**Algorithmique / Travaux pratiques**

S01 – Type Abstrait, Piles





23.02.2015

Adriano De Almeida Silva – T-1f

Alex Travasso – T-1f

package s01;

public class CharStack {

private int topIndex;

private char[] buffer;

// -------------------------------------

private static final int DEFAULT\_SIZE = 10;

// -------------------------------------

public CharStack() {

this(DEFAULT\_SIZE);

// -------------------------------------

public CharStack(int estimatedSize) {

buffer = new char[estimatedSize];

topIndex = -1;

// -------------------------------------

public boolean isEmpty() {

boolean resultat = false;

if (topIndex == -1) {

resultat = true;

return resultat;

// -------------------------------------

public char top() {

char resultat = '\0';

if (topIndex >= 0){

resultat = buffer[topIndex];

return resultat;

// -------------------------------------

public char pop() {

char result = buffer[topIndex];

buffer[topIndex] = '\0';

topIndex--;

return result;

// -------------------------------------

public void push(char x) {

if (topIndex == buffer.length - 1) {

char[] tabResult = new char[buffer.length \* 2];

int newTopIndex = topIndex;

for (int i = buffer.length - 1; i >= 0; i--) {

tabResult[i] = this.pop();

buffer = new char[tabResult.length];

buffer = tabResult;

topIndex = ++newTopIndex;

buffer[topIndex] = x;

} else {

buffer[++topIndex] = x;

public static boolean isBalanced(String l) {

boolean resultat = false;

char c;

CharStack s = new CharStack();

for (int i = 0; i < l.length(); i++) {

c = l.charAt(i);

if (isOpeningParenth(c)) {

s.push(c);

} else if (isClosingParenth(c) && isMatchingParenth(s.top(), c)

&& !s.isEmpty()) {

s.pop();

}else if (isClosingParenth(c) && s.isEmpty()){

return resultat;

}

}

if(s.isEmpty()) resultat = true;

return resultat;}

* 1. Méthode CharStack ()

L’appel à cette méthode sera toujours valide.

* 1. Méthode CharStack (int estSize)

Pour que l’appel à cette méthode soit valide, il faut l’appeler avec un nombre entier, positif et plus petit ou égal à 2'147'483'647.

* 1. Méthode isEmpty ()

L’appel à cette méthode sera toujours valide.

* 1. Méthode top ()

L’appel à cette méthode sera toujours valide.

* 1. Méthode pop ()

L’appel à cette méthode sera toujours valide.

* 1. Méthode push (char x)

Pour que l’appel à cette méthode soit valide, il faut l’appeler avec un seul caractère valide

/\* Première méthode

Crée une pile avec chaque lettre du string passé en paramètre, puis enlève lettre après lettre le contenu de la pile pour le mettre dans un string ---> Inverse l'ordre d'un string

\*/

/\* Deuxième méthode

Calcule le nombre de fois qu'apparait un caractère donné, dans une pile donnée.

\*/