DoubleConsumer Functional Interface

We discussed [Consumer](http://data-structure-learning.blogspot.com/2015/07/java-lambda-consumer-functional.html) interface previously. I would highly recommend you to read about Consumer interface before reading DoubleConsumer. Consumer interface has a generic type T while DoubleConsumer is a specialized version of Consumer for double.

DoubleConsumer has 2 methods of which one is abstract and another is andThen() method. andThen() method is default method.

|  |
| --- |
| **accept() method**  **void** accept(**double** value);  This method is used to perform this operation on given argument. Let us take a simple example to print the square of a number.  DoubleConsumer power = (x) -> {  **double** result = Math.*pow*(x, 2);  System.***out***.println("Power Consumer " + result); //Outputs 36  };  power.accept(6); |
| **andThen() method**  **default** DoubleConsumer andThen(DoubleConsumer after) {  Objects.*requireNonNull*(after);  **return** (**double** t) -> { accept(t); after.accept(t); };  }  This method returns the composed DoubleConsumer with *this* and *after* DoubleConsumer. Both of this DoubleConsumer will be executed in sequence. If *after* DoubleConsumer is null then it will throw NullPointerException.  Let us take simple example of 2 different DoubleConsumer of which one will work as power and another will work as adder.  DoubleConsumer power = (x) -> {  **double** result = Math.*pow*(x, 2);  System.***out***.println("Power Consumer " + result);  };  DoubleConsumer adder = (x) -> {  x = x + 10;  System.***out***.println("Adder Consumer " + x);  };  power.andThen(adder).accept(10);  Output:  Power Consumer 100.0  Adder Consumer 20.0 |

That’s all on DoubleConsumer interface.