

Seminar 2. Lucrare practică

Introducere

Timp de lucru estimat: 5 minute

Acest document reprezintă Seminarul 2 pentru cursul de introducere Programarea calculatoarelor. El conține o parte teoretică despre tipurile de date, operatorii din Java, metoda `main` și transmiterea parametrilor din linia de comandă, precum și o parte de aplicații practice.

1 Partea Teoretică

Timp de lucru estimat: 35 minute

1.1 Tipuri de date în Java

Java oferă mai multe tipuri de date care pot fi împărțite în două categorii: tipuri primitive și tipuri de referință.

1.1.1 Tipuri de date primitive

Tipurile primitive sunt tipuri de date predefinite în Java:

- **byte** - numere întregi pe 8 biți (interval: -128 până la 127)
- **short** - numere întregi pe 16 biți (interval: -32,768 până la 32,767)
- **int** - numere întregi pe 32 biți (interval: -2,147,483,648 până la 2,147,483,647)
- **long** - numere întregi pe 64 biți (interval foarte mare)
- **float** - numere reale în virgulă mobilă pe 32 biți



- **double** - numere reale în virgulă mobilă pe 64 biți
- **char** - caracter Unicode pe 16 biți
- **boolean** - valoare logică (**true** sau **false**)

Exemplu:

```
int varsta = 25;  
float greutate = 70.5f;  
double pi = 3.1415;  
char litera = 'A';  
boolean esteAdevarat = true;
```

1.1.2 Tipuri de date de referință

Tipurile de referință se referă la obiecte și includ clase, interfețe și tipuri de matrice (array).

Exemplu:

```
String nume = "Ion Popescu";  
int[] numere = {1, 2, 3, 4, 5};
```

1.2 Operatori în Java

Operatorii sunt simboluri care efectuează operații pe variabile și valori.

1.2.1 Operatori aritmetici

- **+** - Adunare
- **-** - Scădere
- ***** - Înmulțire
- **/** - Împărțire
- **%** - Restul împărțirii (modulo)



Exemplu:

```
int a = 10;  
int b = 3;  
int suma = a + b; // 13  
int rest = a % b; // 1
```

1.2.2 Operatori de atribuire

- = - Atribuire simplă
- += - Atribuire cu adunare
- -= - Atribuire cu scădere
- *= - Atribuire cu înmulțire
- /= - Atribuire cu împărțire

Exemplu:

```
int x = 5;  
x += 3; // x devine 8
```

1.2.3 Operatori de relație

- == - Egalitate
- != - Diferență
- > - Mai mare decât
- < - Mai mic decât
- >= - Mai mare sau egal cu
- <= - Mai mic sau egal cu

Exemplu:

```
boolean rezultat = (a > b); // true dacă a este mai mare decât b
```



1.3 Metoda main în Java

Metoda `main` este punctul de intrare al oricărui program Java executabil. Fără această metodă, programul nu va rula.

Sintaxa metodei main:

```
public static void main(String[] args) {  
    // Codul programului  
}
```

Explicație:

- `public` - specificator de acces; metoda este accesibilă de oriunde.
- `static` - metoda aparține clasei și nu necesită o instanță pentru a fi apelată.
- `void` - metoda nu returnează nicio valoare.
- `main` - numele metodei care este punctul de intrare în program.
- `String[] args` - parametru care reprezintă un array de obiecte `String`; conține argumentele transmise programului prin linia de comandă.

Exemplu simplu de program Java cu metoda main:

```
public class Salut {  
    public static void main(String[] args) {  
        System.out.println("Salut, lume!");  
    }  
}
```

Când rulați acest program, va afișa mesajul `"Salut, lume!"` în consolă.



1.4 Transmiterea parametrilor către un program Java din linia de comandă

În Java, putem transmite argumente unui program prin linia de comandă. Aceste argumente sunt accesibile în program prin intermediul parametrului `String[] args` al metodei `main`.

