

```
/**
 * -----
 * @Fichier      : TestStack.java
 * @Labo         : Laboratoire 7 : Tours de Hanoi
 * @Auteurs      : Slimani Walid & Baume Oscar
 * @Date         : 25.11.2022
 *
 * @Description  : Le but de ce programme est d'effectuer divers tests afin de contrôler
 *                  l'efficacité de notre classe Stack.
 * @Remarque     : / Aucune remarque
 * @Modification : / Aucune modification
 * -----
 */

package test;
import util.*;

public class TestStack {

    public static void main(String[] args) {
        // region Test des fonctionnalités de la classe Stack
        System.out.println("Construction de la stack :");
        Stack<String> stack = new Stack<>();
        System.out.println(stack);
        //test pop sur stack vide
        try{
            stack.pop();
        } catch (RuntimeException e){
            System.out.println(e.getMessage());
        }
        // test top sur stack vide
        try{
            stack.top();
        } catch (RuntimeException e){
            System.out.println(e.getMessage());
        }
        // test création itérateur sur stack vide
        try{
            ElementIterator it = new ElementIterator(stack.getHead());
        } catch (RuntimeException e){
            System.out.println(e.getMessage());
        }
        }

        stack = new Stack<String>();
        System.out.println("On ajoute dans la stack les valeurs suivantes : 7, 14, 95, 33 et 444 :");
        stack.push("7");
        stack.push("14");
        stack.push("95");
        stack.push("33");
        stack.push("444");
        System.out.println(stack);

        System.out.println("On dépile deux fois donc 444 et 33 disparaissent :");
        stack.pop();
        stack.pop();
        System.out.println(stack);

        System.out.println("On ajoute 23 :");
        stack.push("23");
        System.out.println(stack);

        System.out.println("On dépile deux fois donc 23 et 95 disparaissent, puis on affiche le sommet (14) :");
        stack.pop();
        stack.pop();
        System.out.println("Top = " + stack.top() + "\n");

        System.out.println("Affichage de la stack :");
        System.out.println(stack);
        // endregion

        // region Test de la génération d'un tableau d'objets à partir d'une Stack
        System.out.println("Construction d'une autre stack (10, 20, 30, 40, 50) : ");
        Stack<String> stack2 = new Stack<>("10", "20", "30", "40", "50");
        System.out.println(stack2);

        System.out.println("Génération d'un tableau d'objets à partir de la deuxième stack :");
```

```
Object[] tab = stack2.toArray();

System.out.print("[ ");
for(Object o : tab) {
    System.out.print("<" + o + ">" + " ");
}
System.out.println("]\n");
// endregion

// region Test des itérateurs
System.out.println("Création d'un itérateur qui pointe sur le sommet de la deuxième stack :");
ElementIterator it = new ElementIterator(stack2.getHead());
System.out.println("l'itérateur pointe sur -> " + it + "\n");

System.out.println("Tant que la pile contient des élément on itère et affiche le contenu :");
while(it.hasNext()) {
    System.out.println("next -> " + it.next());
}
// endregion
}
```