

# java.util.function Interfaces Cheat Sheet

Classification		Interface	Single Abstract Method (SAM)
<b>Consumer</b>		BiConsumer<T,U>	<b>void</b> accept(T t, U u)
		Consumer<T>	<b>void</b> accept(T t)
		DoubleConsumer	<b>void</b> accept( <b>double</b> value)
		IntConsumer	<b>void</b> accept( <b>int</b> value)
		LongConsumer	<b>void</b> accept( <b>long</b> value)
		ObjDoubleConsumer<T>	<b>void</b> accept(T t, <b>double</b> value)
		ObjIntConsumer<T>	<b>void</b> accept(T t, <b>int</b> value)
		ObjLongConsumer<T>	<b>void</b> accept(T t, <b>long</b> value)
<b>Function</b>	Object(s) to Object	BiFunction<T,U,R>	<b>R</b> apply(T t, U u)
		Function<T,R>	<b>R</b> apply(T t)
	Object(s) to Primitive	ToDoubleBiFunction<T,U>	<b>double</b> applyAsDouble(T t, U u)
		ToDoubleFunction<T>	<b>double</b> applyAsDouble(T value)
		ToIntBiFunction<T,U>	<b>int</b> applyAsInt(T t, U u)
		ToIntFunction<T>	<b>int</b> applyAsInt(T value)
		ToLongBiFunction<T,U>	<b>long</b> applyAsLong(T t, U u)
		ToLongFunction<T>	<b>long</b> applyAsLong(T value)
	Primitive to Object	DoubleFunction<R>	<b>R</b> apply( <b>double</b> value)
		IntFunction<R>	<b>R</b> apply( <b>int</b> value)
		LongFunction<R>	<b>R</b> apply( <b>long</b> value)
	Primitive to Primitive	DoubleToIntFunction	<b>int</b> applyAsInt( <b>double</b> value)
		DoubleToLongFunction	<b>long</b> applyAsLong( <b>double</b> value)
		IntToDoubleFunction	<b>double</b> applyAsDouble( <b>int</b> value)
		IntToLongFunction	<b>long</b> applyAsLong( <b>int</b> value)
		LongToDoubleFunction	<b>double</b> applyAsDouble( <b>long</b> value)
		LongToIntFunction	<b>int</b> applyAsInt( <b>long</b> value)
	<b>Operator</b>	<b>Binary</b>	BinaryOperator<T>
DoubleBinaryOperator			<b>double</b> applyAsDouble( <b>double</b> left, <b>double</b> right)
IntBinaryOperator			<b>int</b> applyAsInt( <b>int</b> left, <b>int</b> right)
LongBinaryOperator			<b>long</b> applyAsLong( <b>long</b> left, <b>long</b> right)
<b>Unary</b>		DoubleUnaryOperator	<b>double</b> applyAsDouble( <b>double</b> operand)
		IntUnaryOperator	<b>int</b> applyAsInt( <b>int</b> operand)
		LongUnaryOperator	<b>long</b> applyAsLong( <b>long</b> operand)
		UnaryOperator<T>	<b>T</b> apply(T t)
<b>Predicate</b>		BiPredicate<T,U>	<b>boolean</b> test(T t, U u)
		DoublePredicate	<b>boolean</b> test( <b>double</b> value)
		IntPredicate	<b>boolean</b> test( <b>int</b> value)
		LongPredicate	<b>boolean</b> test( <b>long</b> value)
		Predicate<T>	<b>boolean</b> test(T t)
<b>Supplier</b>		BooleanSupplier	<b>boolean</b> getAsBoolean()
		DoubleSupplier	<b>double</b> getAsDouble()
		IntSupplier	<b>int</b> getAsInt()
		LongSupplier	<b>long</b> getAsLong()
		Supplier<T>	<b>T</b> get()