

Ejercicio 1: Clase genérica Par

Enunciado:

Crea una clase genérica `Par<T, U>` que almacene dos valores (un par de datos), uno de tipo `T` y otro de tipo `U`.

Implementa los métodos `getPrimero()` y `getSegundo()` para obtener cada valor del par, y un método `toString()` que devuelva una representación de ambos valores.

Ejercicio 2: Clase genérica Par con métodos de acceso

Enunciado:

Crea una clase genérica `Par<T, U>` que almacene dos objetos de tipos `T` y `U` (atributos). Implementa métodos `getPrimero()` y `getSegundo()` para obtener cada valor y un método `toString()` para devolver una representación de ambos valores.

En el main, crea un `Par` que almacene un `Integer` y un `String`, y otro `Par` que almacene un `Double` y un `Character`.

```
System.out.println(par1); // Salida: Par [primero=100, segundo=Cien]
System.out.println(par2); // Salida: Par [primero=3.14, segundo=π]
```

Ejercicio 3: Método genérico compararElementos

Enunciado:

Crea una clase `Utilidades` que contenga un **método** genérico `compararElementos(T elemento1, T elemento2)` que reciba dos elementos de tipo `T` y devuelva `true` si ambos elementos son iguales (`elemento1.equals(elemento2)`) y `false` en caso contrario.

En el main, prueba el método `compararElementos` con diferentes tipos de datos, como `String` e `Integer`.