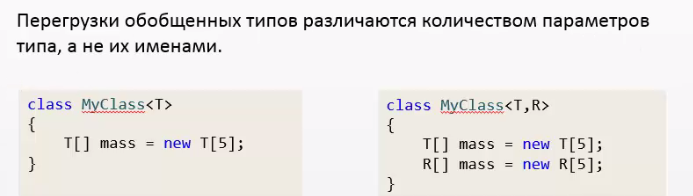
Параметризированный класс (обобщенный класс)

Перегруженные классы



class MyClass<T>

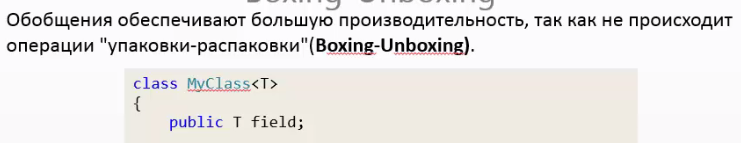
{

}

class MyClass

{

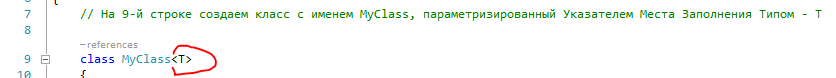
}



Перечисления (Enum) не могут быть параметризированы

А то что передается внутри параметра класса называется места заполнения указателя типа

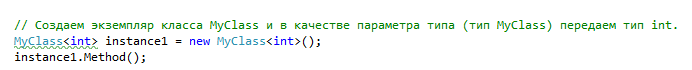
Создаем класс именем MyClass с одним параметризированным указателем места заполнения типом



Правильно два варианта

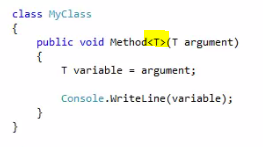
Создаем экземпляр класса MyClass и закрываем его типом int

Создаем экземпляр класса MyClass и в качестве параметра типа передаем тип int

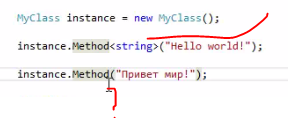


Параметризированный метод

(обобщенный метод)



Два варианта вызова параметризированного метода



Параметризированные делегаты

// На 11-й строке создаем класс-делегата с именем MyDelegate, параметризированный двумя Указателями Места Заполнения Типом - Т и R,

R – возвращаемое значение, (всегда указатель места заполнения типа ставится последним)!!!

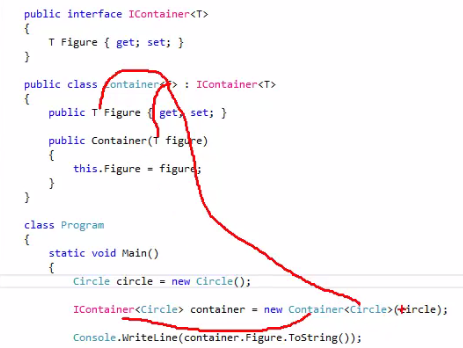


метод который будет сообщен с экземпляром данного класса-делегата, будет принимать один аргумент, типа Указателя Места Заполнения Типом - Т,

и возвращать значение типа Указателя Места Заполнения Типом - R.

Ковариантность обобщений – это апкаст (UpCast) параметра типа

ОБЫЧНОЕ ПРИВЕДЕНИЕ (ковариантности и контрвариантности нет)

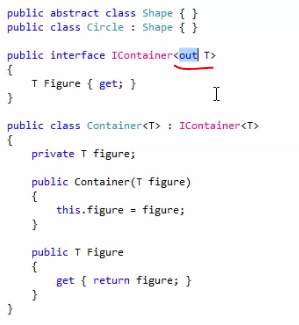


Ковариантность (не явный упкаст (неявное приведение к базовому типу))

Интерфейс в параметризированный указателем мета заполнения типом Т который помечен ключевым словом out (out определяет ковариантность в обобщениях)

