекта создается копия, которая потом разрушается

```
X f () {  
    X myObj;  
    return myObj;  
}  
...  
X res = f();
```

Когда нужен копирующий конструктор?

3. Инициализация одного объекта другим при создании (используется оператор =)

```
X myObj;
```

```
// побитовое копирование
```

```
X myNewObj = myObj;
```

Копирующий конструктор: пример

```
class Array {  
    int* p;  
    int size;  
public:  
    Array(int sz) : size(sz) {  
        p = new int[sz];  
    }  
    ~Array() { delete [] p; }  
    Array(const Array &a) : size(a.size) {  
        p = new int[size];  
        for(int i = 0; i < a.size; ++i) { p[i] = a.p[i]; }  
    }  
};
```

Копирующий конструктор

Если в классе есть конструктор копирования, то в классе нужно реализовать деструктор.

Задание

Выберите правильные объявления
конструктора копирования в классе Date.

- 1) Date(const Date& d);
- 2) Date(const Date d);
- 3) Date(Date& d);
- 4) Date(Date d);
- 5) void Date(Date& d);

Вопросы?