

Less
is
More

LESS IS YOURS

Curs 1

!!! Se poate veni și cu seră c la curs / laborator + nu se face prezentă.

I Introducere OOA

• Decompoziția OO

Descompunem sistemul într-un set de obiecte. Datele și structura algoritmului sunt spart în mai multe clase și structuri.

Def: Este o metodă de implementare a programelor în care acestea sunt organizate ca și colțuri de obiecte.

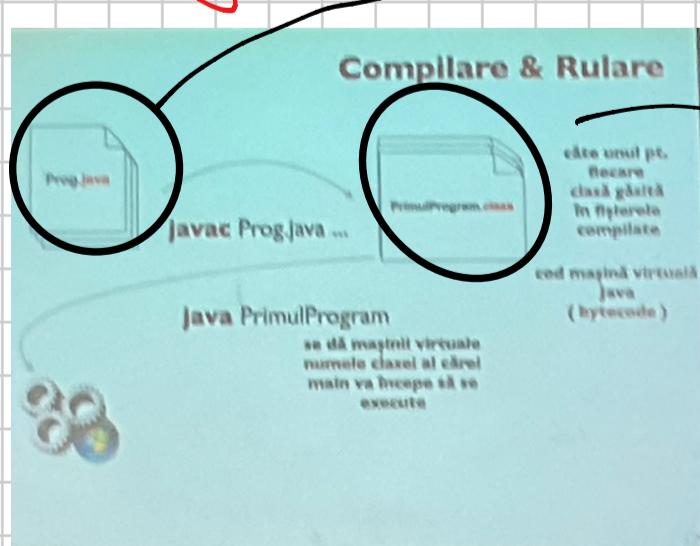
Limbajul folosit este Java.

De ce Java?

Este limbaj de top și mai simplu pt introducerea în OOA.

! Java de la 8 în sus. !

Bytecode → cod maximă virtuală Java



program main

→ program secundar (h-ul din C++). Apelăm o clasă din programul secundar în main.

Nu există tipuri de date primitive.

Ex: short int, long long, etc.

Variabilele pot începe cu litera -, -, \$.

Se declară variabile locale în clase, și toate valoarele trebuie initializat.

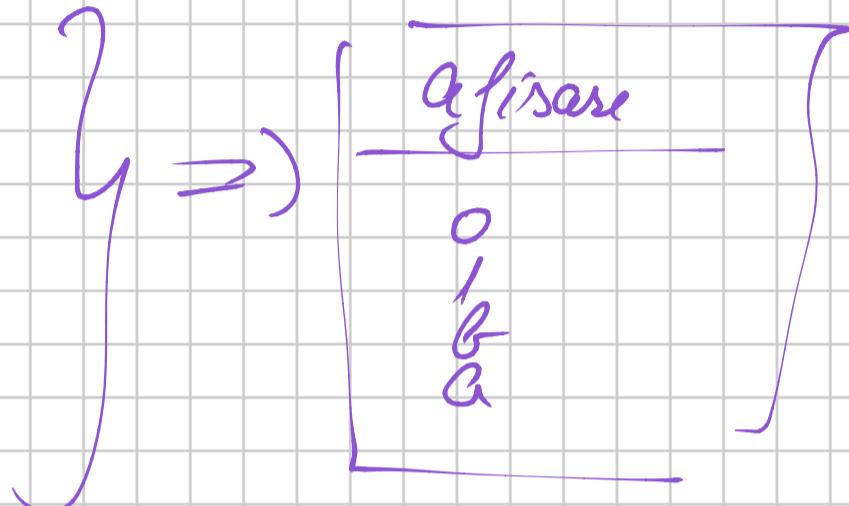
int $a = 0$; System.out.println ($a++$)

System.out.println (a)

char $c = 'b'$

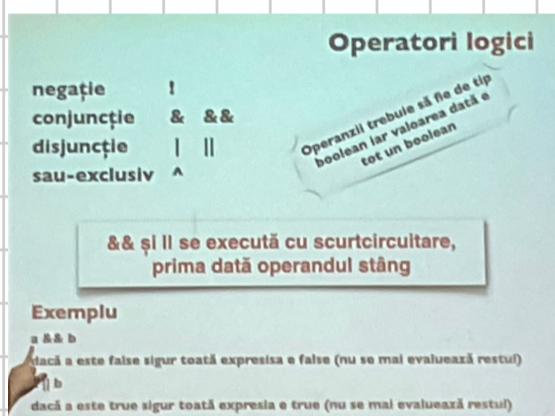
System.out.println ($c--$)

System.out.println (c)



Năcă unul dintre operanți e string, automat și celălalt e convertit la string.

System.out.println ($a + 1 + " = a "$) $\Rightarrow [11 = a]$



- Ex operator de atribuire

$$a = a + 2 \Leftrightarrow a += 2.$$

$$\begin{array}{l} a = 2 \quad b = 10 \quad c \\ c = a = b \\ c = a += b \end{array} \quad \Rightarrow \quad \begin{array}{l} c = 10 \quad a = 10 \quad b = 10 \\ c = 12 \end{array}$$

La fizare case se pune sfârșit de execuție.

II Clase și obiecte

Un obiect este o instanță a unei clase.

O clasă definește un set de obiecte.

Clasa definește implementarea unui obiect.

```
class clock {  
    private int ora, minut, secunda;  
}
```

În Java nu trebuie eliberată memoria, de acesta îl ocupe colectorul de deșeuri.

- Invocarea operatorilor

```
Clock clock1;
```

```
clock1 = new Clock();
```

```
clock1.setTime (10, 0, 0);
```

setTime este o funcție din clasă. Deși în clasa funcția este pe public.

Variabilele din clock pot fi modificate apoi clasa sunt pe public.

Nu ești foarte să pun this în fișă.

Aliasing → preluarea datelor dintr-un obiect și punerea lor în alt obiect.

Clock class 1, class 2;

class 1 = new Clock();

class 2 = new Clock();

class 1. setTime (10, 0, 0);

class 2 = class 1 ⇒ Aliasing

Mai multi pointeri care referă la fel.

Constructor

1. cu același nume ca și clasa.

2. nu cu tip returnat

3. și folosește la encapsul pentru a initializa obiectul
Dacă am un constructor trebuie să fie și clată

Clock class 1;

class 1 = new Clock();

Fără constructor

Clock class 1;

class 1 = Clock(10, 0, 0);

Cu constructor

1. Pentru a defini un constructor, constructorul trebuie să ai altă acelasi nume ca și clasa
2. Constructorul nu returnează nimic, tipul de date o să fie void.
3. Constructorul se folosește pentru a inițializa obiectul

Cum se defineste un constructor.

class student {

private int numar;

public void student (int nr)

numar = nr;

}

}