В ковариантных параметризированных интерфейсах только может быть указанно только возвращающее значение





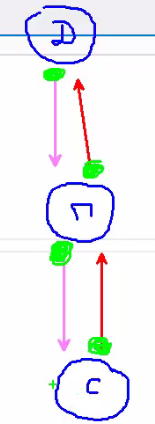
Оба параметра типа должны привестись не явно



Дед, отец, сын

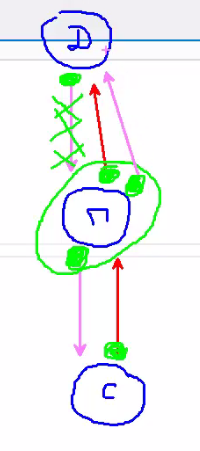
Фиолетовым ковариантность

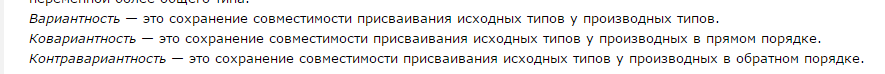
Красным Контрвариантность



Ковариантность и контрвариантность К ОДНОМУ ОБЪЕКТУ одновременно называется – ИНВАРИАНТНОСТЬ

НО объекты параметризированные указателем места заполнения типа Т инвариантными быть не могут



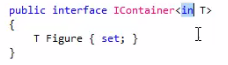


КОНТРВАРИАНТНОСТЬ – даункаст (DownCast) параметра типа

ГЛАВНОЕ ПРАВИЛО ТО ЧТО ДАУНКАСТ НЕ ВОЗМОЖЕН БЕЗ ПРЕДВАРИТЕЛЬНОГО УПКАСТА

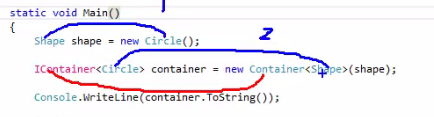
Используется ключевое слово in

И только на запись (на чтение нет)



1-Изначально привели к базовому типу

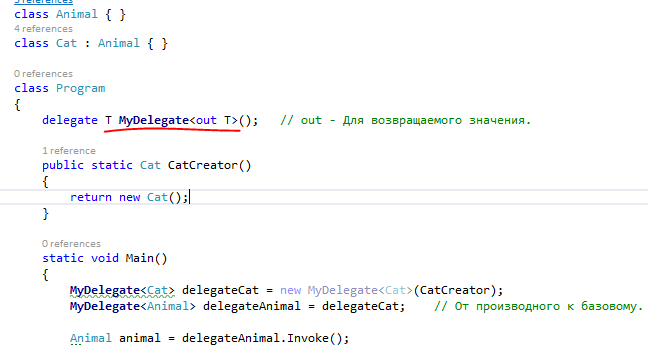
2- А после привели к типу производного класса



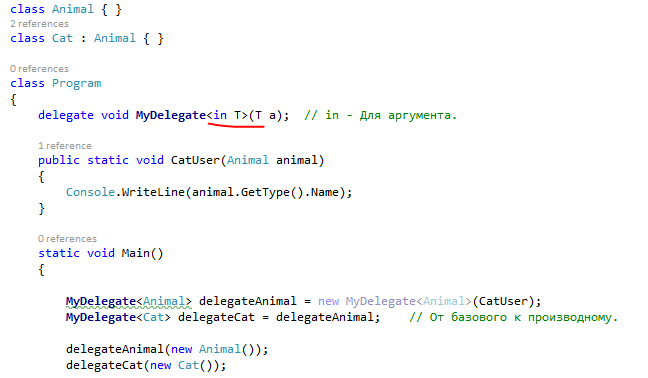
Параметризированные делегаты

Ковариантность и контрвариантность

Out - ковариантность



In – контрвариантность



Частичные методы (partial) не могут иметь out параметров

