Interfaces Funcionales - java.util.function

Interface	Método	Description
Predicate<T>	boolean test(T)	Represents a predicate (boolean-valued function) of one argument.
BiPredicate<T,U>	test(T,U)	Represents a predicate (boolean-valued function) of two arguments.
IntPredicate	test(int)	Represents a predicate (boolean-valued function) of one int-valued argument.
DoublePredicate	test(double)	Represents a predicate (boolean-valued function) of one double-valued argument.
LongPredicate	test(long)	Represents a predicate (boolean-valued function) of one long-valued argument.
Consumer<T>	void accept(T)	Represents an operation that accepts a single input argument and returns no result.
BiConsumer<T,U>	accept(T,U)	Represents an operation that accepts two input arguments and returns no result.
IntConsumer	accept(int)	Represents an operation that accepts a single int-valued argument and returns no result.
DoubleConsumer	accept(double)	Represents an operation that accepts a single double-valued argument and returns no result.
LongConsumer	accept(long)	Represents an operation that accepts a single long-valued argument and returns no result.
ObjIntConsumer<T>	accept(T,int)	Represents an operation that accepts an object-valued and a int-valued argument, and returns no result.
ObjDoubleConsumer<T>	accept(T,double)	Represents an operation that accepts an object-valued and a double-valued argument, and returns no result.
ObjLongConsumer<T>	accept(T,long)	Represents an operation that accepts an object-valued and a long-valued argument, and returns no result.
Function<T,R>	R apply(T)	Represents a function that accepts one argument and produces a result.
BiFunction<T,U,R>	R apply(T,U)	Represents a function that accepts two arguments and produces a result.
IntFunction<R>	R apply(int)	Represents a function that accepts an int-valued argument and produces a result.
DoubleFunction<R>	R apply(double)	Represents a function that accepts a double-valued argument and produces a result.
LongFunction<R>	R apply(long)	Represents a function that accepts a long-valued argument and produces a result.
IntToDoubleFunction	dou apply(int)	Represents a function that accepts an int-valued argument and produces a double-valued result.
IntToLongFunction	long apply(int)	Represents a function that accepts an int-valued argument and produces a long-valued result.
DoubleToIntFunction		Represents a function that accepts a double-valued argument and produces an int-valued result.
DoubleToLongFunction		Represents a function that accepts a double-valued argument and produces a long-valued result.
LongToDoubleFunction		Represents a function that accepts a long-valued argument and produces a double-valued result.
LongToIntFunction		Represents a function that accepts a long-valued argument and produces an int-valued result.
ToIntFunction<T>	int apply(T)	Represents a function that produces an int-valued result.
ToDoubleFunction<T>		Represents a function that produces a double-valued result.
ToLongFunction<T>		Represents a function that produces a long-valued result.
ToIntBiFunction<T,U>	int apply(T,U)	Represents a function that accepts two arguments and produces an int-valued result.
ToDoubleBiFunction<T,U>		Represents a function that accepts two arguments and produces a double-valued result.
ToLongBiFunction<T,U>		Represents a function that accepts two arguments and produces a long-valued result.
BinaryOperator<T>	T apply(T,T)	Represents an operation upon two operands of the same type, producing a result of the same type.
IntBinaryOperator	int apply(int,int)	Represents an operation upon two int-valued operands and producing an int-valued result.
DoubleBinaryOperator		Represents an operation upon two double-valued operands and producing a double-valued result.
LongBinaryOperator		Represents an operation upon two long-valued operands and producing a long-valued result.
UnaryOperator<T>	T apply(T)	Represents an operation on a single operand that produces a result of the same type as its operand.
IntUnaryOperator	int apply(int)	Represents an operation on a single int-valued operand that produces an int-valued result.
DoubleUnaryOperator		Represents an operation on a single double-valued operand that produces a double-valued result.
LongUnaryOperator		Represents an operation on a single long-valued operand that produces a long-valued result.
Supplier<T>	T get()	Represents a supplier of results.
IntSupplier	int get()	Represents a supplier of int-valued results.
DoubleSupplier	double get()	Represents a supplier of double-valued results.
LongSupplier	long get()	Represents a supplier of long-valued results.
BooleanSupplier	boolean get()	Represents a supplier of boolean-valued results.