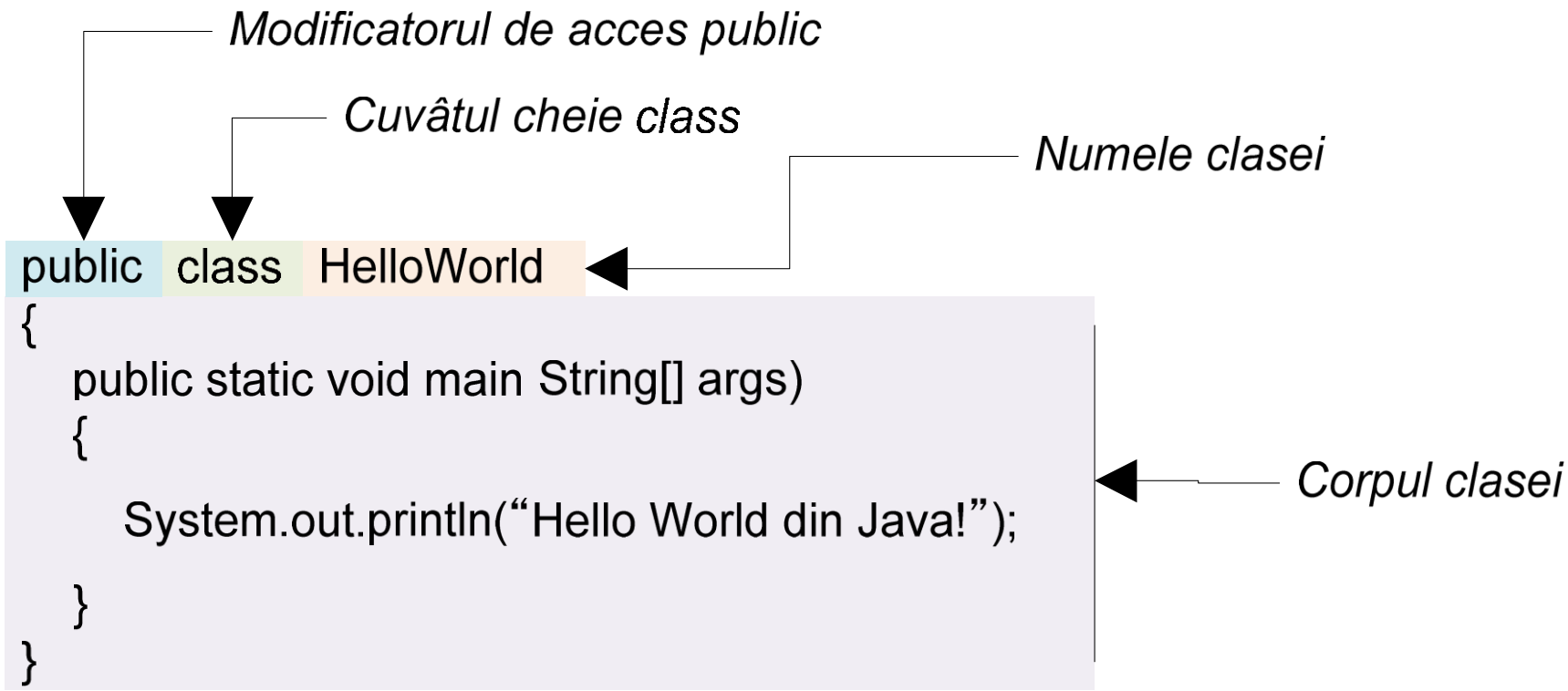
Recapitulare

Material didactic pentru Informatică

MariaGutu

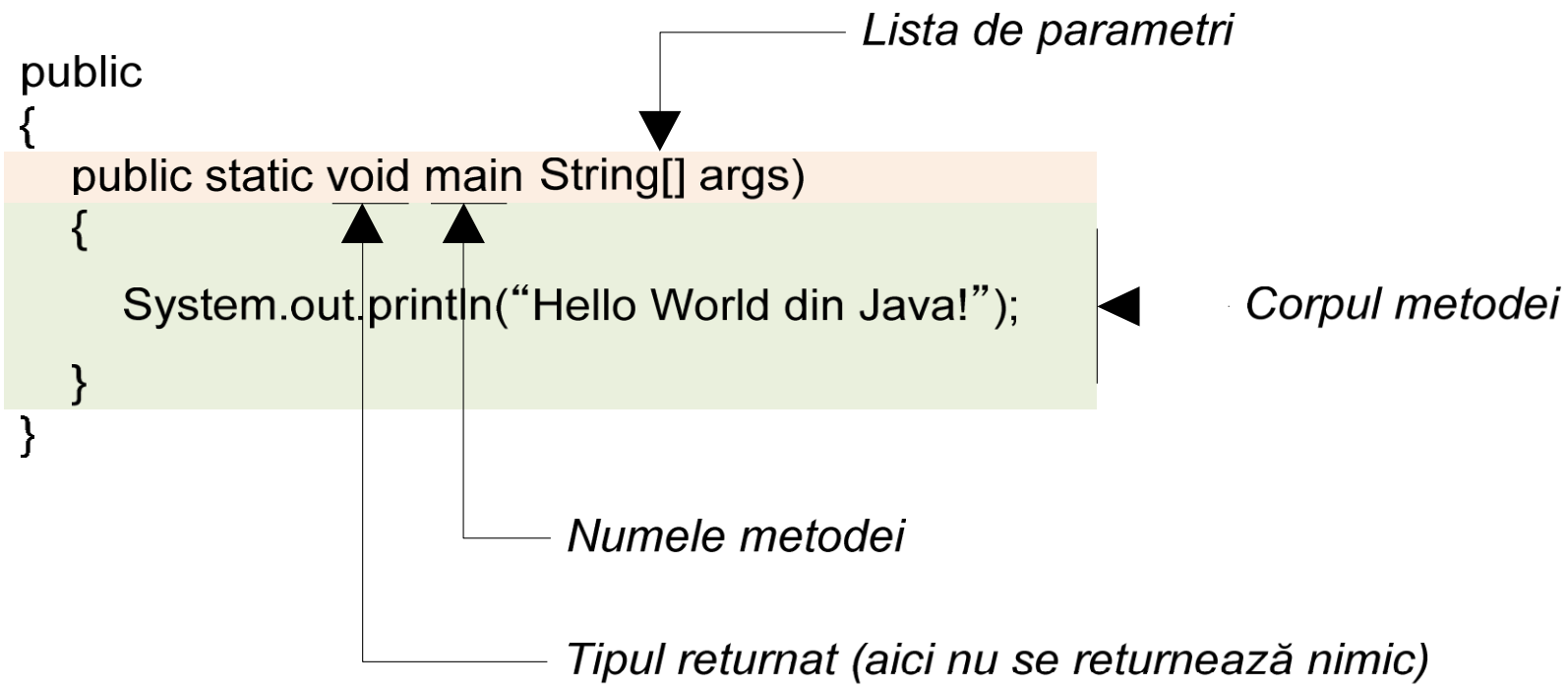
Structura unui program

Definiția clasei



Structura unui program

Definiția metodei



Tipurile de date primitive

- Tipuri întregi: *byte, short, int* și *long*;
- Tipuri reale: *float* și *double;*
- Tipul caracter: *char*;
- Tipul ligic: *boolean*.

Tipurile de date primitive

Tipul	Octeți alocați	Acoperirea
`byte`	1	-128 până la 127
`short`	2	-32768 până la 32767
`int`	4	-2147483648 până la 2147483647
`long`	8	-9223327036854755808L până la 9223327036854755807L

Tipul	Octeți alocați	Acoperirea
`float`	4	Aproximativ: ± 3.40282347E+38F (6-7 zecimale)
`double`	8	Aproximativ: ± 1.79769313486231570E+308 (13-14 zecimale)

Tipurile de date primitive

Secvența escape	Valoarea Unicode	Denumire
\b	\u0008	Backspace
\t	\u0009	tab (TAB)
\n	\u000A	Linefeed (LF)
\r	\u000D	carriage return (CR)
\"	\u0022	ghilimele
\'	\u0027	apostrof
\\	\u005C	backslash

- **final double PI = 3.14159**

Operatori aritmetici

```
int a = 54;
int mod;
mod = a % 10;
```

+ adunarea a două numere;

- scăderea a două numere;

***** înmulţirea a două numere;

/ împărţirea a două numere (rezultatul împărţirii pentru numere reale, câtul împărţirii pentru numere întregi);

% modulo (restul împărţirii a două numere întregi);

++ incrementarea (mărirea unei valori cu o unitate);

-- decrementarea (micşorarea unei valori cu o unitate).

