December, 8th 2015

Benjamin Fanjeaux – Paul Lemee

ECE Paris

Software Quality

Unit tests

# Question 1

See code

# Question 2

See code

# Question 3

If we were to implement the Stack Class differently, we would use inheritance.

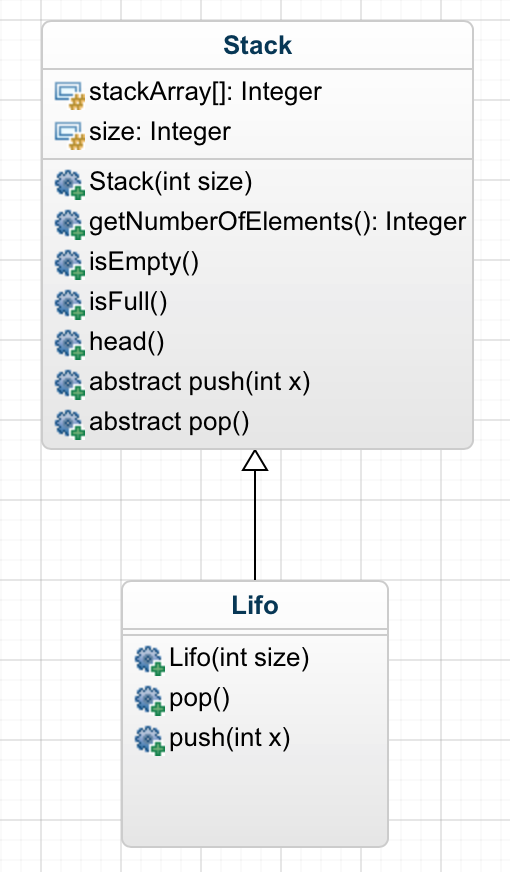


Figure 1: Class diagram of the Stack class

This architecture allows us to easily implement a potential FIFO stack. Indeed, the super class « Stack » implement all the common methods between FIFO and LIFO. The other two methods (push and pop) are abstract and must be defined in the subclasses. The mechanism of these two are different depending on the subclass.

# Question 4

We have to re-implement pop, it mustn’t be “LIFO” specific anymore. Other remain the same. Then we add one test to check if we performed a LIFO pop.

# Question 5

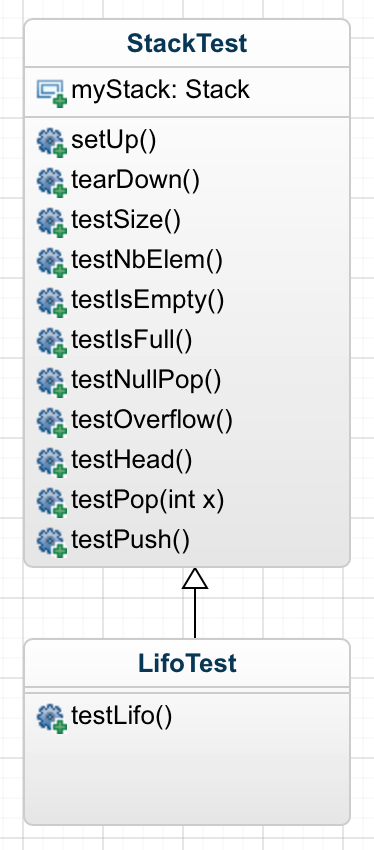


Figure : Class Diagram of tests class

We keep pop and push in the « StackTest » because we would only check if an element had been deleted or added. Not the value of the deleted/added element.

# Question 6

See code

# Question 7

See code

# Question 8

See code

# Question 9

We could re-implement the FIFO class by using the Super class « Stack » defined before.

