***Инициализаторы*** *объектов* предоставляют еще один способ создания объекта и инициализации

его полей и свойств. Если используются инициализаторы объектов, то вместо обычного вызова конструктора

класса указываются имена полей или свойств, инициализируемых первоначально задаваемым

значением. Следовательно, синтаксис инициализатора объекта предоставляет

альтернативу явному вызову конструктора класса. Синтаксис инициализатора объекта

используется главным образом при создании анонимных типов в LINQ-выражениях. Но поскольку инициализаторы объектов можно, а иногда и должно использовать в именованном классе,

то ниже представлены основные положения об инициализации объектов. (***glava8\_12***)

class MyClass

{

public int Count;

public string Str;

}

class ObjInitDemo

{

static void Main()

{

MyClass obj = new MyClass { Count = 100, Str = "Test" };

Console.WriteLine(obj.Count + " " + obj.Str);

}

}

Обратите внимание на то, что в классе MyClass отсутствуют явно определяемые конструкторы и не используется обычный синтаксис конструкторов.

MyClass obj = new MyClass { Count = 100, Str = "Test" };

В этой строке кода имена полей указываются явно вместе с их первоначальными

значениями. Это приводит к тому, что сначала конструируется экземпляр объекта

типа MyClass (с помощью неявно вызываемого по умолчанию конструктора), а затем

задаются первоначальные значения переменных Count и Str данного экземпляра.

Инициализаторы объектов обычно не используются в именованных классах, как, например,

в представленном выше классе MyClass, хотя это вполне допустимо. Вообще,

при обращении с именованными классами используется синтаксис вызова обычного

конструктора. И, как упоминалось выше, инициализаторы объектов применяются в

основном в анонимных типах, формируемых в LINQ-выражениях.