**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 2

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

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

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

**Chișinău 2024**

# Tema lucrării:

Supraîncărcarea metodelor

# Scopul lucrării:

Însuşirea principiilor de supraîncărcare a metodelor şi realizarea lor;

Însuşirea principiilor de suprascriere a metodelor şi realizarea lor;

# Sarcina lucrării

Să se realizeze supraîncărcarea şi override a trei metode (utilizarea unei clase cu posibilităţile altei clase).

1. Sa se creeze obiectul clasa «Linie», folosind clasa «Cuvint».

# Listingul programului

|  |
| --- |
| import java.util.Scanner;  class Cuvant  {      private String cuvant;      public Cuvant()      {          cuvant = null;      }      public Cuvant(String cuvant)      {          this.cuvant = cuvant;      }      public String getCuvant()      {          return this.cuvant;      }        public void setCuvant()      {          this.cuvant = null;      }        public void setCuvant(String cuvant)      {          this.cuvant = cuvant;      }      @Override      public String toString()      {          return cuvant;      }  }  class Linie  {      Cuvant[] cuvinte;      int nrCuvinte;      int pos;      public Linie()      {          cuvinte = null;          nrCuvinte = 0;          pos = 0;      }      public Linie(int nrCuvinte)      {          this.nrCuvinte = nrCuvinte;          this.cuvinte = new Cuvant[nrCuvinte];          this.pos = 0;            boolean pass = false;          try (Scanner in = new Scanner(System.in))          {              while (!pass)              {                  System.out.println("Introduceti linia: ");                  String buffer = in.nextLine();                  String[] cuvinteInput = buffer.split(" ");                  for (String tempCuvant : cuvinteInput)                  {                      if (pos < nrCuvinte)                      {                          this.cuvinte[pos++] = new Cuvant(tempCuvant);                      }                  }                  if (pos < nrCuvinte) // daca sunt insuficiente cuvinte                  {                      System.out.println("Insuficiente cuvinte introduse. Introduceti din nou.");                  }                  else                  {                      pass = true;                  }              }          }      }      public void printLinie()      {          for (Cuvant cuvant : this.cuvinte)          {              System.out.print(cuvant + " ");          }            System.out.println();      }        public void setCuvant(String cuvant, int index)      {          Cuvant replacement = new Cuvant(cuvant);          this.cuvinte[index] = replacement;      }  }  public class Main2  {      public static void main(String[] args)      {          System.out.println("Introduceti nr de cuvinte: ");          try (Scanner in = new Scanner(System.in))          {              int nrCuvinte = in.nextInt();              Linie linie = new Linie(nrCuvinte);              System.out.println("Sirul inainte de inlocuire: ");              linie.printLinie();              linie.setCuvant("inlocuire", 1);              System.out.println("Sirul dupa inlocuire ");              linie.printLinie();          }      }  } |

# Concluzii:

În concluzie, programul realizat permite manipularea unui șir de cuvinte prin utilizarea claselor personalizate Cuvant și Linie. Acesta facilitează citirea, stocarea și modificarea cuvintelor dintr-o linie, oferind utilizatorului posibilitatea de a înlocui un cuvânt la o poziție specificată. Programul utilizează concepte fundamentale ale programării orientate pe obiecte, precum încapsularea, constructorii și suprascrierea metodei toString. Pe parcursul implementării, am reușit să înțeleg mai bine gestionarea array-urilor de obiecte, fluxurile de intrare și ieșire, precum și interacțiunea dintre clase în Java. Astfel, acest program oferă o bază solidă pentru învățarea și aplicarea principiilor OOP în rezolvarea problemelor practice.