

ADIA

COMPTE RENDU

Effectué du : 3/12/2023

TP1 : Programmation JAVA

Réalisé par :

-Elhillali naoual

Encadrant :

- Yassine
Oukdache

Année Universitaire

2023/20

Premier programme complémentation de fichier Pile.java.

Attributs de classe :

MAX : une constante qui représente la taille maximale de la pile.

t : un tableau de caractères pour stocker les éléments de la pile.

top : un indice pour suivre le sommet de la pile.

Constructeur :

La classe a un constructeur qui initialise le tableau t avec la taille maximale MAX et initialise top à -1 pour indiquer que la pile est vide initialement.

Méthodes de la classe :

estVide() : renvoie true si la pile est vide, sinon false.

estPleine() : renvoie true si la pile est pleine, sinon false.

empiler(char c) : empile un caractère c sur la pile si elle n'est pas pleine.

depiler() : dépile un caractère de la pile et le renvoie si la pile n'est pas vide.

sommet() : renvoie le caractère situé au sommet de la pile sans le retirer.

afficherPile() : affiche le contenu de la pile.

Méthode main :

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3448]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dell\OneDrive\Bureau\exercice7de tp1>javac Pile.java

C:\Users\dell\OneDrive\Bureau\exercice7de tp1>java Pile.java
Element A empilé sur la pile.
Element B empilé sur la pile.
Element C empilé sur la pile.
Contenu de la pile: A B C
Element C dépilé de la pile.
Contenu de la pile: A B
Sommet de la pile: B

C:\Users\dell\OneDrive\Bureau\exercice7de tp1>
```

Deuxième programme qui lit une chaîne de caractère et l'imprime inversé.

Importation de la classe Scanner :

Cela importe la classe Scanner qui est utilisée pour lire l'entrée utilisateur.

Déclaration de la classe TestPile :

La méthode principale du programme.

Initialisation du scanner :

Un objet Scanner est créé pour lire l'entrée utilisateur à partir de la console.

Saisie d'une chaîne de caractères :

L'utilisateur est invité à entrer une chaîne de caractères qui est lue à l'aide de scanner.nextLine().

Création d'une pile :

Un objet Pile est créé. La classe Pile doit être définie ailleurs dans le code (peut-être dans un autre fichier) pour que cela fonctionne.

Empilage des caractères dans la pile :

Chaque caractère de la chaîne est empilé dans la pile jusqu'à ce que le caractère # soit rencontré, moment où la boucle s'arrête.

Dépilage et impression des caractères inversés :

Les caractères sont défilés de la pile et imprimés à l'écran, ce qui produit la chaîne inversée.

Fermeture du scanner :

Le scanner est fermé pour éviter des fuites de ressources.

C:\Windows\System32\cmd.exe

```
C:\Users\dell\OneDrive\Bureau\exercice7de tp1>javac Pile.java

C:\Users\dell\OneDrive\Bureau\exercice7de tp1>java Pile.java
Element A empilé sur la pile.
Element B empilé sur la pile.
Element C empilé sur la pile.
Contenu de la pile: A B C
Element C dépilé de la pile.
Contenu de la pile: A B
Sommet de la pile: B

C:\Users\dell\OneDrive\Bureau\exercice7de tp1>javac Pile.java TestPile.java

C:\Users\dell\OneDrive\Bureau\exercice7de tp1>java TestPile.java
Entrez une chaîne de caractères (terminée par #) : naoual#
Element n empilé sur la pile.
Element a empilé sur la pile.
Element o empilé sur la pile.
Element u empilé sur la pile.
Element a empilé sur la pile.
Element l empilé sur la pile.
Chaîne inversée : Element l dépilé de la pile.
lElement a dépilé de la pile.
aElement u dépilé de la pile.
uElement o dépilé de la pile.
oElement a dépilé de la pile.
aElement n dépilé de la pile.
n
```

Programme qui lit un texte contenant une série de parenthèses et qui à la rencontre du caractère (il l'empile sur une pile p à la rencontre du caractère) il dépile la pile ignore tout autre caractère. s'arrête à la lecture du caractère #.

Importation de la classe Scanner :

Déclaration de la classe TestParenthese :

Méthode main :

Initialisation du scanner

Saisie d'un texte :

Création d'une pile

Parcours du texte et gestion des parenthèses :

avec La boucle parcourt chaque caractère du texte et empile les parenthèses ouvrantes (dans la pile. Si une parenthèse fermante) est rencontrée, elle est défilée de la pile s'il y a une parenthèse ouvrante

correspondante. Si la pile est vide lorsque nous rencontrons une parenthèse fermante, cela signifie que les parenthèses sont mal équilibrées, et le programme se termine.

Vérification de l'équilibre des parenthèses à la fin :

Si la pile est vide à la fin du parcours du texte, cela signifie que toutes les parenthèses ouvrantes ont été appariées des parenthèses fermantes, et le texte a des parenthèses bien équilibrées.

Fermeture du scanner :

C:\Windows\System32\cmd.exe

```
Entrez une chaîne de caractères (terminée par #) : naoual#
Element n empilé sur la pile.
Element a empilé sur la pile.
Element o empilé sur la pile.
Element u empilé sur la pile.
Element a empilé sur la pile.
Element l empilé sur la pile.
Chaîne inversée : Element l dépilé de la pile.
lElement a dépilé de la pile.
aElement u dépilé de la pile.
uElement o dépilé de la pile.
oElement a dépilé de la pile.
aElement n dépilé de la pile.
n
C:\Users\dell\OneDrive\Bureau\exercice7de tp1>javac File.java TestPrentthese.java
error: file not found: TestPrentthese.java
Usage: javac <options> <source files>
Use --help for a list of possible options
C:\Users\dell\OneDrive\Bureau\exercice7de tp1>javac File.java TestParenthese.java
C:\Users\dell\OneDrive\Bureau\exercice7de tp1>javac TestPile.java TestParenthese.java
C:\Users\dell\OneDrive\Bureau\exercice7de tp1>java TestParenthese.java
Entrez un texte contenant des parenthèses (terminé par #) : na(wa)l#
Element ( empilé sur la pile.
Element ( dépilé de la pile.
Les parenthèses sont bien équilibrées.
C:\Users\dell\OneDrive\Bureau\exercice7de tp1>
```

جامعة ابن زهر
ትዕሊክ ትፍር ጽዕር
UNIVERSITÉ IBN ZOHR



جامعة ابن زهر
ትዕሊክቲ ደፀር ጽዕር
UNIVERSITÉ IBN ZOHR



جامعة ابن زهر
ትዕሊክ ትፍር ጽዕር
UNIVERSITÉ IBN ZOHR



جامعة ابن زهر
ትዕሊክ ተፅዕኖ
UNIVERSITÉ IBN ZOHR



جامعة ابن زهر
ትዕሊክ ትፍር ጽዕር
UNIVERSITÉ IBN ZOHR



