JDK FUNCTIONAL INTERFACES

Package java.util.function

Interface Summary

Interface	Description
BiConsumer<T,U>	Represents an operation that accepts two
BiFunction<T,U,R>	Represents a function that accepts two ar
BinaryOperator<T>	Represents an operation upon two operar
BiPredicate<T,U>	Represents a predicate (boolean-valued fu
BooleanSupplier	Represents a supplier of boolean-valued
Consumer<T>	Represents an operation that accepts a si
DoubleBinaryOperator	Represents an operation upon two double
DoubleConsumer	Represents an operation that accepts a si
DoubleFunction<R>	Represents a function that accepts a doul
DoublePredicate	Represents a predicate (boolean-valued fu

• • •

+40 @FunctionalInterface

Interface Function<T,R>

$$f(x) \rightarrow y$$

```
R apply(T t);
```

```
Function<Persona, String> nombre = per -> per.getNombre();
```

Composiciones con Function

andThen : añadir funcionalidad posterior

```
Function<Persona, String> nombre = per -> per.getNombre();
nombre = nombre.andThen(it -> it.toUpperCase());
```

compose : añadir funcionalidad anterior

```
Function<Coche, Persona> propietario = it -> it.getPropietario();
Function<Persona, String> nombre = per -> per.getNombre();

Function<Coche, String> nombrePropietario = nombre.compose(propietario);
```

Interface UnaryOperator<T>

$$f:T \rightarrow T$$

Además: interfaces para el caso específico de T primitivo

int, long, double

IntUnaryOperator
LongUnaryOperator
DoubleUnaryOperator

Interface Function<T,R>

primitivo = int, long o double

f (primitivo) -> R	IntFunction<R>
f (T) -> primitivo	ToLongFunction<T>
f (primitivo) -> primitivo	DoubleToIntFunction

Consumer<T>

```
Consumer<String> impresor = (it) -> {System.out.println(it);}
```

Supplier<T>

```
Random random = new Random();
Supplier<Integer> generador = () -> random.nextInt();
```

Predicate<T>

f(x) → boolean

```
Predicate<String> cadenaCorta = it -> it.length() < 10;
```

IntPredicate
LongPredicate
DoublePredicate

Binary *

BinaryOperator<T>	f(t,t) → t
BiFunction<T,U,R>	f(t,u) → r
BiPredicate<T,U>	f(t,u) → boolean
DoubleBinaryOperator	f(double,double) → double

Package java.util.function