

Seminar 6. Vectori in Java

Introducere

Timp de lucru estimat: 5 minute

Acest document reprezinta Seminarul 6 pentru cursul de introducere in programare Java. El contine o parte teoretica despre vectori (tablouri unidimensionale) in Java, 5 probleme rezolvate si 10 probleme propuse pentru exersarea conceptelor.

1 Partea Teoretica

Timp de lucru estimat: 60 minute

1.1 Ce este un vector in Java?

Un **vector** (sau tablou unidimensional) in Java este o structura de date care poate stoca o colectie de elemente de acelasi tip. Fiecare element din vector este accesat folosind un indice numeric.

1.2 Declararea si initializarea vectorilor

1.2.1 Declararea unui vector

Sintaxa generala pentru declararea unui vector in Java este:

```
tip[] numeVector;
```

Exemplu:

```
int[] numere;
```

1.2.2 Alocarea memoriei pentru un vector

Dupa declarare, trebuie sa alocam memorie pentru vector folosind operatorul **new**.

```
numeVector = new tip[dimensiune];
```

Exemplu:

```
numere = new int[5];
```

1.2.3 Declararea si alocarea in acelasi timp

Putem declara si aloca vectorul intr-o singura linie:

```
int[] numere = new int[5];
```

1.2.4 Initializarea vectorului cu valori

Putem initializa vectorul cu valori cunoscute folosind acolade:

```
int[] numere = {1, 2, 3, 4, 5};
```

1.3 Accesarea elementelor din vector

Elementele din vector sunt accesate folosind indicele lor. Indicii in Java incep de la 0.

Exemplu:

```
int primulElement = numere[0];  
numere[2] = 10; // Modifica al treilea element
```

1.4 Parcurgerea vectorilor

Cel mai comun mod de a parcurge un vector este folosind o bucla for.

Exemplu:

```
for (int i = 0; i < numere.length; i++) {  
    System.out.println(numere[i]);  
}
```

1.5 Vectori si metode

Vectorii pot fi transmisi ca argumente in metode si pot fi returnati de metode.

Exemplu:

```
public void afiseazaVector(int[] vector) {  
    // Cod pentru afisarea vectorului  
}
```

2 Partea Practica

Timp de lucru estimat: 90 minute

2.1 Probleme Rezolvate

2.1.1 Problema 1: Calcularea sumei elementelor unui vector

Enunt: Scrieti un program care citeste n numere intregi de la tastatura, le stocheaza intr-un vector si calculeaza suma elementelor din vector.

Cod Java:

```
import java.util.Scanner;

public class SumaVector {
    public static void main(String[] args) {
        Scanner scanner = new
            Scanner(System.in);
        System.out.print("Introduceti
            numarul de elemente: ");
        int n = scanner.nextInt();

        int[] vector = new int[n];
        int suma = 0;

        System.out.println("Introduceti
            elementele vectorului:");
        for (int i = 0; i < n; i++) {
            System.out.print("Elementul
                " + (i + 1) + ": ");
            vector[i] =
                scanner.nextInt();
            suma += vector[i];
        }

        System.out.println("Suma
            elementelor este: " + suma);
    }
}
```

2.1.2 Problema 2: Gasirea elementului maxim si minim dintr-un vector

Enunt: Scrieti un program care citeste n numere reale, le stocheaza intr-un vector si determina valoarea maxima si minima.

Cod Java:

```
import java.util.Scanner;

public class MaxMinVector {
    public static void main(String[] args) {
        Scanner scanner = new
            Scanner(System.in);
```

```

        System.out.print("Introduceti
                           numarul de elemente: ");
        int n = scanner.nextInt();

        double[] vector = new double[n];

        System.out.println("Introduceti
                           elementele vectorului:");
        for (int i = 0; i < n; i++) {
            System.out.print("Elementul
                               " + (i + 1) + ": ");
            vector[i] =
                scanner.nextDouble();
        }

        double max = vector[0];
        double min = vector[0];

        for (int i = 1; i < n; i++) {
            if (vector[i] > max) {
                max = vector[i];
            }
            if (vector[i] < min) {
                min = vector[i];
            }
        }

        System.out.println("Valoarea maxima
                           este: " + max);
        System.out.println("Valoarea minima
                           este: " + min);
    }
}

```

2.1.3 Problema 3: Verificarea daca un element exista in vector

Enunt: Scrieti un program care citeste n numere intregi, le stocheaza intr-un vector si verifica daca un numar dat de la tastatura se gaseste in vector.

