**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 4

la cursul ***„Tehnici avansate de programare”***

**A efectuat :**   **St. gr. CR-221FR Serba Cristina**

**A verificat: Rotaru Lilia, asist. univ**

**Chișinău 2024**

# Tema lucrării:

Polimorfizmul

# Scopul lucrării:

Însuşirea modalităţilor de realizare a polimorfizmului în Java;

# Sarcina lucrării

Realizarea polimorfismului pentru clasele date cu crearea metodelor conform specificului claselor.

1.      Fraction <- ComplexFracion.

# Listingul programului

|  |
| --- |
| import java.util.Scanner;  class Fraction  {      protected int numarator;      protected int numitor;      public Fraction(int numarator, int numitor)      {          if (numitor == 0) {              throw new IllegalArgumentException("Numitorul nu poate fi 0");          }          this.numarator = numarator;          this.numitor = numitor;      }      public void display()      {          System.out.println(numarator + "/" + numitor);      }      public double value()      {          return (double) numarator / numitor;      }  }  class ComplexFraction extends Fraction  {      private int numaratorImg;      private int numitorImg;      public ComplexFraction(int realnumarator, int realnumitor, int numaratorImg, int numitorImg)      {          super(realnumarator, realnumitor);          this.numaratorImg = numaratorImg;          this.numitorImg = numitorImg;      }      @Override      public void display()      {          String realPart = numarator + "/" + numitor;          String imaginaryPart = numaratorImg + "/" + numitorImg + "i";          System.out.println(realPart + " + " + imaginaryPart);      }      @Override      public double value()      {          return super.value();      }  }  public class Main4  {      public static void main(String[] args)      {          System.out.println("Numaratorul si numitorul: ");          try (Scanner in = new Scanner(System.in)) {              int numarator = in.nextInt();              int numitor = in.nextInt();                Fraction fraction = new Fraction(numarator, numitor);              fraction.display();              System.out.println("Valoarea calculata: " + fraction.value());                System.out.println("Numaratorul si numitorul (real si imaginar respectiv): ");              int numaratorReal = in.nextInt();              int numitorReal = in.nextInt();              int numaratorImg = in.nextInt();              int numitorImg = in.nextInt();              Fraction complexFraction = new ComplexFraction(numaratorReal, numitorReal, numaratorImg, numitorImg);              complexFraction.display();              System.out.println("Valoarea calculata: " + complexFraction.value());          }      }  } |

# Concluzii:

În concluzie, programul demonstrează utilizarea moștenirii în Java pentru a modela fracții reale și complexe. Clasa de bază Fraction este utilizată pentru reprezentarea fracțiilor simple, oferind funcționalități precum afișarea fracției și calculul valorii sale numerice. Clasa derivată ComplexFraction extinde această funcționalitate prin adăugarea unei părți imaginare reprezentate de o altă fracție.

Programul permite utilizatorului să introducă atât fracții reale, cât și complexe, să le afișeze într-un format corespunzător și să calculeze valoarea numerică pentru partea reală. Prin această implementare, sunt demonstrate concepte fundamentale ale programării orientate pe obiecte, precum moștenirea, suprascrierea metodelor și utilizarea constructorilor pentru inițializarea atributelor.