Functional Interface

Functional interface is an interface have only one abstract method and number of non-abstract methods.

Non –abstract methods like:

Default method

Static method

Private static/non static methods

Public static methods

About @Functional Interface:

E.g.: You have written one functional interface in your project and did not define that interface as @FunctionalInterface.

After one developer came and added one more abstract method. Then entire concept of functional interface changed. And in lambda expression code show s error.

To avoid this issue we need to define functional interface class with @FunctionalInterface.

It is introduced in Java 8.

Lambda expressions can be used at only using Functional interface concept.

Interface Examples:

Predicate

**boolean** test (T t);

It is a functional interface takes a single arguments and return true or false.

Consumer

**void** accept (**int** value);

It is a functional interface takes single arguments and does not return any result.

Function

T R

Function<String, String>

It takes an argument of type T and returns a value of type R.

Supplier

T get ();

It doesn’t take any arguments and returns a value of type T.

