Cod Java:

```

import java.util.Scanner;

public class CautareVector {
    public static void main(String[] args) {
        Scanner scanner = new
            Scanner(System.in);
        System.out.print("Introduceti
                           numarul de elemente: ");
    }
}

```

```

        int n = scanner.nextInt();

        int[] vector = new int[n];

        System.out.println("Introduceti
            elementele vectorului:");
        for (int i = 0; i < n; i++) {
            System.out.print("Elementul
                " + (i + 1) + ": ");
            vector[i] =
                scanner.nextInt();
        }

        System.out.print("Introduceti
            numarul de cautat: ");
        int numarCautat = scanner.nextInt();
        boolean gasit = false;

        for (int i = 0; i < n; i++) {
            if (vector[i] ==
                numarCautat) {
                gasit = true;
                break;
            }
        }

        if (gasit) {
            System.out.println("Numarul
                " + numarCautat + " se
                gaseste in vector.");
        } else {
            System.out.println("Numarul
                " + numarCautat + " nu
                se gaseste in vector.");
        }
    }
}

```

2.1.4 Problema 4: Inversarea elementelor unui vector

Enunt: Scrieti un program care citeste n numere intregi, le stocheaza intr-un vector si afiseaza vectorul in ordine inversa.

Cod Java:

```

import java.util.Scanner;

public class InversareVector {
    public static void main(String[] args) {

```

```

Scanner scanner = new
    Scanner(System.in);
System.out.print("Introduceti
    numarul de elemente: ");
int n = scanner.nextInt();

int[] vector = new int[n];

System.out.println("Introduceti
    elementele vectorului:");
for (int i = 0; i < n; i++) {
    System.out.print("Elementul
        " + (i + 1) + ": ");
    vector[i] =
        scanner.nextInt();
}

System.out.println("Vectorul
    inversat este:");
for (int i = n - 1; i >= 0; i--) {
    System.out.print(vector[i]
        + " ");
}

}
}

```

2.1.5 Problema 5: Sortarea elementelor unui vector

Enunt: Scrieti un program care citeste n numere intregi, le stocheaza intr-un vector si le sorteaza in ordine crescatoare folosind algoritmul Bubble Sort.

Cod Java:

```

import java.util.Scanner;

public class SortareVector {
    public static void main(String[] args) {
        Scanner scanner = new
            Scanner(System.in);
        System.out.print("Introduceti
            numarul de elemente: ");
        int n = scanner.nextInt();

        int[] vector = new int[n];

        System.out.println("Introduceti
            elementele vectorului:");
        for (int i = 0; i < n; i++) {

```

```

        System.out.print("Elementul
            " + (i + 1) + ": ");
        vector[i] =
            scanner.nextInt();
    }

    // Bubble Sort
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i -
            1; j++) {
            if (vector[j] >
                vector[j + 1]) {
                int temp =
                    vector[j];
                vector[j] =
                    vector[j
                    + 1];
                vector[j +
                    1] =
                    temp;
            }
        }
    }

    System.out.println("Vectorul sortat
        este:");
    for (int i = 0; i < n; i++) {
        System.out.print(vector[i]
            + " ");
    }
}
}

```

2.2 Probleme Propuse

1. Scrieti un program care citeste n numere intregi si determina media aritmetica a elementelor din vector.

2. Creati un program care citeste n numere intregi si afiseaza doar numerele pare din vector.

3. Scrieti un program care citeste un vector de n numere intregi si verifica daca vectorul este ordonat crescator.

4. Implementati un program care citeste doi vectori de n elemente si calculeaza produsul scalar al acestora.

5. Scrieti un program care citeste n numere intregi si elimina din vector toate aparitiile unui numar dat de la tastatura.

6. Creati un program care citeste un vector de n elemente si afiseaza elementul care

apare de cele mai multe ori (moda).

7. Scrieti un program care citeste un vector de **n** numere intregi si afiseaza suma numerelor prime din vector.

8. Implementati un program care inverseaza elementele unui vector fara a folosi un vector auxiliar (inversare in-place).

9. Scrieti un program care citeste un vector de **n** numere intregi si rotunjest vectorul la stanga cu o pozitie (primul element devine ultimul).

10. Creati un program care citeste **n** numere intregi si afiseaza secventa de lungime maxima de elemente egale din vector.

2.3 Instructiuni pentru Rezolvare

Pentru fiecare problema propusa:

- Analizati enuntul si identificati cerintele.
- Declarati si initializati vectorii necesari.
- Utilizati structurile de control adecvate (**for**, **while**, **if-else**).
- Testati programul cu diferite seturi de date pentru a verifica corectitudinea.