JAVA

GENERICS

Emiliano Loya Flores

Programación Orientada a Objetos

Genéricos en Java

Las clases y los métodos genéricos permiten con una sola declaración del método o de la clase especificar el tipo. También proporcionan seguridad en tiempo de compilación ya que permiten al desarrollador detectar tipos no validos en tiempo de compilación.

Métodos Genéricos

Se puede declarar un método genérico que se puede llamar con argumentos de distintos tipos, los cuales solo pueden ser referencias, no pueden ser tipos primitivos.

```java
public class Test {
    public static <T> void printArray(T[] typeT) {
        for (T type : typeT) {
            System.out.println(" " + type);
        }
    }
    public static void main(String args[]) throws Exception {
        Integer[] intArray = { 1, 2, 3 };
        printArray(intArray);
    }
}
```

Clases e Interfaces Genéricas

Así como los métodos, las clases e interfaces pueden usar genericos.

- Se pueden usar <> vacíos para invocar al constructor de la clase.
- Se pueden usar mas de un tipo de parametro:

Example<K, V>

- Los parametros deben ser referencias y no primitivos.
- Se pueden utilizar como parametros otros tipos de parametros:

Example<V, Box<Integer>>

```
class Simple<T> {
    T type;

    public Simple(T typeT) {
        this.type = typeT;
    }
}
public class Test {
    public static void main(String args[]) throws Exception {
        Simple<Integer> simple = new Simple<>(10);
    }
}
```

Tipos de parametros acotados

Se pueden restringir los tipos de parametros que se pasan al método, a la clase o interfaz, para esto se usan los tipos de parámetros acotados.

Para hacer esto se coloca el nombre del tipo del parámetro seguido de la palabra reservada "extends" y después la acotación superior. En este contexto, "extends" tiene un uso general para referirse a "extends" como una clase e "implements" como en una interfaz

Tipos de parametros acotados

Por ejemplo, es posible que un método que funciona con números solo quiera aceptar instancias de Número o sus subclases.

```java
public class Test {
    public static <T extends Number> void printArray(T[] typeT) {
        for (T type : typeT) {
            System.out.println(" " + type);
        }
    }
    public static void main(String args[]) throws Exception {
        Integer[] intArray = { 1, 2, 3 };
        printArray(intArray);
    }
}
```

Tipos de parametros acotados

Puede suceder lo mismo en una clase o interfaz que con el método anterior.

```
class classExample<T extends Number> {
}

interface interfaceExample<T extends Number> {
}
```

Convenciones de nombres de parámetros

Por convención, los nombres de los parámetros son singulares y en mayúsculas.

Los nombre más comunes para los tipos de parámetros son:

- E: Elemento (Usado en las colecciones del framework de Java).
- K: Key.
- N: Número.
- T: Tipo.
- V: Valor.
- S, U, V, etc: 2do, 3er, 4to ... tipos.

Ejercicio:

Crear una clase generica llamada "GenericLinkedList" que contenga lo siguiente:

- Atributos:
  - genericList: LinkedList con el generico de la clase.
- Constructor:
  - Creará una instancia de genericList.
- Métodos:
  - void Stack: Añadirá un elemento al final de la lista
  - void showGenericList: Imprimirá los elementos de la lista.
  - main: se creara una instancia de GenericLinkedList con el tipo que se desee, se añadirán 3 elementos a la lista y se mostraran los elementos con el método showGenericList.