TestStack.java

```java
/**
 * @author  Sebastian Diaz & Guillaume Dunant
 * Date   : 04.12.2023
 * Fichier: TestStack.java
 */

import java.util.ArrayList;
import util.Iterator;
import util.Stack;

/**
 * Classe pour tester le bon fonctionnement de l'implémentation de la stack
 */
public class TestStack {

    //Code ANSI pour afficher en couleur dans la console
    private static final String GREEN = "\u001B[32m";
    private static final String RED   = "\u001B[31m";
    private static final String WHITE = "\u001B[37m";

    /**
     * Vérifie si la valeur attendue et la valeur obtenue sont égales
     * @param expected Valeur attendue
     * @param result Valeur obtenue
     */
    private static void checkResult(String expected, String result){
        System.out.println("Valeur attendue: " + expected);
        System.out.println("Valeur obtenue : " + result);
        System.out.println("Résultat: " +
            (expected.equals(result)? (GREEN + "Réussi") : (RED + "Echoué"))
            + WHITE + "\n");
    }

    /**
     * Vérifie si l'objet passé en paramètre est null
     * @param o Objet à vérifier
     */
    private static void checkNull(Object o){
        System.out.println("Valeur attendue: null");
        System.out.println("Valeur obtenue : " + (o == null? "null" : o.toString()));
        System.out.println("Résultat: " +
            (o == null? (GREEN + "Réussi") : (RED + "Echoué"))
            + WHITE + "\n");
    }

    /**
     * Programme principal de test
     * @param args Arguments du programme
     */
    public static void main(String[] args) {
        System.out.println("**Programme de test de la stack**\n");

        //Test ajout d'éléments
        System.out.println("Test Stack.push()");

        Stack<Integer> stck = new Stack<>();
```

```java
        String expectedResult = "[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]";
        String result = "";

        for(int i = 1; i <= 10; ++i){
            stck.push(i);
        }
        result = stck.toString();

        checkResult(expectedResult, result);

        //Test récupération des élémemts
        System.out.println("Test Stack.pop()");
        ArrayList<Integer> resultArray = new ArrayList<>();
        for(int i = 1; i <= 10; ++i){
            resultArray.add(stck.pop());
        }

        checkResult(expectedResult, resultArray.toString());

        //Test que la stack soit bien vide
        System.out.println("Vérifictation que la stack est vide");
        expectedResult = "[]";

        checkResult(expectedResult, stck.toString());

        //Test de Stack.pop() sur une stack vide
        System.out.println("Test de Stack.pop() sur une stack vide");

        checkNull(stck.pop());

        //Test de l'itérateur
        System.out.println("Test de l'itérateur");
        resultArray = new ArrayList<>();
        expectedResult = "[9, -32, 45, 21]";
        stck.push(21);
        stck.push(45);
        stck.push(-32);
        stck.push(9);

        Iterator<Integer> ite = stck.getIterator();
        while (ite.hasNext()) {
            resultArray.add(ite.next());
        }

        checkResult(expectedResult, resultArray.toString());

        //Test itérateur sur une stack vide
        System.out.println("Test itérateur sur une stack vide");
        stck = new Stack<>();
        ite = stck.getIterator();

        System.out.println("Iterator.hasNext(): " + ite.hasNext());
        checkNull(ite.next());
    }
}

```