Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

**Лабораторна робота №2**

Виконав

ст.гр. ІП-22-1

Токарєв С.О.

Перевірив:

Григорчук Л.І.

Івано-Франківськ

2023

**Теоретичні відомості:**

Визначення стеку:

Стек є однією з основних структур даних, що використовується для зберігання та управління даними. Стек - це абстрактна структура даних, яка працює за принципом "останній введений, перший вийшов" (Last-In, First-Out, або LIFO). Це означає, що останній елемент, доданий до стеку, буде першим, який видаляється.

**Операції стеку:**

Основні операції, які можна виконувати зі стеком, включають:

-push(item): Додати елемент до вершини стеку.

-pop(): Видалити та повернути елемент з вершини стеку.

-isEmpty(): Перевірити, чи стек порожній.

-size(): Повернути кількість елементів у стеці.

**Приклад використання стеку:**

Одним із типових випадків використання стеку є обхід дерева в глибину (DFS, Depth-First Search). Під час обходу дерева за допомогою рекурсії або ітеративного підходу з використанням стеку, ви можете зберігати вузли для подальшого оброблення.

**Застосування стеків:**

Стеки широко використовуються у великій кількості алгоритмів та програм, таких як:

Розбор арифметичних виразів (постфіксна нотація).

Управління викликами функцій та рекурсивними викликами.

Моделювання структур даних, таких як стек викликів.

Розв'язання задач на графах, такі як пошук шляху в глибину (DFS).

**Постановка задачі:**

1.Написати програму, яка передбачає введення в стек цілих чисел з клавіатури.

Вивести стек на екран.

2. Створити і роздрукувати два стеки - з парними та непарними числами серед

введених.

3. Будемо розглядати послідовності круглих і квадратних дужок, що відкриваються 1 1 закриваються ( ) [ ]. Серед усіх таких послідовностей виділимо правильні - ті, котрі можуть бути отримані за такими правилами:

-порожня послідовність правильна.

-якщо А и В правильні, то й АВ правильна.

- якщо А правильна, то [А] і (А) правильні. Наприклад, послідовності 0, [[1], [O[010] правильні, а послідовності ], )(. (],

())] - неправильні.

**Код:**

**Клас Stack.java;**

package lab3;

import java.util.Scanner;

public class Stack {

static Scanner *sc* = new Scanner(System.***in***);

private static Node *top*;

private int height;

public class Node {

char ch;

int value;

Node next;

public Node(int value) {

this.value = value;

}

public Node(char ch) {

this.ch = ch;

}

}

// Конструктори

public Stack(int value) {

Node newNode = new Node(value);

*top* = newNode;

height = 1;

}

public Stack() {

};

// Гетери

public void printStack() {

Node temp = *top*;

System.***out***.print("Stack -> : ");

while (temp != null) {

System.***out***.print(temp.value + " ");

temp = temp.next;

}

}

public void printChStack() {

Node temp = *top*;

System.***out***.print("Stack -> : ");

while (temp != null) {

System.***out***.print(temp.ch + " ");

temp = temp.next;

}

}

public void getTop() {

System.***out***.println(*top*.value);

}

public void getHeight() {

System.***out***.println(height);

}

// Метод для додавання елементів

public void push(int value) {

Node newNode = new Node(value);

if (height == 0) {

*top* = newNode;

} else {

newNode.next = *top*;

*top* = newNode;

}

height++;

}

public void push(char ch) {

Node newNode = new Node(ch);

if (height == 0) {

*top* = newNode;

} else {

newNode.next = *top*;

*top* = newNode;

}

height++;

}

public Node pop() {

if (height == 0)

return null;

else {

Node temp = *top*;

*top* = *top*.next;

temp.next = null;

height--;

return temp;

}

}

// завдання №1 додавання чисел в стек

public static void addNums() {

Stack numbers = new Stack();

System.***out***.print("Enter the number amount: ");

int amount = *sc*.nextInt();

for (int i = 0; i < amount; i++) {

System.***out***.print("Enter the number: ");

int num = *sc*.nextInt();

numbers.push(num);

}

numbers.printStack();

}

// Завдання №2 сортування чисел в стеки

public static void sortStack() {

Stack even = new Stack();

Stack odd = new Stack();

