**Default Method**

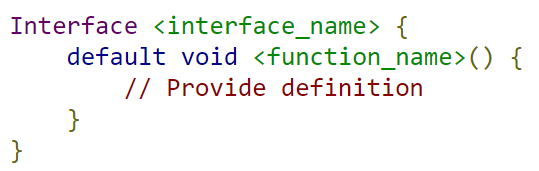
Java 8 introduces a new technique where it allows us to write implementation of default method in interfaces.

Single interface can have any number of default method implementation.

It provides us the backward compatibility to add some functionality for existing interfaces without changing existing logic of classes implementation the same interface.

For example, if we have an interface and same is implementing by multiple classes. After adding a default method or methods will not force us to update the existing classes implementing the same interface.

**Syntax:**



1. What is Functional Interface?

An Interface that contains only one abstract method is known as functional interface. It can have any number of default and static methods. It can also declare methods of object class. It can have any number of default and static methods. It can also declare methods of object class.

Functional interfaces are also known as Single Abstract Method Interfaces (SAM Interfaces). A functional interface can extend another interface only when it does not have any abstract method.

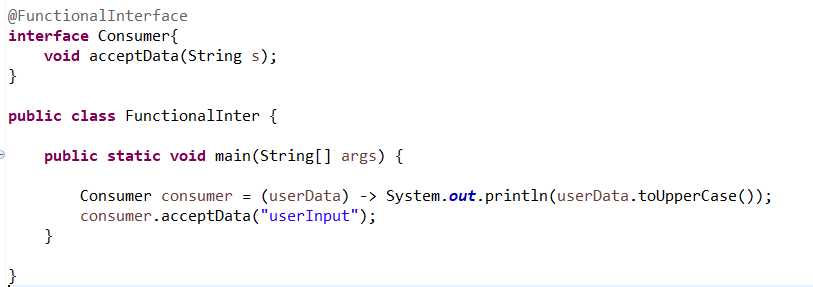
2. Which are the existing Functional Interfaces before Java 8?

1. Runnable
2. ActionListener
3. Comparable

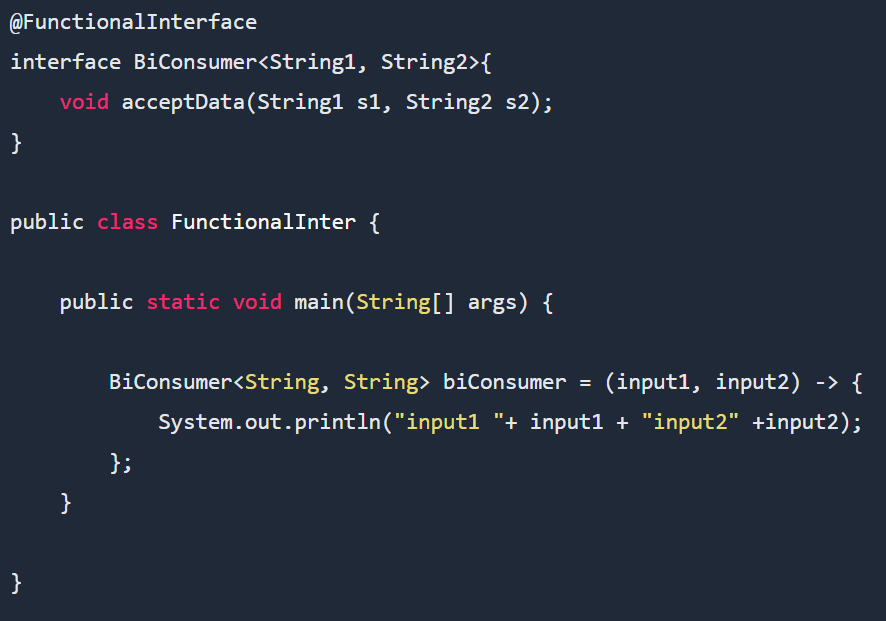
3. Name and explain the Java 8 Predefined Functional Interface?

There are 4 main functional interface which could be used in different scenarios.

