***Конструктор*** служит для создания и инициализации объектов класса. В отличие от других методов при описании конструктора не указывается тип возвращаемого значения, а его имя совпадает с именем класса.

[модификаторы] имя\_класса (список\_параметров)

{

тело конструктора

}

В зависимости от вариантов инициализации объекта при создании можно определять несколько конструкторов, отличающихся списком параметров. В этом случае говорят о перегрузке конструкторов. Если в классе явно не определено ни одного конструктора, то он предоставляется по умолчанию. Этот конструктор присваивает нулевые значения всем полям класса.

В качестве модификаторов могут использоваться следующие: ∙ static;

∙ модификаторы доступа: public, protected, private, internal.

Статический конструктор служит для инициализации статических полей класса. Он вызывается один раз при создании первого объекта класса в программе.

Если в классе существуют несколько конструкторов, можно один конструктор вызывать из другого. Это делается с помощью ключевого слова this и указания всех параметров вызываемого конструктора.

Если объект больше не нужен, память, занимаемая объектом, должна быть освобождена. С# обладает механизмом освобождения ресурсов памяти, называемым ***«сборщик мусора»***. Если к объекту не происходит обращение (в программе не осталось ссылок на объект), этот механизм освобождает память автоматически. Сборщик мусора вызывается периодически в процессе работы программы.

Когда объект использует внешние ресурсы (например, файлы, соединение с другими устройствами, подключение к базам данных и т.д.), для корректного освобождения этих ресурсов может использоваться специальный метод, который называется ***деструктором***. Он вызывается сборщиком мусора при уничтожении объекта. Деструктор в классе может быть только один. Его имя совпадает с именем класса, предваренное символом ‘~’. У деструктора нет возвращаемого значения и нет параметров. Он не имеет модификаторов.

~Имя\_Класса()

{

. . .

}