Operatori aritmetici

Incrementarea / decrementarea

v++ - incrementare;

--v – decrementare.

Se pot folosi şi instrucţiuni de pre/pos incrementare/decrementare:

- post-incrementare: x = a++;

  - este echivalentă cu: x = a; a = a + 1;

- pre-incremenatare: x = ++a;

  - este echivalentă cu: a = a + 1; x = a;

Pentru decrementare se procedează în mod analog.

Operatori aritmetici

Incrementarea / decrementarea

a = ? b = ? c = ? x = ? y = ?

```
int a = 5, b, c, x, y;
b = a++;
c = ++a;
x = a--;
y = --a;
```

x = ? y = ? z = ? t = ?

```
int x = 1, y, z = 3,t;
x += y = z -= t = 5;
```

Operatori

Operatori relaționali

Operator	Acțiune
>	Mai mare decât
>=	Mai mare sau egal
<	Mai mic
<=	Mai mic sau egal
==	Egal
!=	Diferit

Operatori

Operatori logici

Operator	Acțiune
&&	ȘI
||	SAU
!	NU

Afișarea informației

Numere întregi

Ce va afișa următorul program?

```
public class TestFormat {

    public static void main(String[] args) {
      long a = 461012;
      System.out.format("%d%n", a);
//    System.out.format() ⇔ System.out.printf()
      System.out.format("%25d%n", a);
      System.out.format("%+8d%n", a);
      System.out.format("%,8d%n", a);
      System.out.format("%+,8d%n%n", a);
}
}
```

Afișarea informației

Numere întregi

Ce va afișa următorul program?

```
public class TestFormat {

    public static void main(String[] args) {
      long a = 461012;
      System.out.format("%d%n", a);
      System.out.format("%25d%n", a);
      System.out.format("%+8d%n", a);
      System.out.format("%,8d%n", a);
      System.out.format("%+,8d%n%n", a);
}
}
```

Output:

```
461012
                   461012
 +461012
 461 012
+461 012
```

Afișarea informației

Numere reale

Ce va afișa următorul program?

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

    double pi = Math.PI;

    System.out.format("%f%n", pi);
    System.out.format("%.3f%n", pi);
    System.out.format("%10.3f%n", pi);
    System.out.format("%-10.3f%n", pi);
    }
}
```

Afișarea informației

Numere reale

Ce va afișa următorul program?

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

    double pi = Math.PI;

    System.out.format("%f%n", pi);
    System.out.format("%.3f%n", pi);
    System.out.format("%10.3f%n", pi);
    System.out.format("%-10.3f%n", pi);
    }
}
```

Output:

```
3,141593
3,142
     3,142
3,142
```

Afișarea informației

Specificator de format	Conversia aplicată
%%	Inserarea simbolului %
%d	Specificatorul aplicat numerelor întregi (**byte, short, int, long**)
%f	Specificatorul aplicat numerelor reale **(float, double)**
%c	Specificatorul aplicat tipului de date **char**
%b sau %B	Specificatorul aplicat tipului de date **boolean**
%e sau %E	Specificatorul pentru **notația științifică**
%s sau %S	Specificatorul aplicat tipului de date **String**
%n	Trecerea din linie nouă a informației aflată după acest specificator

Scanner class

```
import java.util.Scanner;
class MyClass {
  public static void main(String[] args) {
    Scanner  tastatura = new  Scanner(System.in);

    System.out.println(„Introdu numele, varsta și nota medie:");
    String nume = tastatura.nextLine(); // Se introduce numele de la tastatura

     int varsta = tastatura.nextInt(); // Se introduce varsta in ani
    double notaMedie = tastatura.nextDouble(); // Se introduce nota medie

    // Afișarea în consolă
    System.out.println("Nume: " + nume);
    System.out.println("Varsta: " + varsta);
    System.out.println("Nota medie: " + notaMedie);
    tastatura.close();
  }
}
```

Scanner class

Numele metodei	Descrierea
nextBoolean()	Citește o valoare de tip **boolean** de la tastatură
nextByte()	Citește o valoare de tip **byte** de la tastatură
nextShort()	Citește o valoare de tip **short** de la tastatură
nextInt()	Citește o valoare de tip **int** de la tastatură
nextLong()	Citește o valoare de tip **long** de la tastatură
nextFloat()	Citește o valoare de tip **float** de la tastatură
nextDouble()	Citește o valoare de tip **double** de la tastatură
nextLine()	Citește o valoare de tip **String** de la tastatură
next().charAt(0)	Citește o valoare de tip **char** de la tastatură