Conversia argumentelor la tipuri numerice

Argumentele din linia de comandă sunt preluate ca șiruri de caractere (`String`). Pentru a le utiliza ca numere, trebuie să le convertim folosind metode precum:

- `Integer.parseInt(String s)` - pentru conversia la `int`
- `Float.parseFloat(String s)` - pentru conversia la `float`
- `Double.parseDouble(String s)` - pentru conversia la `double`

Exemplu:

```
public class ExempluParametri {  
    public static void main(String[] args) {  
        int paramIntreg = Integer.parseInt(args[0]);  
        float paramFloat = Float.parseFloat(args[1]);  
  
        System.out.println("Parametru întreg: " + paramIntreg);  
        System.out.println("Parametru float: " + paramFloat);  
    }  
}
```

Pentru a rula acest program și a transmite parametri, utilizați comanda:

```
java ExempluParametri 10 3.14
```

Rezultatul va fi:

```
Parametru întreg: 10  
Parametru float: 3.14
```

Explicație: Programul citește primul argument ca un `int` și al doilea ca un `float`, apoi le afișează.



2 Partea de Aplicații

Timp de lucru estimat: 50 minute

2.1 Problemă rezolvată

Enunț: Scrieți un program Java care calculează aria și perimetrul unui dreptunghi având lungimea și lățimea date ca argumente din linia de comandă, unde lungimea este un număr întreg și lățimea este un număr real (float).

Soluție:

```
public class Dreptunghi {
    public static void main(String[] args) {
        if (args.length < 2) {
            System.out.println("Utilizare: java Dreptunghi <lungime> <latime>");
            return;
        }

        int lungime = Integer.parseInt(args[0]);
        float latime = Float.parseFloat(args[1]);

        float perimetru = 2 * (lungime + latime);
        float aria = lungime * latime;

        System.out.println("Perimetrul dreptunghiului este: " + perimetru);
        System.out.println("Aria dreptunghiului este: " + aria);
    }
}
```

Pentru a rula programul cu lungimea 5 și lățimea 3.5, utilizați comanda:

```
java Dreptunghi 5 3.5
```

Explicație: Programul citește lungimea ca `int` și lățimea ca `float`, apoi calculează perimetrul și aria folosind aceste valori.



2.1.1 Detalii suplimentare

Observați că atunci când efectuăm operații între un `int` și un `float`, rezultatul este de tip `float`. Acest lucru este important pentru a evita erori de tip în program.

2.2 Probleme propuse

1. Scrieți un program Java care calculează aria și circumferința unui cerc având raza dată ca argument din linia de comandă. Raza este de tip `float`.

2. Scrieți un program Java care calculează volumul și aria totală a unui cub având latura dată ca argument din linia de comandă. Latura este de tip `int`.

3. Scrieți un program Java care convertește o temperatură dată în grade Celsius (transmisă ca argument de tip `float`) în grade Fahrenheit.

4. Scrieți un program Java care calculează media aritmetică a trei numere întregi date ca argumente din linia de comandă.

5. Scrieți un program Java care calculează aria și perimetrul unui pătrat având latura dată ca argument din linia de comandă. Latura este de tip `int`.

6. Scrieți un program Java care calculează distanța în mile pornind de la o distanță dată în kilometri (transmisă ca `float` din linia de comandă).

7. Scrieți un program Java care calculează valoarea finală după aplicarea unui discount procentual la un preț dat. Prețul și discount-ul sunt de tip `float` și sunt transmise ca argumente.

8. Scrieți un program Java care calculează indicele de masă corporală (IMC) având greutatea (în kilograme) și înălțimea (în metri) date ca argumente de tip `float` din linia de comandă.

9. Scrieți un program Java care calculează volumul și aria totală a unui cilindru având raza și înălțimea date ca argumente de tip `float` din linia de comandă.

10. Scrieți un program Java care calculează dobânda simplă având capitalul inițial (`int`), rata dobânzii (`float`) și perioada de timp (`int`) date ca argumente din linia de comandă.

Timp de lucru estimat pentru fiecare problemă: 5 minute