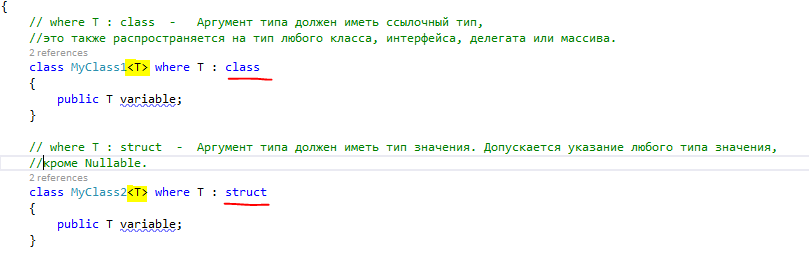
Для всех ссылочных типов с одинаковым Т (например string) будет создаваться один тип (если мы в коде создадим коллекцию какого либо класса и закроем ее string), а для типов значений, будет создаваться постоянно новый тип!!!

Ограничения параметризированных классов (ограничения - Constraint)

В качестве параметра типа должны подставить только ссылочный тип

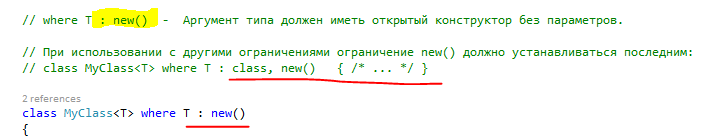
Передаваемый тип должен быть ссылочным

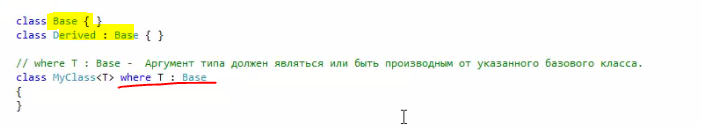
Во втором примере передаваемы тип должен быть структурным (закрыть должны структурой)



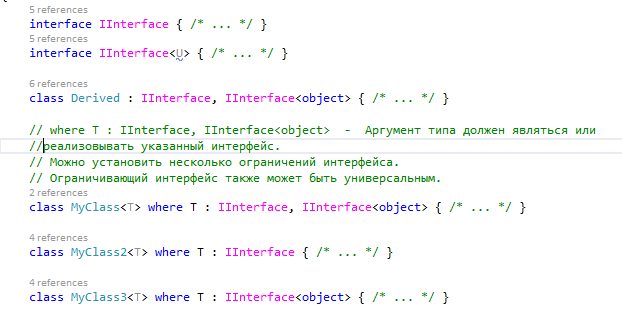
Аргумент типа должен иметь открытый конструктор без параметров

Если в ограничения включены class, struct то new() должен идти в последовательности за ними

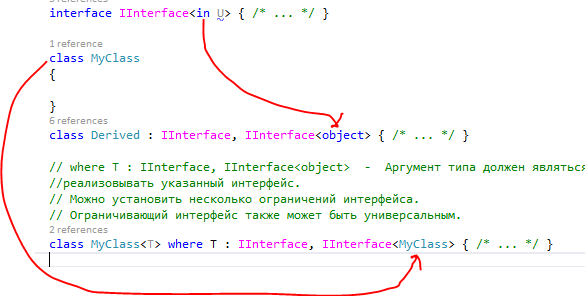


Аргумент типа должен являться указанным типом или быть производны от указанного типа

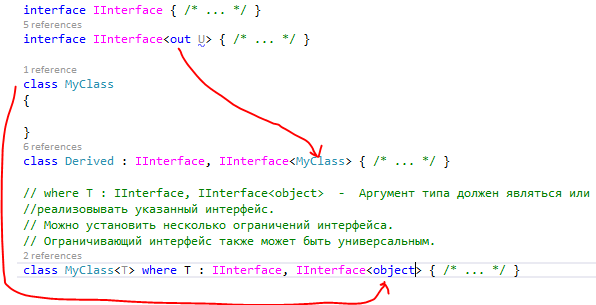
Аргумент типа должен быть интерфейсом или реализовывать интерфейс



Пример контрвариантности



И ковариантности



 В частности, ***ковариантность*** позволяет присвоить делегату метод, возвращаемым типом которого служит класс, производный от класса, указываемого в возвращаемом типе делегата. А***контравариантность*** позволяет присвоить делегату метод, типом параметра которого служит класс, являющийся базовым для класса, указываемого в объявлении делегата.

20 02