Условие:

Создать программу, наглядно демонстрирующую работу с generic-классами, generic-методами и generic-делегатами на языке программирования C#

Исходный код программы:

**using** System;  
  
**namespace** GenericExample  
{  
  
  
 **class** MyGenericClass<T>  
 {  
  
 T outgeneric;  
 T TempData;  
 **public** MyGenericClass(T value)  
 {  
 TempData = default(T);  
 Console.WriteLine(TempData);  
 outgeneric = value;  
 }  
  
 **public void** Notify(**int** i)  
 {  
  
 }  
  
 **public** T genericMethod(T genericParameter)  
 {  
 Console.WriteLine("Параметр со значением по-умолчанию: {0}, значение: {1}", **typeof**(T).ToString(), TempData);  
 Console.WriteLine("Параметр метода: {0}, значение: {1}", **typeof**(T).ToString(), genericParameter);  
 Console.WriteLine("Параметр который инициализировал конструктор: {0}, значение: {1}", **typeof**(T).ToString(), outgeneric);  
  
 **return** outgeneric;  
 }  
  
 }  
  
 **class** Program  
 {  
 **public delegate void** Del<T>(T item);  
  
 **public static void** CHECK(**int** i) { }  
 **public static void** CHECK(**string** i) { }  
  
 **static void** Main(**string**[] args)  
 {  
 Del<**int**> m1 = **new** Del<**int**>(CHECK);  
 Del<**string**> m2 = **new** Del<**string**>(CHECK);  
  
 MyGenericClass<**string**> generic\_obj\_string = **new** MyGenericClass<**string**>("Meow");  
 generic\_obj\_string.genericMethod("MEOW");  
  
 Console.WriteLine("----");  
 MyGenericClass<**bool**> generic\_obj\_boolean = **new** MyGenericClass<**bool**>(true);  
 generic\_obj\_boolean.genericMethod(true);  
  
 Console.WriteLine("----");  
 MyGenericClass<**int**> generic\_obj\_int = **new** MyGenericClass<**int**>(42342);  
 generic\_obj\_int.genericMethod(228123);  
 }  
 }  
  
}

Скриншоты программы:

