Tipos genéricos en Java

TIPO DE DATO GENÉRICO

Un tipo de dato genérico, es una forma de crear un tipo de dato compuesto en el que el tipo de uno o varios de sus atributos se pueden establecer en tiempo de compilación.

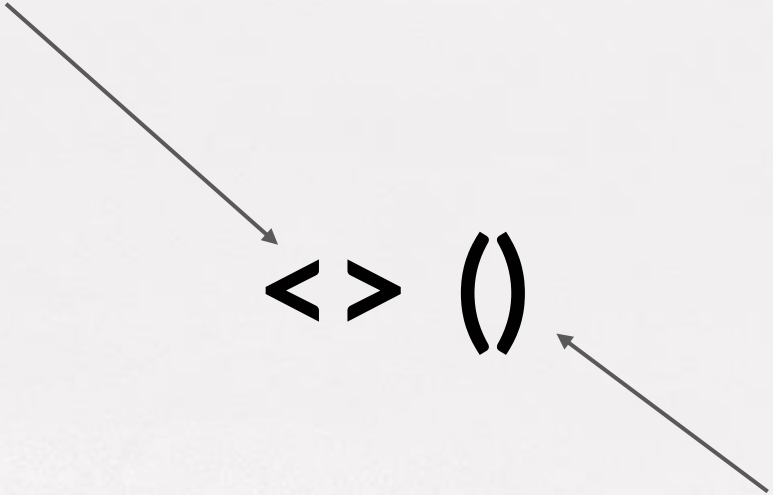
Es una tipo de dato en el que uno o varios de sus atributos se pueden parametrizar.

ArrayList<>

ArrayList<String>

ArrayList<Integer>

Parametrizamos tipos



Parametrizamos valores

EJERCICIO

Implementa una Lista con los métodos add, *delete*, *get* y *size* pero que cuando hagamos *get* nos borre el elemento de la lista, y queremos que la lista solo sea para Integers

```
public class Lista {
   private ArrayList<Integer> lista = new ArrayList<Integer>();

   public Lista() {}

   public Integer get(int index) {
                return lista.remove(index);
   }

   public void delete(int index) {
                lista.remove(index);
   }

   public void add(Integer integer) {
                lista.add(integer);
   }

   public int size() {
                return lista.size();
   }
}
```

EJERCICIO

Implementa la lista anterior para cualquier Clase que pueda existir.

```
public class ListaGenerica <T>{

    private ArrayList<T> lista = new ArrayList<T>();

    public ListaGenerica() {}

    public T get(int index) {
                return lista.remove(index);
    }

    public void delete(int index) {
                lista.remove(index);
    }

    public void add(T object) {
                lista.add(integer);
    }

    public int size() {
                return lista.size();
    }

}
```

ListaGenerica<Integer>

ListaGenerica<String>

```
public class Pareja<K,V> {

    private K key;
    private V value;

    public Pareja(K key, V value) {
                this.key = key;
                this.value = value;
    }

    public K getKey() {
                return key;
    }

    public void setKey(K key) {
                this.key = key;
    }
    public V getValue() {
                return value;
    }
    public void setValue(V value) {
                this.value = value;
    }

}
```

```
public class Pareja<K, V extends Usuarios>{
```

CONVENCIONES DE NOMBRES

E - Element (Usado extensivamente en la Java Collections Framework)
K - Key
N - Number
T - Type
V - Value
S, U, V etc - 2º, 3º 4º tipos

¡GRACIAS!

¿ALGUNA PREGUNTA?