# 02324 Advanced Programming, exercises Week 1

### Task 1. Define a Java interface for complex numbers from existing class implementation

The class which implements complex numbers is given here:

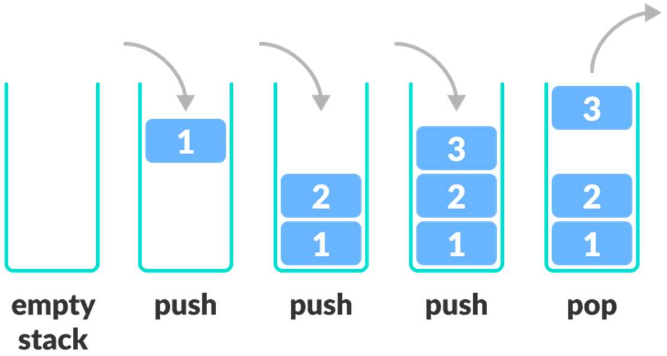
https://introcs.cs.princeton.edu/java/32class/Complex.java.html

You should:

1. Make the interface IComplex.java that contains all the methods from the class Complex.java
2. Modify Complex.java so that it implements IComplex.java

### Task 2. Two different implementations of Stack abstract data type

A stack is a classical data structure, which is sometimes called a LIFO (last in first out). A stack is a linear data structure where elements can be added (pushed) and removed (popped) only at one fixed end of the structure, which is called top. The following image illustrate stack operations:



In this task you should implement 2 classes ArrayStak and LinkedStak, each of which implements the following interface Stack: ¨

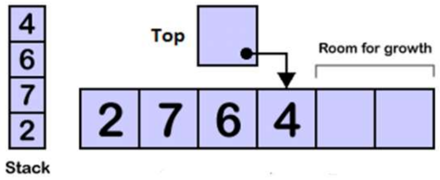
```
interface Stack {

        void push(String (e);

        String pop();

        boolean isEmpty();

        boolean isFull();

        void show();

}
```

Explanation of operations:

- push adds the item first in the stack.
- pop removes the top element from the stack.
- isEmpty should return whether the stack is empty.
- isFull should return whether the stack is full.
- show must print all the elements from the top to the bottom of the stack. It is not expected that the method empties the stack. The stack can only be emptied using. successive pop().

You should:

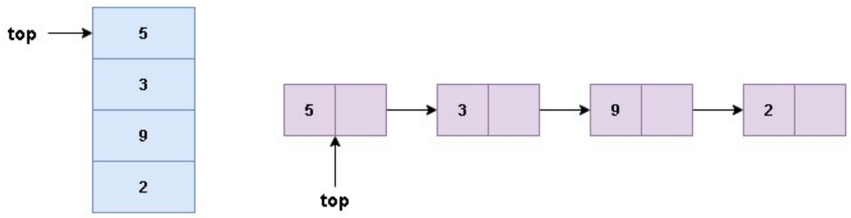
1) create a class ArrayStack, which implements the interface Stack. Use an array to store the elements on the stack. The following picture illustrates the implementation:



Note that an array has a fixed size, but if a stack size changes - how is it dealt with? How to keep track of where the next element is to be inserted?

2) Then, create a class StakTest, which create a stack (with your implementation), and saves the following items on the stack: ”This”, ”is”, ”a”, ”strange”, ”sentance”. You need to make sure to test all the methods in this subtask - Like with a JUnit test. The method show() should for example. write the sentence "sentence strange a is This".

3) Then, create a class LinkedStak which also implements the interface Stack. This time, a linked list should be used to store the items on the stack. A linked list consists of items with each item knowing the next item in the list:



4) Repeat subtask 2) but now with the class LinkedStak

5) Try also with other elements than those indicated in 2)