Rezolvarea problemelor

5. Să se scrie un program care calculează valoarea funcției $y = x^2 + \left(\frac{x^3}{2}\right)^2 + 3$.

6. Se consideră coordonatele a două puncte A(x1, y1) și B(x2, y2). Să se scrie un program care calculează distanța dintre aceste puncte.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Rezolvarea problemelor

7. Se consideră că numerele *a, b, c* reprezintă lungimile laturilor unui triunghi oarecare, unde *a,b,c* – sunt introduse de la tastatură. Să se scrie un program care calculează aria acestui triunghi, folosind formula lui Heron $S = \sqrt{p(p-a)(p-b)(p-c)}$, unde *a, b, c* sunt laturile triunghiului, iar *p* este semiperimetrul lui (p=(a+b+c)/2).

Instrucțiuni condiționale

Java IF Statement

Material didactic pentru Informatică

Clasa a X-a

MariaGutu

Java IF Statement

Instrucțiunea IF permite execuția condiționată a unei instrucțiuni sau a unui bloc de instrucțiuni.

Sintaxă generală este:

```
IF (<Condiție>) {
 <BlockInstrucțiuni1>}
[else {< BlockInstrucțiuni2>] }
//instrucțiune sau bloc de cod spre execuție luate între {}
```

Java IF Statement

IF (<**Condiție**>)
<**Instrucțiune1**> [else < **Instrucțiune2**>]

Dacă rezultatul evaluării *ExpresieiCondiție* este *true*, atunci se execută Instrucțiunea1 iar, dacă *ExpresiaCondiție* are valoare *false*, atunci se execută Instrucțiunea2, dacă există, în caz contrar se trece la următoarea instrucțiune a programului.

Java IF Statement

IF (<**Condiție**>)
<**Instrucțiune1**> [else < **Instrucțiune2**>]

Dacă rezultatul evaluării *ExpresieiCondiție* este *true*, atunci se execută Instrucțiunea1 iar, dacă *ExpresiaCondiție* are valoare *false*, atunci se execută Instrucțiunea2, dacă există, în caz contrar se trece la următoarea instrucțiune a programului.

Java IF Statement/Exemple

Un exemplu de instrucțiune IF în care partea else lipsește:

```
int x = 5;
if (x == 5) x +=7;
System.out.println(x);
```

Output:
12

// variabila x este inițializată cu valoarea 5, după care se verifică egalitatea (x == 5), dacă rezultatul evaluării este true se execută x +=7, altfel se trece la următorul rând al programului, adică la System.out.println(x).

Java IF Statement/Exemple

Un exemplu de instrucțiune IF în care partea else lipsește:

```
int x = 3;
if (x == 5) x +=7;
System.out.println(x);
```

// variabila x este inițializată cu valoarea 3, după care se verifică egalitatea (x == 5), dacă rezultatul evaluării este true se execută x +=7, altfel se trece la următorul rând al programului, adică la System.out.println(x).

Java IF Statement/Exemple

Ce va afișa următoarea secvență de program?

```
int x = 3;
if (x % 2 == 0)
System.out.println("Numar par");
else System.out.println("Numar impar");
```

// variabila x este inițializată cu valoarea 3, după care se verifică dacă restul împărțirii numărului x la 2 este 0, atunci se afișează în consolă "Numar par" altfel se va afișa "Numar impar".

Java IF Statement/Probleme

1. Se consideră 2 numere întregi. Dacă primul număr este pozitiv, atunci de ridicat acest număr la pătrat, în caz contrar se calculează suma lor.

2. Se consideră un cerc cu raza R și un pătrat cu latura A. Să se scrie un program care determină dacă cercul încape în pătrat.

