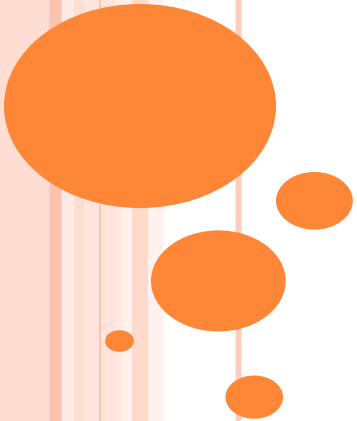
Interfaces funcionales Java 8



Introducción

- Conjunto de interfaces funcionales incorporadas en Java SE 8 dentro del paquete java.util.function.
- Utilizadas en contextos de manipulación de datos a través de colecciones y streams.
- Interfaces principales:
  - Predicate. Establecimiento de condiciones
  - Function. Transformación de datos y cálculos
  - Consumer. Procesamiento de datos
  - Supplier. Generación de datos

Interfaz Predicate<T>

- **Dispone del método abstracto *test*, que a partir de un objeto realiza una comprobación y devuelve un boolean:**

boolean test(T t)

- **Ejemplo:**

```
//implementación que comprueba una cadena
//y devuelve true si su longitud es mayor que 10
Predicate<String> pr=a->a.length()>10;
```

```
//uso
if(pr.test("hola")){
   System.out.println("aceptada");
}else{
   System.out.println("no aceptada");
}
```

- **Variante BiPredicate<T,U> con dos parámetros:**

boolean test(T t, U u)

Interfaz Function<T,R>

- **Método abstracto *apply*, que a partir de un objeto realiza una operación y devuelve un resultado:**

R apply(T t)

- **Ejemplo:**

```
//implementación que transforma
//un String en un Integer
Function<String, Integer> fr=(String a)->Integer.getInteger(a);
```

```
//uso
Integer num=fr.apply("45");
```

- **Variante BiFunction<T,U,R> con dos parámetros:**

R apply(T t, U u)

Interfaz Consumer<T>

- **Dispone del método abstracto *accept*, que realiza algún tipo de procesamiento con el objeto recibido:**

```
void accept(T t)
```

- **Ejemplo:**

```
//Muestra el dato recibido
Consumer<String> cr=(String a)->System.out.println(a);

//uso
cr.accept("hola");
```

- **Variante BiConsumer<T,U> con dos parámetros:**

```
void accept(T t, U u)
```

Interfaz Supplier<T>

- **Dispone del método abstracto *get*, que no recibe ningún parámetro y devuelve como resultado un objeto:**

```
T get()
```

- **Ejemplo:**

```
//simula la obtención de una conexión
//con una base de datos
Supplier<Connection> sp=()->{
    String cad="cadena conexión BD";
    String user="usuario";
    String pwd="password";
    Connection cn;
     cn=DriverManager.getConnection(cad,user,pwd);
    return cn;
}
```

```
//uso
Connection cn=sp.get();
```

Interfaz UnaryOperator<T>

- **Subinterfaz de Function donde el tipo de entrada coincide con el de devolución:**

  T apply(T t)

- **Equivale a Function<T,T>**

- **Variante BinaryOperator<T> equivalente a BIFunction<T,T,T>**

Interfaces para tipos primitivos

- **IntPredicate** ⟹ boolean test(int t)
- **IntFunction<R>** ⟹ R apply(int t)
- **IntConsumer** ⟹ void accept(int t)
- **IntSupplier** ⟹ int getAsInt()
- **IntUnaryOperator** ⟹ int applyAsInt(int t)
- **Versiones también para long y double**