

Тема 17: Инициализация и разрушение объектов



▣ Конструктор

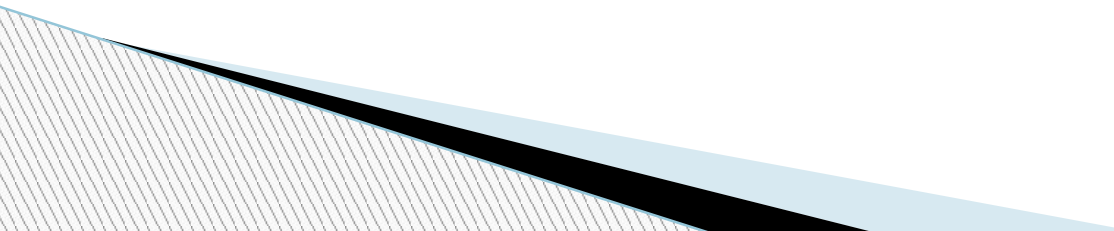


- ▣ Функция, называемая конструктором (constructor) класса, имеет то же имя, что и класс. Подобно другим функциям C++, конструктору могут передаваться параметры, используемые для инициализации одного или более данных-полей класса. В классе Rectangle конструктору дается имя Rectangle, и он принимает параметры l и w, используемые для инициализации длины и ширины объекта, соответственно. Заметьте, что эти параметры имеют значения по умолчанию, которые указывают, что используется значение 0, когда параметр l или w не передается явно.

▣ *Деструктор*

▣

- ▣ Динамическое выделение памяти для объектов какого – либо класса создаёт необходимость в освобождении этой памяти при уничтожении объекта. Такую возможность даёт специальный метод класса – *деструктор*. Так если в конструкторе класса «матрица» выделяется память для 10 целочисленных элементов, то в деструкторе эта память должна освобождаться.

- ▣ *Конструкторы и деструкторы* особые члены класса, служащие для инициализации и уничтожения объекта. Особенности:
 - ▣
 - ▣ они не возвращают никакого значения даже `void`;
 - ▣ не участвуют в механизме наследования;
 - ▣ нельзя получить их адрес;
 - ▣ конструкторы не могут быть виртуальными;
 - ▣ деструктор не имеет аргументов, а значит его нельзя перегружать;
 - ▣ автоматически вызываются при создании и уничтожении объекта.
- 

- Конструкторы имеют тоже имя, что и имя класса. Конструктор принимающий ссылку на собственный класс, называется *конструктором копирования*. Конструктор без аргументов называется *пустым конструктором*. Если в классе конструкторы явно не определены, то создается пустой конструктор по умолчанию. Также если пустой конструктор определен, то он используется для инициализации объекта по умолчанию.
-
- Деструктор имеет тоже имя, что и имя класса, но перед именем ставится знак `~`. При инициализации объекта, вначале инициализируются его родительские объекты в порядке включения.