System.***out***.println("Enter the number amount: ");

int amount = *sc*.nextInt();

for (int i = 0; i < amount; i++) {

System.***out***.print("Enter the number: ");

int num = *sc*.nextInt();

if (Math.*abs*(num) % 2 == 0)

even.push(num);

else

odd.push(num);

}

System.***out***.print("-Even: ");

even.printStack();

System.***out***.println();

System.***out***.print("-Odd: ");

odd.printStack();

}

public static boolean isValidSequence(String sequence) {

Stack stack = new Stack(); // Ваша реалізація стеку

for (char c : sequence.toCharArray()) {

if (c == '(' || c == '[' || c == '{') {

stack.push(c);

} else if (c == ')' || c == ']' || c == '}') {

if (stack.isEmpty()) {

return false;

}

if ((c == ')' && *top*.ch != '(') || (c == ']' && *top*.ch != '[') || (c == '}' && *top*.ch != '{')) {

return false;

}else {

stack.pop();

}

}

}

return stack.isEmpty();

}

public boolean isEmpty() {

return *top* == null; // Порожній стек має значення top рівне null

}

}

}

**Клас Main:**

package lab3;

import java.util.Scanner;

public class Main {

public static void menu() {

Scanner c = new Scanner(System.***in***);

int k;

do{

System.***out***.println("1. Task #1 Enter the numbers from keyboard to stack");

System.***out***.println("2. Task #2 Sort out even and odd numbers");

System.***out***.println("3. Task #3 Check if sequence of brackets is correct");

System.***out***.println("4. Exit");

k = c.nextInt();

switch (k) {

case 1:

Stack.*addNums*();

System.***out***.println("\n--------------------------");

break;

case 2:

Stack.*sortStack*();

System.***out***.println("\n--------------------------");

break;

case 3:

System.***out***.print("Enter the sequence of brackets: ");

c.next();

String seq = c.nextLine();

if (Stack.*isValidSequence*(seq) == true) {

System.***out***.println(seq + " - This sequence is correct");

} else {

System.***out***.println(seq + " - This sequence is incorrect");

}

System.***out***.println("\n--------------------------");

case 4:

System.***out***.println("\n\nBye!");

}

}while (k != 4);

}

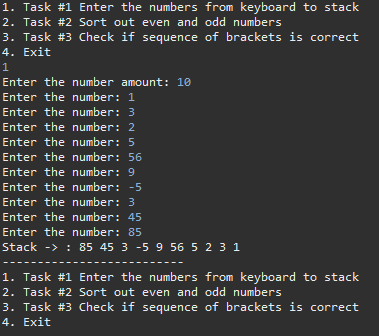
public static void main(String[] args) {

*menu*();

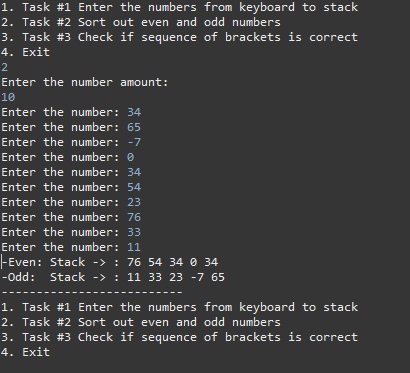
}

}

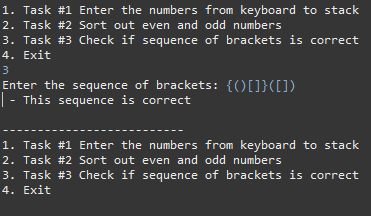
**Приклад виконання:**

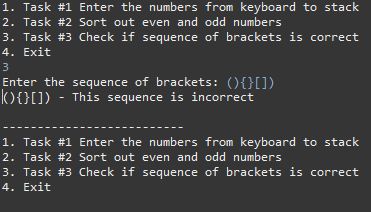


Завдання №1 - внесення чисел з клавіатури в стек



Завдання №2 - Сортування чисел з клавіатури в 2 стеки парних і непарних чисел





Завдання №3 - перевірка послідовності дужок на правильність

**Висновок:**

У лабораторній роботі навчився створювати стеки, працювати з ними на прикладі завдань з додаванням чисел до стеки, сортування стеків і завдання на перевірку послідовності дужок.