

При динамичен памет в класовете с инициализация от компилатора к.к. и оп-получаваме shallow copy - дв-различни указатели сочат към един и същ външен ресурс.

⇒ 1) можем да забравим копирането

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

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

2) можем да фиксируем експлицитно копирането и триенето

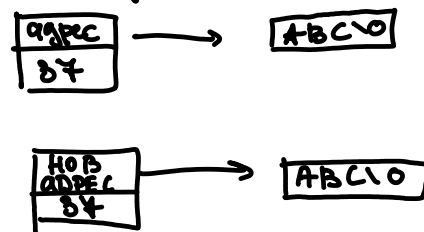
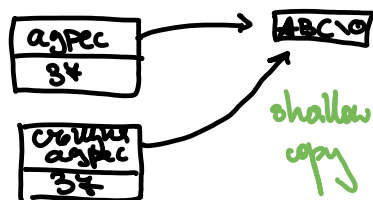
При динамичен памет в клас се налага да фиксируем:

- деф. к.р.
- к.к
- оп=
- дефр

К.к - копира
оп = - трие и копира
дефр - трие

⇒ и носим триенето и копирането в отделни ф-ии

Големата четворка



```
struct Person {
private:
    char* name;
    int age;
}
```

Големата 4-ка:

```
Person(const char* name, int age)
{
    setName(name);
    setAge(age);
}
```

```
Person(const Person& other)
{
    copyFrom(other);
}
```

```
Person& operator = (const Person& other)
{
    // ...
}
```

```

Person & operator = (const Person & other)
{
    if (&this == &other)
    {
        free();
        copyFrom(other);
    }
    return *this;
}

~Person()
{
    free();
}

```