1. Consumer: It represents an operation that accepts a single argument and returns no result

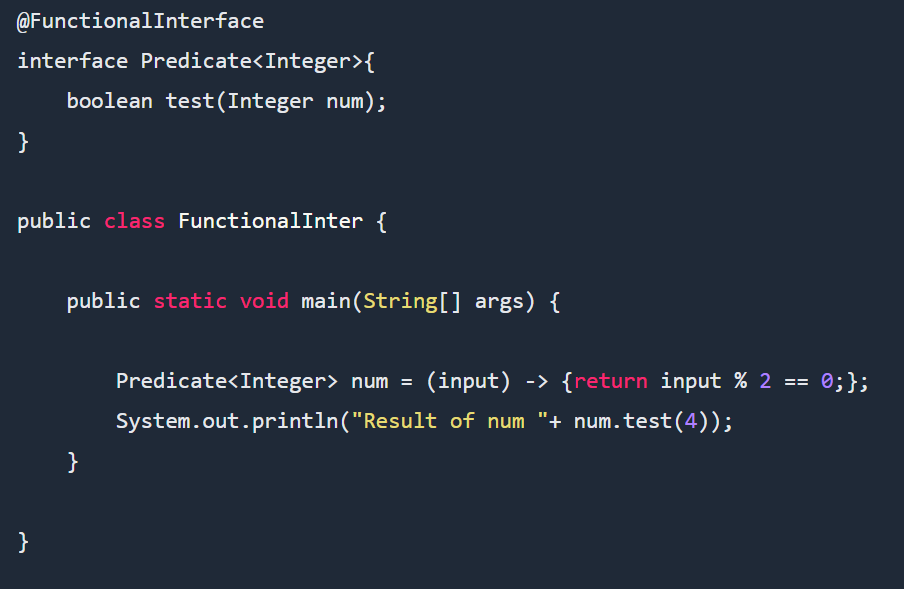


a. BiConsumer: It represents an operation that accepts two input arguments and returns no result. It is a subtype of Consumer.



2. Predicate

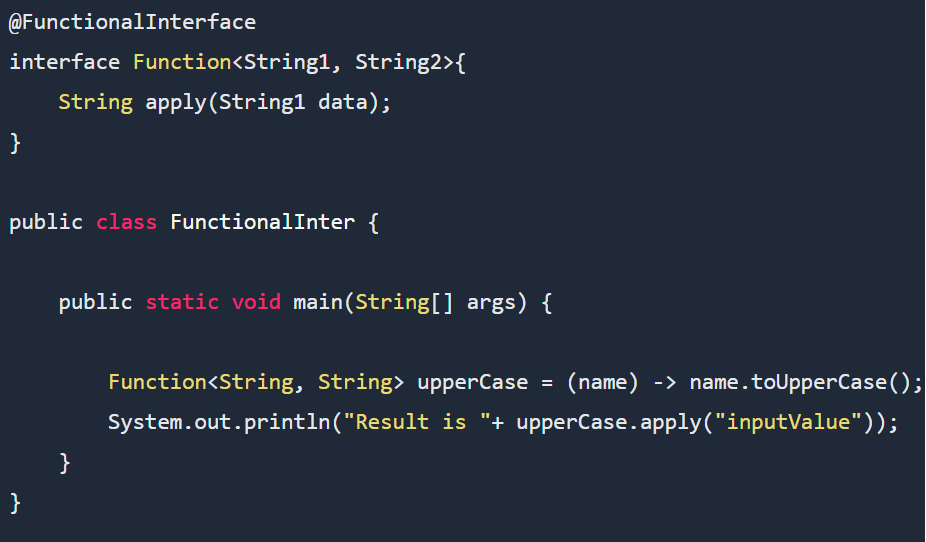
Predicate will accept one argument, do some processing, and then return boolean



a. BiPredicate Instead of one argument, BiPredicate will accept two arguments and return boolean. And BiPredicate is a subtype of Predicate.

3. Function

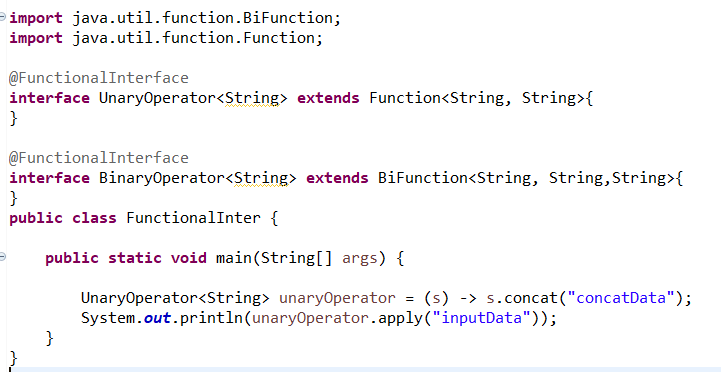
This interface accepts one argument and returns a value after the required processing. It is defined as below. The required processing logic will be executed on the invocation of the apply method.



a. BiFunction : The BiFunction is similar to Function except it accepts two inputs, whereas Function accepts one argument.

b. UnaryOperator and BinaryOperator

UnaryOperator and BinaryOperator,which extends the Function and BiFunction respectively.



From the above interfaces, it is easy to understand that the UnaryOperator accepts a single argument and return a single argument, but both the input and output argument should be of same or similar type.

On the other hand, BinaryOperator accepts two arguments and returns one argument similar to BiFunction, but the type of all the input and output argument should be of similar type.

4. Supplier: Supplier functional interface does not accept any input; rather returns a single output. The following interface code is given for understanding.

