

## Тема 6.

Динамична памет в класовете. Голямата сетворка.  
Пример за клас Student с име и масив с оценки (с произволна дължина)

---

### Динамична памет в класовете

- При генерираните от компилатора копирани констр. и оператор = се ползвава проблем
- ползвава се shallow copy, т.е. има 2 различни обекта с 2 указателя, които сочат към една и съща динамична памет. В този случай при унищожаване чрез 2 пъти едно и също външен ресурс

Решение: 1) забраняваме копирането:

$A(\text{const } A\& \text{ other}) = \text{delete};$

$A\& \text{ operator} = (\text{const } A\& \text{ other}) = \text{delete};$

2) разписваме експлицитно копирането:

```
class A {
```

```
    int x;
```

```
    B* ptr;
```

```
    A(const A& other) {
```

```
        x = other.x
```

```
        ptr = new B(other.ptr);
```

```
    }
```

```

A & operator = (const A & other) {
    if (this != &other) {
        x = other.x;
        ptr = new B(other.ptr);
    }
    return *this;
}
};

```

} може да се изнесе в отделна функция

Така създаваме deep copy. Заделили сме нова памет и указателите сочат към различни места.

## Голямата четворка

- копиращ конструктор
- def. конструктор (по време на излизане от к.к.)
- оператор =
- деструктор (изчиства взетите ресурси)

Разписваме голямата четворка само когато имаме динамична памет в класа