TRAVAILLE A RENDRE

TP1 : POO JAVA **NOM ET PRENOM :IZIKKI Hajar**

PROF ENCADRANT : M. OUKDACH

Exercice 7 :

La déclaration de la classe PILE :

```java
class Pile {

    static final int MAX = 8 ;
    char t[];
    int top;

    public Pile() {
        t = new char[MAX];
        top = -1;
    }

    public void empiler(char c) {
        if (!estPleine()) {
            t[++top] = c;
        } else {
            System.out.println(x:"Pile pleine");
        }
    }

    public char sommet() {
        if (!estVide()) {
            return t[top];
        } else {
            System.out.println(x:"Pile vide");
            return '\0'; // Caractère nul pour indiquer une pile vide
        }
    }

    public void depiler() {
        if (!estVide()) {
            top--;
        } else {
            System.out.println(x:"Pile vide, impossible de dépiler");
        }
    }

    public boolean estVide() {
        return top == -1;
    }

    public boolean estPleine() {
        return top == MAX - 1;
    }
}
```

On a créé une classe en Java appelée `Pile` qui implémente une structure de pile (stack). Cette classe comporte les opérations basiques d'empilement (`empiler`), de consultation du sommet de la pile (`sommet`), de dépilement (`depiler`), et de vérification si la pile est vide (`estVide`) ou pleine (`estPleine`).

Voici une explication de chaque méthode de la classe `Pile` :

- **Constructeur `Pile()` :** Initialise une pile vide en créant un tableau de caractères de taille `MAX` et initialise `top` à -1 pour représenter une pile vide.
- **Méthode `empiler(char c)` :** Ajoute un élément à la pile si elle n'est pas pleine. Si la pile est pleine, affiche "Pile pleine".
- **Méthode `sommet()` :** Retourne le caractère en haut de la pile (le sommet) si la pile n'est pas vide. Si la pile est vide, affiche "Pile vide" et retourne le caractère nul (`'\0'`) pour indiquer une pile vide.

- **Méthode `depiler()` :** Supprime l'élément du haut de la pile si elle n'est pas vide. Si la pile est vide, affiche "Pile vide, impossible de dépiler".
- **Méthode `estVide()` :** Renvoie `true` si la pile est vide (si `top` est égal à -1), sinon `false`.
- **Méthode `estPleine()` :** Renvoie `true` si la pile est pleine (si `top` est égal à `MAX - 1`), sinon `false`.

Question 1 :

TestPile.java > Algo > main(String[])

```java
import java.util.Scanner;
import java.util.Stack;
class Algo {


    Run | Debug
    public static void main(String[] args) {
        // Créer une pile
        Stack<Character> pile = new Stack<>();

        // Créer un objet Scanner pour lire l'entrée
        Scanner scanner = new Scanner(System.in);

        // Lire le premier caractère
        System.out.print(s:"Entrez un caractère (ou '#' pour terminer): ");
        char input = scanner.next().charAt(index:0);

        // Tant que c n'est pas '#', empiler c sur la pile et lire le caractère suivant
        while (input != '#') {
            pile.push(input);
            System.out.print(s:"Entrez un caractère (ou '#' pour terminer): ");
            input = scanner.next().charAt(index:0);
        }

        // Tant que la pile n'est pas vide
        while (!pile.isEmpty()) {
            // Récupérer le caractère en haut de la pile
            input = pile.peek();

            // Afficher le caractère
            System.out.print(input);

            // Dépiler la pile
            pile.pop();
        }
    }
}
```

Ce code est une manière simple de saisir des caractères et de les afficher dans l'ordre inverse de leur saisie en utilisant une pile.

L’exécution de ce code :

On remarque que le programme affiche les caractères entrée d’une manière inverse lorsque En tape : #

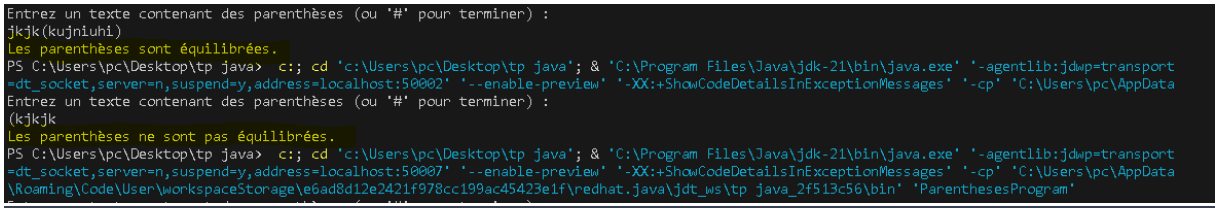
```
Exception in thread "main" java.util.NoSuchElementException
        at java.base/java.util.Scanner.throwFor(Scanner.java:945)
        at java.base/java.util.Scanner.next(Scanner.java:1486)
PS C:\Users\pc\Desktop\tp java>  c:; cd 'c:\Users\pc\Desktop\tp java'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-agentlib:jdwp=transpor
=dt_socket,server=n,suspend=y,address=localhost:50057' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\pc\AppDat
\Roaming\Code\User\workspaceStorage\e6ad8d12e2421f978cc199ac45423e1f\redhat.java\jdt_ws\tp java_2f513c56\bin' 'Algo'
Entrez un caractère (ou '#' pour terminer): r
Entrez un caractère (ou '#' pour terminer): a
Entrez un caractère (ou '#' pour terminer): j
Entrez un caractère (ou '#' pour terminer): a
Entrez un caractère (ou '#' pour terminer): h
Entrez un caractère (ou '#' pour terminer):


i
Entrez un caractère (ou '#' pour terminer): k
Entrez un caractère (ou '#' pour terminer): k
Entrez un caractère (ou '#' pour terminer): i
Entrez un caractère (ou '#' pour terminer): z
Entrez un caractère (ou '#' pour terminer): i
Entrez un caractère (ou '#' pour terminer): #
izikkihajar
PS C:\Users\pc\Desktop\tp java>
```

Question 2 :

```java
import java.util.Scanner;
import java.util.Stack;

class ParenthesesProgram {
    Run | Debug
    public static void main(String[] args) {

        Stack<Character> pile = new Stack<>();


        Scanner scanner = new Scanner(System.in);


        System.out.println(x:"Entrez un texte contenant des parenthèses (ou '#' pour terminer) :");
        String texte = scanner.nextLine();


        for (int i = 0; i < texte.length(); i++) {
            char c = texte.charAt(i);


            if (c == '(') {
                pile.push(c);
            }

            else if (c == ')' && !pile.isEmpty()) {
                pile.pop();
            }



            else if (c == '#') {
                break;
            }
        }


        if (pile.isEmpty()) {
            System.out.println(x:"Les parenthèses sont équilibrées.");
        } else {
            System.out.println(x:"Les parenthèses ne sont pas équilibrées.");
        }
    }
}
```

Ce programme fournit une vérification simple pour déterminer si les parenthèses dans la chaîne de texte sont correctement équilibrées ou non.

L'exécution de ce code :



Il y a une parenthèse ouvrante '(' suivie d'un espace, puis '#', puis un autre espace et enfin une parenthèse fermante ')'. Cependant, il n'y a pas de parenthèse ouvrante correspondante pour la parenthèse fermante. C'est pourquoi le programme indique que les parenthèses ne sont pas équilibrées.

Pour que les parenthèses soient correctement équilibrées, chaque parenthèse ouvrante '(' doit avoir une parenthèse fermante correspondante ')', et elles doivent être correctement appariées dans le texte saisi.