

სემინარი 3. შაბლონები ჯავაში

1. შექმენით კლასი Math, რომელსაც აქვს მეთოდი add ორი მთელი რიცხვის შესაკრებად. კლასის იმპლემენტაციის დაწერამდე შექმენით JUnit-ის ტესტი აღნიშნული მეთოდის შესამოწმებლად.

2. შექმენით კლასი MyStack რომელიც მასივის გამოყენებით გააკეთებს სტეკის რეალიზაციას. კონკრეტულად, კლასში აღწერეთ მეთოდები:

push - მეთოდს გადაეცემა ობიექტი, რომელიც მან უნდა ჩაამატოს სტეკში.

pop - მეთოდი აბრუნებს ობიექტს სტეკიდან.

addAll - მეთოდს გადაეცემა Collection ტიპის ობიექტი. მან Collection-ის ყველა ელემენტი უნდა ჩაამატოს სტეკში.

size - მეთოდი აბრუნებს სტეკში არსებული ელემენტების რაოდენობა.

კლასის იმპლემენტაციის დაწერამდე შექმენით JUnit ტესტები მეთოდების შესამოწმებლად.

Math კლასი

```
public class MyMath {  
    public int add(int a, int b){  
        return a+b;  
    }  
}
```

JUnit ტესტი MyStack კლასისთვის

```
import static org.junit.Assert.*;  
  
import org.junit.Test;  
  
public class MyMathTest{  
  
    @Test  
    public void testAdd1() {  
        MyMath fixture = new MyMath();  
        int a = 3;  
        int b = 7;  
        int result = fixture.add(a, b);  
        assertEquals(10,result);  
    }  
}
```

MyStack კლასი

```
import java.util.Collection;  
  
public class MyStack<T> {  
    private static final int INITIAL_SIZE = 2;  
    private Object[] arr;  
    private int logLen;  
  
    public MyStack() {  
        arr = new Object[INITIAL_SIZE];  
        logLen = 0;  
    }  
}
```

```

    public void push(T a){
        if (arr.length == logLen){
            grow();
        }
        arr[logLen++] = a;
    }

    private void grow() {
        Object[] temp = new Object[2*arr.length];
        for (int i=0; i<arr.length; i++){
            temp[i] = arr[i];
        }
        arr = temp;
    }

    @SuppressWarnings("unchecked")
    public T pop(){
        return (T)arr[--logLen];
    }

    public void addAll(Collection<T> a){
        for (T temp: a){
            push(temp);
        }
    }

    public int size(){
        return logLen;
    }
}

```

JUnit ტესტი MyStack კლასისთვის

```

import static org.junit.Assert.*;

import org.junit.Test;

```

```
public class MyStackTest {

    @Test
    public void testPush() {
        MyStack<String> a = new MyStack<String>();
        a.push("bla");
        a.push("blu");
        assertEquals(2, a.size());
    }

    @Test
    public void testSize1() {
        MyStack<Integer> a = new MyStack<Integer>();
        for (int i=0; i<100; i++){
            a.push(i);
            assertEquals(i+1, a.size());
        }
    }

    @Test
    public void testSize2() {
        MyStack<Integer> a = new MyStack<Integer>();
        for (int i=0; i<100; i++){
            a.push(i);
        }
        for (int i=0; i<100; i++){
            a.pop();
            assertEquals(99-i, a.size());
        }
    }
}
```