

Деструктори, оператор delete

Оператор delete

Както споменахме по-рано, можем да заделяме памет в `heap`-а чрез оператора `new`. Да припомним - **когато заделяме динамична памет, програмата или операционната система не се грижи за това тази памет да се освободи**

Как става това освобождаване на памет ?

Освобождаването на памет заделена с `new` се изтрива с оператора `delete`.

Примери

```
int* dynamicInt = new int; //Заделяме памет за 1 int
...

...
delete dynamicInt; //Изтриваме паметта за този int
```

```
int* dynamicArr = new int[10]; //Заделяме памет за масив
...

...
delete[] dynamicArr;
```

```
int** matrix = new int*[10]; //Заделяме памет за 2d масив
for(int i = 0; i < 10; i++)
{
    matrix[i] = new int[10]; //Заделяме колоните
}
...

...
for(int i = 0; i < 10; i++)
{
    delete[] matrix[i]; //Трием първо колоните
}
delete[] matrix; //След това трием първият масив
```

Когато имаме вложена динамична памет:

- Заделяме външна към вътрешна
- Изтриваме вътрешна към външна

Жизнен цикъл на обекти

Нека разгледаме примерът:

```
int foo()  
{  
    int a = 5;  
    //Момент в  
    return a;  
}  
int main()  
{  
    //Момент а  
    int b;  
    //Момент б  
    b = foo();  
    //Момент г  
}
```

Колко ще е заетата памет от променливи във всеки от един от четирите момента ?

Момент а - 0 (няма заделени променливи)

Момент б - 4В (заделена е само **b**)

Момент в - 8В (заделени са **a** и **b**)

Момент г - 4В (отново е заделена само **b**)

Забелязваме, че след като е завършено изпълнението на някоя функция, променливите в нея се изтриват от паметта. Същото се случва и в обектите на класовете. Функцията, която се грижи за това изтриване на обекти се нарича **деструктор**

Деструктор

Деструкторът е специален метод на класовете, чиято цел е да се погрижи за "разрушаването" на обектите от даден клас. **Важно е да се подчертае, че ако не използваме динамична памет, не е нужно да се дефинира явен деструктор. Компиляторът прави свои си деструктор, ако в кода не е посочен такъв. Всеки клас има точно един деструктор**

Деструктор се дефинира по следният начин:

```
class Person  
{  
    char* name;  
  
    Person()  
    {  
        name = new char[50];  
    }  
    ...  
  
    ...  
    ~Person()  
}
```

```

    {
        delete[] name;
    }
};

```

Нека разгледаме деструктора за нашия клас `MyVector`

```

#pragma once

class MyVector
{
    private:
        int* values;
        int size;
        int maxSize;

        void resize();
    public:
        MyVector(); //Default
        MyVector(const MyVector&); //Copy
        MyVector& operator=(const MyVector&); //Operator=

        void set_element(int, int); //pos, val
        void push_back(int);
        void insert(int, int); //pos, val
        void remove(int); //pos
        void print() const;

        int get_element(int) const;
        int pop_back();
        int begin() const;
        int end() const;
        int get_size() const;

        ~MyVector();
};

```

```

MyVector::~MyVector()
{
    delete[] values;
}

```