* 1. **Деструкторы**

**Деструктор** – это особый вид метода, применяющийся для освобождения памяти, занимаемой объектом. Деструктор вызывается автоматически, когда объект выходит из области видимости:

* для *локальных* объектов – при выходе из блока, в котором они объявлены;
* для *глобальных* – как часть процедуры выхода из **main;**
* для *объектов, заданных через указатели*, деструктор вызывается неявно при использовании операции **delete.**

 Имя деструктора начинается с тильды (**~**), непосредственно за которой следует имя класса. Деструктор:

* не имеет аргументов и возвращаемого значения;
* не может быть объявлен как **const** или **static**;
* не наследуется;
* может быть виртуальным.

Если деструктор явным образом не определен, компилятор автоматически создает пустой деструктор.

Описывать в классе деструктор явным образом требуется в случае, когда объект содержит указатели на память, выделяемую динамически – иначе при уничтожении объекта память, на которую ссылались его поля-указатели, не будет помечена как свободная.

Деструктор для рассмотренного примера в Лекции 3 должен выглядеть так:

prostr:: ~prostr()

{

for(int i=0;i<k;i++)

delete[] a[i];

delete[] a;

}

Деструктор можно вызвать явным образом путем указания полностью уточненного имени, например:

prostr p; ...

p.~ prostr();

Это может понадобиться для объектов, которым с помощью перегруженной операции **new** выделялся конкретный адрес памяти. Без необходимости явно вызывать деструктор объекта не рекомендуется.