ObjIntConsumer Functional Interface

ObjIntConsumer represents an operation that accepts an argument of type T or object-valued argument and int valued argument and returns no result.

This is the specialization of [BiConsumer](http://data-structure-learning.blogspot.com/2015/07/java-lambda-biconsumer-functional.html) interface. We have discussed it previously. BiConsumer interface represents two input (object) arguments and returns no result. BiConsumer is two-arity specialization of [Consumer](http://data-structure-learning.blogspot.com/2015/07/java-lambda-consumer-functional.html) interface.

*accept()* method

**void** accept(T t, **int** value);

performs this operation on given arguments. Let us take simple example to accept String object and int value.

ObjIntConsumer<String> objIntConsumer = (**final** String t, **final** **int** a) -> {

String result = t.concat(String.*valueOf*(a));

System.***out***.println(result); //Outputs Number is 10

};

objIntConsumer.accept("Number is ", 10);

That’s all on ObjIntConsumer Interface.

Read about important java.util.function package’s interface [here](http://data-structure-learning.blogspot.com/p/functional-programming-in-java.html). [Consumer](http://data-structure-learning.blogspot.com/2015/07/java-lambda-consumer-functional.html), [Function](http://data-structure-learning.blogspot.com/2015/07/java-lambda-function-functional.html), [Supplier](http://data-structure-learning.blogspot.com/2015/07/java-lambda-supplier-functional.html), [BinaryOperator](http://data-structure-learning.blogspot.com/2015/07/java-lambda-binaryoperator-functional.html) & [Predicate](http://data-structure-learning.blogspot.com/2015/07/java-lambda-predicate-functional.html) Functional Interfaces. I have also written on [High Order functions](http://data-structure-learning.blogspot.com/2015/07/higher-order-functions-using-function.html) using Function functional interface.