# JAVA FUNCTIONAL PROGRAMMING

<https://www.youtube.com/watch?v=rPSL1alFIjI>

**Declarative programming** focuses on specifying the desired outcome or result of a program without explicitly detailing the steps or instructions required to achieve that outcome.

**Imperative programming**, on the other hand, focuses on explicitly describing the steps or instructions that a program should follow to achieve a desired outcome.

## Streams

**In Java, a stream is a sequence of objects that supports various methods which can be pipelined to produce the desired result.** It is not a data structure, but rather a way to express and process collections of objects. Streams are used to perform operations like filtering, mapping, reducing, and sorting on collections without altering the original data structure. They are created from collections, arrays, or I/O channels and support functional-style operations on elements. Streams are widely used in Java for processing and manipulating data efficiently.

## [Method reference (::) A computer screen shot of a program Description automatically generated](https://www.geeksforgeeks.org/stream-in-java/" \t "_blank)

## Package java.util.function

<https://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html>

A screenshot of a computer

Description automatically generated

This package contains functonal interfaces.

**A functional interface has a single abstract method.**

It can have multiple default method.

A screenshot of a computer program

Description automatically generated

## Interface Function<T,R>

A black text on a white background

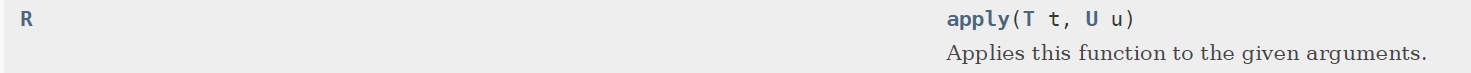
Description automatically generated



## Interface BiFunction<T,U,R>

A close up of black text

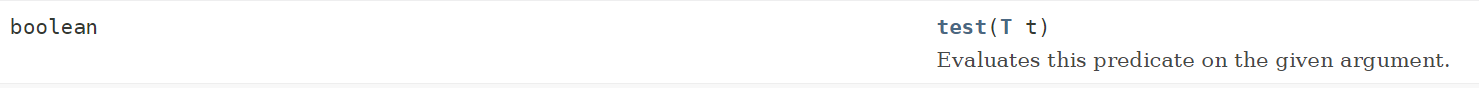
Description automatically generated



## Interface Predicate<T>

A black text on a white background

Description automatically generated



## Interface Consumer<T>

A black and white text

Description automatically generated



## Interface BiConsumer<T,U>

A close up of words

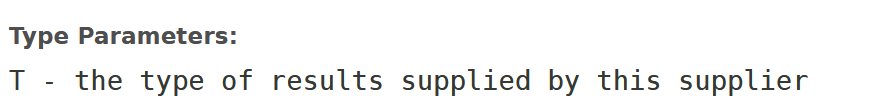
Description automatically generated

A black text on a white background

Description automatically generated

Si pronuncia “baiconsumer”

## Interface Supplier





## Streams

A screenshot of a computer program

Description automatically generated

## Optionals

A screen shot of a computer program

Description automatically generated

ofNullable() methods returns a description of the value if not null and an empty Optional if null.

## Combinator pattern

The combinator pattern is a functional pattern used to organize libraries around the idea of combining simple elements to create more complex structures.

A computer screen shot of a program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

## Callbacks

A screenshot of a computer program

Description automatically generated

## Lambdas

A computer screen shot of a program code

Description automatically generated

**2.11**