D0745378 薛竣祐

M5.2

MyStack

package M52;  
public class MyStack {  
 private final int[] data;  
 private final int size;  
 private int index = 0;  
 public MyStack(int size) throws Exception{  
 if (size<1){  
 throw new Exception("Not a stack");  
 }  
 this.data = new int[size];  
 this.size = size;  
 }  
 public void push(int addData) throws Exception{  
 if (is\_full())  
 throw new Exception("Full");  
 data[index++] = addData;  
 }  
 public int pop() throws Exception{  
 if (is\_empty())  
 throw new Exception("Empty");  
 return data[--index];  
 }  
 public boolean is\_full(){  
 return size == index;  
 }  
 public boolean is\_empty(){  
 return index==0;  
 }  
}

MyStackTest

package M52;  
import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class MyStackTest {  
 MyStack stack;  
 @BeforeEach  
 void set(){  
 stack = null;  
 }  
 @Test  
 void push() throws Exception {  
 stack = new MyStack(3);  
 stack.push(10);  
 stack.push(10);  
 stack.push(10);  
 *assertThrows*(Exception.class,()->stack.push(10));  
 }  
  
 @Test  
 void pop() throws Exception {  
 stack = new MyStack(3);  
 *assertThrows*(Exception.class,()->stack.pop());  
 stack.push(10);  
 *assertEquals*(10,stack.pop());  
 stack.push(10);  
 stack.push(20);  
 *assertEquals*(20,stack.pop());  
  
 }  
 @Test  
 void initSize() throws Exception {  
 *assertThrows*(Exception.class,()->stack = new MyStack(0));  
 *assertThrows*(Exception.class,()->stack = new MyStack(-1));  
 stack = new MyStack(1);  
 stack.push(100);  
 *assertEquals*(100,stack.pop());  
 }  
 @Test  
 void fullAndEmpty() throws Exception{  
 stack = new MyStack(5);  
 *assertFalse*(stack.is\_full());  
 *assertTrue*(stack.is\_empty());  
 stack.push(10);  
 *assertFalse*(stack.is\_full());  
 *assertFalse*(stack.is\_empty());  
 stack.pop();  
 *assertTrue*(stack.is\_empty());  
 stack.push(1);  
 stack.push(2);  
 stack.push(-1);  
 stack.push(999);  
 stack.push(0);  
 *assertTrue*(stack.is\_full());  
 *assertFalse*(stack.is\_empty());  
 }  
}

測試結果一張含有 文字 的圖片

自動產生的描述

涵蓋度測試