**Lucrare de laborator nr. 7**

**Panfil Ion Grupa: CR-221 GitHub:** https://github.com/Panfil-Ion/TAP.git

**Tema lucrării:** De realizat sarcinile utilizînd colecţiile

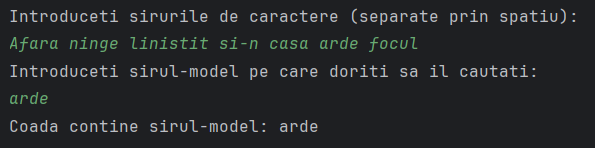
**Varianta 2**

Creaţi o clasă Queue bazată pe colecţii. Introduce o serie de siruri de caractere şi stabili dacă există un şir-model în această coadă.

**Codul**

import java.util.LinkedList;  
import java.util.Scanner;  
  
public class Queue {  
 private LinkedList<String> queue;  
  
 public Queue() {  
 this.queue = new LinkedList<>();  
 }  
  
 public void enqueue(String item) {  
 queue.addLast(item);  
 }  
  
 public boolean contains(String pattern) {  
 return queue.contains(pattern);  
 }  
  
 public static void main(String[] args) {  
 Queue myQueue = new Queue();  
 Scanner scanner = new Scanner(System.in);  
  
 // Introduce unele siruri de caractere in coada  
 System.out.println("Introduceti sirurile de caractere (separate prin spatiu):");  
 String[] inputStrings = scanner.nextLine().split(" ");  
 for (String str : inputStrings) {  
 myQueue.enqueue(str);  
 }  
  
 // Verifica daca coada contine un sir-model  
 System.out.println("Introduceti sirul-model pe care doriti sa il cautati:");  
 String pattern = scanner.nextLine();  
 if (myQueue.contains(pattern)) {  
 System.out.println("Coada contine sirul-model: " + pattern);  
 } else {  
 System.out.println("Coada nu contine sirul-model: " + pattern);  
 }  
  
 scanner.close();  
 }  
}

Rezultatul codului



**Concluzie**

Codul dat demonstrează o implementare simplă a unei cozi utilizând o listă dublu înlănțuită în limbajul Java. Prin intermediul clasei **Queue**, utilizatorul poate introduce șiruri de caractere în coadă și verifica dacă aceasta conține un șir-model specificat. Folosind colecțiile din Java, acest cod oferă o modalitate eficientă de a simula operațiile de coadă și de a efectua căutări în cadrul acesteia.