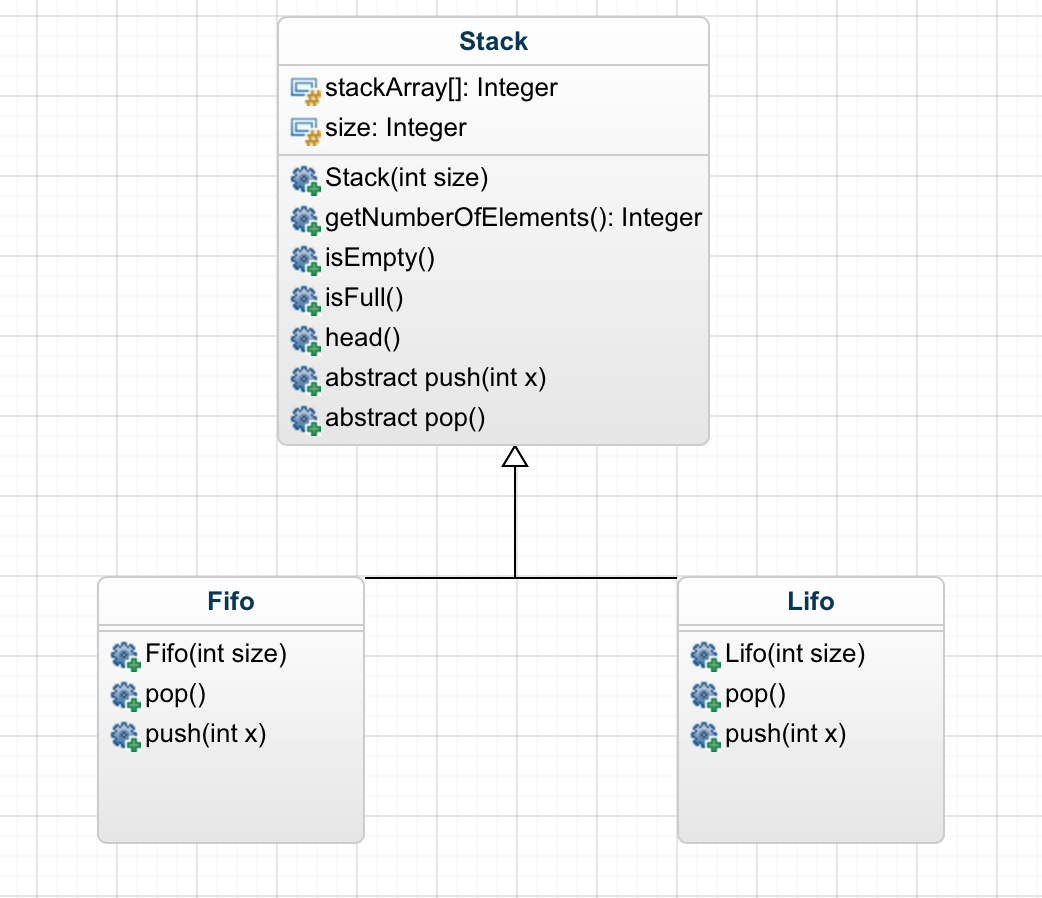
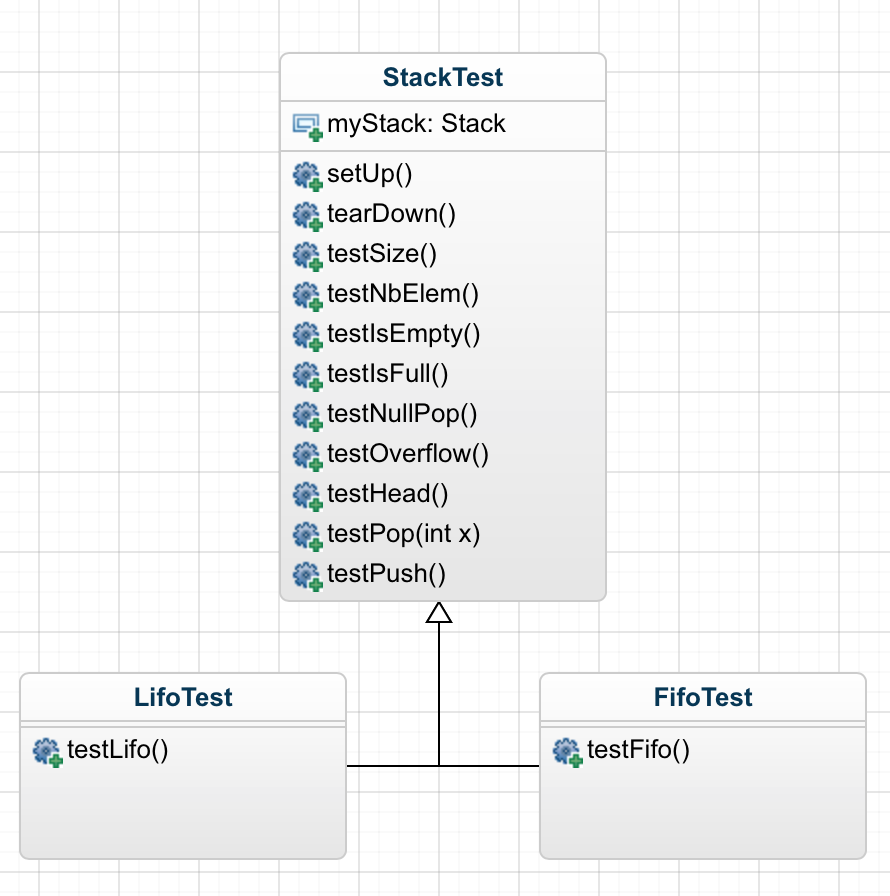


Figure : Class Diagram of FIFO/LIFO classes

In the FIFO class, we re-implement pop and push to fit FIFO constraints.   
Then we have to implement the test class to check whether or not we performed a FIFO pop.

# Question 10

Class diagram to perform the tests:



# Question 11

First, we should declare an Array of « Objects » instead of Integers. Thus, we could use this class to store all kind of data types.

You should also add a function “editSize(int x)” to increase/decrease the stack size without having to delete it.

# Question 12

See code

# Question 13

See code