Java For Loop

Bucla FOR indică execuția repetată a unei instrucțiuni în funcție de valoarea unei variabile de control și are următoarea sintaxă generală:

```
FOR ([<ExpresieInitializare>];
 [<ExpresieCondiție>];
[<ExpresieActualizare>]) {
//instrucțiune sau bloc de cod spre execuție
}
```

Java For Loop

```
FOR ([<ExpresieInitializare>];
 [<ExpresieCondiție>]; [<ExpresieActualizare>]) {
//instrucțiune sau bloc de cod spre execuție
}
```

<ExpresieInitializare> este executată, o singură dată, înainte de execuția blocului de cod;

<ExpresieCondiție> definește condiția pentru executarea blocului de cod;

<ExpresieActualizare> este executată, de fiecare dată, după executarea blocului de cod.

Java For Loop/Exemple

```
for (int i = 0; i < 5; i++) {
  System.out.println(i);
}
```

Output:
0
1
2
3
4

Instrucțiunea i=0, stabilește o variabilă înainte de a începe bucla;

Condiția i<5, definește condiția de rulare a buclei (trebuie să fie mai mică de 5);

Expresia i++, incrementează valoarea variabilei i (i=i+1) de fiecare dată când blocul de cod din buclă a fost executat.

Java For Loop/Exemple

Calculați factorialul numărului *n*:

Exemplu:

```
int fact=1;
    for (int i=1; i<=5; i++) {
        fact = fact * i;
    }
System.out.println(fact);
```

Java For Loop/Exemple

Dacă există o buclă în corpul altei bucle, bucla se numește îmbricată.

Exemplu:

```
for (int i=1;i<=5; i++) {
   for (int j=1; j<=i; j++) {
   System.out.printf("%3d", i);
   // dacă înlocuim i cu j în S.o.p?
      }
   System.out.println();
  }
```

Output:

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Java For Loop/Probleme

1. Calculați suma: 1 + 3 + 5 + 7 + … + 2n-1, unde n este introdus de la tastatură.

2. Calculați suma: 0.1+0.2+0.3+...+1.8

3. Calculați $1 + \sqrt{1+2} + \sqrt{1+2+3} + \cdots + \sqrt{1+2+3+\cdots+n}$, unde n este introdus de la tastatură.

Java While Loop

Bucla while indică execuția repetată a unei instrucțiuni sau bloc de instrucțiuni în funcție de valoarea unei <ExpresiiCondiție> și are următoarea sintaxă generală:

```
while(<ExpresieCondiție>) <Instrucțiune>
//instrucțiune sau bloc de cod spre execuție
```

Java While Loop

while(<**ExpresieCondiție**>) <**Instrucțiune**>
//instrucțiune sau bloc de cod spre execuție

Dacă valoarea <ExpresieiCondiție> este true atunci se va executa instrucțiunea sau blocul de cod aflat în bucla while, în caz contrar, instrucțiunea sau blocul de cod ce se află în bucla while nu se va executa, ci se va trece la următoarea secvență din program.

Java While Loop/Exemple

```
int i = 0;
while (i < 5) {
  System.out.println(i);
 i++;
}
```

Output:
0
1
2
3
4

Instrucțiunea int i=0, inițializează valoarea zero pentru variabila i;

Condiția i<5, definește condiția de rulare a buclei (trebuie să fie mai mică de 5);

Expresia i++, incrementează valoarea variabilei i (i=i+1) de fiecare dată când blocul de cod din buclă a fost executat.

Java While Loop/Exemple

```
int i = 0;
while ( i < 5) {
  System.out.println(i);
    i += 2;
}
```

Output:
0
2
4

Expresia i+=2, crește valoarea variabilei i cu 2 (i=i+2).

Java While Loop/Exemple

Dacă există o buclă în corpul altei bucle, bucla se numește îmbricată.

Exemplu:

```
int i = 1;
while (i<=5) {
   int j = 1;
   while (j<=i) {
   System.out.printf("%3d", i);
   j++; }
   System.out.println();
   i++;   }
```

Output:

```
1
2  2
3  3  3
4  4  4  4
5  5  5  5  5
```

Java While Loop/Probleme

1. Să se afle toate numerele de trei cifre, fiecare având suma cifrelor egală cu numărul natural dat *n*.

Link-uri utile

1. https://www.slideshare.net/m_gutu/afiarea-informaiei-alfanumerice-i-citirea-datelor-de-la-tastatur-n-java
2. https://www.slideshare.net/m_gutu/java-if-statement
3. https://www.slideshare.net/m_gutu/java-for-loop-221703781
4. https://www.slideshare.net/m_gutu/java-while-loop-225654400
5. https://www.youtube.com/watch?v=oV58qTuN7mA&t=28s

Mult succes!