**1. Tema lucrării:**

Crearea excepţiilor

**2. Scopul lucrării:**

⮚ Însuşirea modalităţilor de creare şi realizare a excepţiilor în Java;

**3. Etapele de realizare:**

1. Realizarea mai mulror tipuri de excepţii;

2. Realizarea excepţiilor standarte ;

3. Crearea excepţiilor poprii ;

**4. Crearea interfeţii programului;**

**5. Prezentarea lucrării.**

**Sarcina**

Introduceţi un număr, sub forma de cifre în stivă. Afişaţi numărul la care cifrele sunt inversate.

**V2**

**Codul**

import java.util.Scanner;  
  
public class NumberStack {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
  
 try {  
 System.out.print("Introduceți un număr sub formă de cifre: ");  
 String input = scanner.nextLine();  
  
 validateInput(input); // Verificăm input-ul utilizatorului  
  
 // Inversăm cifrele și afișăm rezultatul  
 String reversedNumber = reverseNumber(input);  
 System.out.println("Numărul cu cifrele inversate este: " + reversedNumber);  
 } catch (InvalidNumberException e) {  
 System.out.println("Eroare verificată: " + e.getMessage());  
 } catch (NumberFormatException e) {  
 System.out.println("Eroare: Numărul introdus nu este valid.");  
 } catch (RuntimeException e) {  
 System.out.println("Eroare de timpul execuției: " + e.getMessage());  
 } finally {  
 // Închidem scanner-ul în blocul finally pentru a ne asigura că resursele sunt eliberate corect  
 if (scanner != null) {  
 scanner.close();  
 }  
 }  
 }  
  
 // Metoda pentru validarea input-ului utilizatorului  
 public static void validateInput(String input) throws InvalidNumberException {  
 if (!input.matches("[0-9]+")) {  
 throw new InvalidNumberException("Numărul introdus conține caractere nevalide.");  
 }  
 }  
  
 // Metoda care inversează cifrele numărului  
 public static String reverseNumber(String number) {  
 StringBuilder reversedNumber = new StringBuilder();  
 for (int i = number.length() - 1; i >= 0; i--) {  
 reversedNumber.append(number.charAt(i));  
 }  
 return reversedNumber.toString();  
 }  
}  
  
// Excepție verificată pentru numărul introdus nevalid  
class InvalidNumberException extends Exception {  
 public InvalidNumberException(String message) {  
 super(message);  
 }  
}

Rezultatul codului  




Concluzie

În acest laborator, am dezvoltat o aplicație simplă în Java care primește un număr de la utilizator, validează input-ul și afișează inversul acestuia. Am utilizat excepții pentru a gestiona situațiile neașteptate și am definit o excepție personalizată pentru cazul în care input-ul nu este valid. Aplicând blocuri **try-catch** și **finally**, ne-am asigurat că resursele sunt eliberate corect. Prin aceste metode, am dobândit o înțelegere mai profundă a gestionării excepțiilor și a asigurării fiabilității aplicațiilor Java.