

Hoja de trabajo 4

Javier Jose Alvarez Flores, 20171074, alvarez171074@unis.edu.gt

25 de febrero del 2018

1 Explicacion

Las ventajas que existen de utilizar metodos genericos a diferencia de no utilizarlos radican en varias partes, la primera de todas es que al no utilizar tipos genericos yo no puedo utilizar el mismo metodo (en este caso el metodo Head) para poder obtener el primer indice de un array de strings si el metodo solo recibe enteros, tendria que escribir otro metodo Head para poder hacer esta accion, y asi con cada tipo que me interese.

Esta problematica tiene solucion la cual es obtener un arreglo de tipo object ya que de object heredan todos. Pero esto tiene un problema. Al hacer esto yo no puedo controlar que tipo de datos ha sido enviado ni que tipo de dato sera retornado, lo cual se puede ver en el ejemplo brindado, el compilador no presentara problema ya que tanto int como string heredan de object, y al momento de obtener la respuesta necesito castearla al tipo de dato de mi interes, lo cual lanzaria una excepcion debido a que no puedo castear implicitamente un string a un entero, causando un error en tiempo de ejecucion.

Esto se soluciona con los metodos genericos, ya que con ellos puedo controlar el tipo de dato que se envia como parametro y tambien puedo restringir el retorno, tambien evitando la necesidad del casteo de la respuesta ya que conozco el tipo que sera retornado.

```
class Ejemplo
{
    public static object Head(object[] arreglo)
    {
5         return arreglo[0];
    }

    public static T HeadGenerico<T>(T[] arreglo)
    {
10         return arreglo[0];
    }

    public static void Prueba()
    {
15         //Hay un grave error ya que el programa lanzara una excepcion en tiempo
        //de ejecucion al intentar castear una cadena a un entero
        string[] valores = new string[] { "hola", "mundo" };
        int respuesta = (int)Head(valores);

20         //A diferencia del metodo generico, el cual me permite mantener un control
        //del tipo de dato que se obtendra y no hay mas necesidad de seguir
```

```
25      //casteando los objetos
      int[] valoresInt = new int[] { 1, 2, 3, 4 };
      int respuestaint = HeadGenerico(valoresInt);
    }
}
```