

## 9.A. Introducción a las estructuras de almacenamiento.

### 3. Clases y métodos genéricos (II).

¿Crees que el código es más legible al utilizar genéricos o que se complica? La verdad es que al principio cuesta, pero después, el código se entiende mejor que si se empieza a insertar conversiones de tipo.

Las clases genéricas son equivalentes a los métodos genéricos pero a nivel de clase, permiten definir un parámetro de tipo o genérico que se podrá usar a lo largo de toda la clase, facilitando así crear clases genéricas que son capaces de trabajar con diferentes tipos de datos base.

Para crear una clase genérica se especifican los parámetros de tipo al lado del nombre de la clase:

```
public class Util<T> {  
  
    T t1;  
  
    public void invertir(T[] array) {  
  
        for (int i = 0; i < array.length / 2; i++) {  
  
            t1 = array[i];  
  
            array[i] = array[array.length - i - 1];  
  
            array[array.length - i - 1] = t1;  
  
        }  
  
    }  
  
}
```

En el ejemplo anterior, la clase `Util` contiene el método `invertir` cuya función es invertir el orden de los elementos de cualquier array, sea del tipo que sea. Para usar esa clase genérica hay que crear un objeto o instancia de dicha clase especificando el tipo base entre los símbolos menor que ("`<`") y mayor que ("`>`"), justo detrás del nombre de la clase. Veamos un ejemplo:

```
Integer[] numeros={0,1,2,3,4,5,6,7,8,9};  
  
Util<Integer> u= new Util<Integer>();  
  
u.invertir(numeros);  
  
for (int i=0;i<numeros.length;i++)  
  
    System.out.println(numeros[i]);
```

Como puedes observar, el uso de genéricos es sencillo, tanto a nivel de clase como a nivel de método. Simplemente, a la hora de crear una instancia de una clase genérica, hay que especificar el tipo, tanto en la definición (`Util <integer> u`) como en la creación (`new Util<Integer>()`).

Los genéricos los vamos a usar ampliamente a partir de ahora, aplicados a un montón de clases genéricas que tiene Java y que son de gran utilidad, por lo que es conveniente que aprendas bien a usar una clase genérica.

Los parámetros de tipo de las clases genéricas solo pueden ser clases, no pueden ser jamás tipos de datos primitivos como `int`, `short`, `double`, etc. En su lugar, debemos usar sus clases envoltorio `Integer`, `Short`, `Double`, etc.

#### Autoevaluación

Dada la siguiente clase, donde el código del método `prueba` carece de importancia, ¿podrías decir cuál de las siguientes invocaciones es la correcta?

```
public class Util {  
  
    public static <T> int prueba (T t) { ... }  
  
};
```

- ☐ `Util.<int>prueba(4);`
- ☐ `Util.<Integer>prueba(new Integer(4));`
- ☐ `Util u=new Util(); u.<int>prueba(4);`

