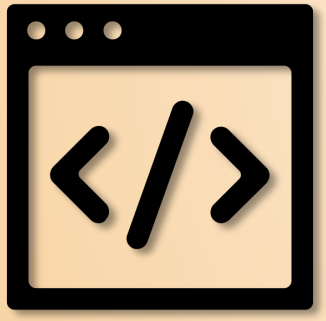
Laborator 1 PPA

Saptamanile: IV – V

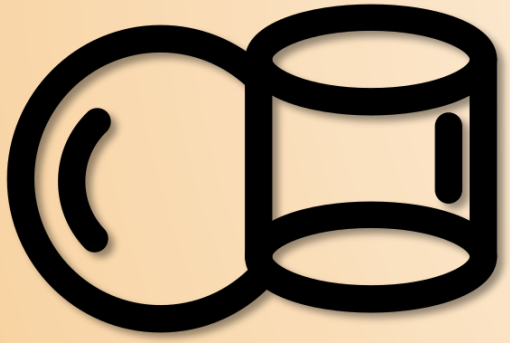
Interval Orar: 17-19

Ce este Java?



- Limbaj multi-**paradigma**, dar puternic Obiect Orientat (**OOP**).
- **Independent** de platforma pe care ruleaza (“Write once, run everywhere”).
- Limbaj hibrid: **Compilat** si **Interpretat**.

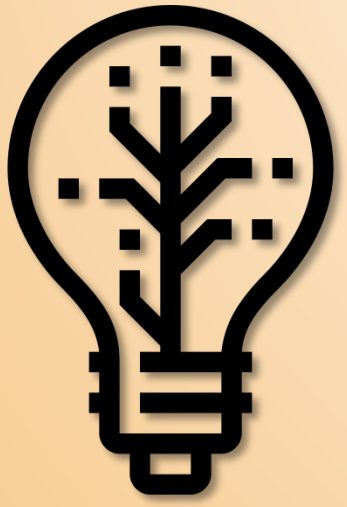
Ce este OOP?



- Paradigma orientata catre **date/obiecte**.
- Formata din:
  - **Clase** – tipuri de date definite de useri
  - **Obiecte** – instante/variabile concrete ale unei clase
  - **Metode** – functiile din interiorul clasei
  - **Atribute** – variabilele din interiorul clasei

Principiile OOP

1. Incapsulare
2. Abstractizare
3. Mostenire
4. Polimorfism



Incapsulare 

- Toate informatiile necesare sunt **incluse** intr-un **obiect**, dar sunt **expuse preferential**.
- Ex:
  - Log-in-ul pe Instagram/Facebook se face in baza e-mailului si a unei parole, nu pe tot profilul utilizatorului (nume, adresa, loc de munca etc).

Abstractizare

- **Ascunde** informatiile redundante in spatele unui strat numit **clasa abstracta/interfata**.
- Ex:
  - Operatiile de la un ATM se efectueaza prin apasarea unor butoane, dar in “spate” se efectueaza sute de operatii.

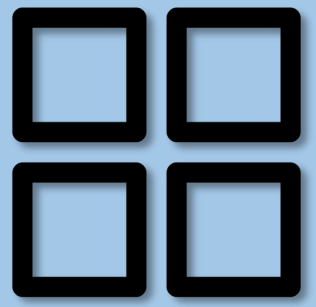
Mostenire

- **Reutilizarea codului** dintr-o alta **clasa**, fara a fi nevoie sa o **rescriem**.
- De obicei se **extinde** comportamentul existent al clasei.
- Ex:
  - Un copil va mosteni o parte din calitatile parintilor, dar va adauga, la randul lui, alte calitati.

Polimorfism

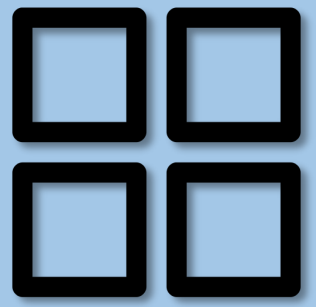
- Metodele cu **acelasi nume** pot face **operatii diferite** in functie de clasa de care apartin.
- Ex:
  - Cainele si pisica sunt ambele mamifere, dar se comporta si comunica in moduri diferite
  - Pisica miauna, cainele latra, dar ambele entitati comunica.

Aplicatia 1



- Se va crea o clasa Animal, avand urmatoarele:
  - name, color – atribute de tip String
  - age – atribut de tip Integer
  - Constructori
  - Getteri si Setteri
  - 3 metode abstracte:
    - eat(); talk(); mood();

Aplicatia 1



- Se vor crea 2 clase concrete (Dog; Cat) care vor **extinde** clasa Animal.

- Se vor instantia 2 obiecte (unul din fiecare clasa concreta) folosind metoda Upcasting si se va apela cateva metoda din fiecare obiect.

Aplicatia 2

- Folosind clasele definite anterior se va scrie, in MainActivity.class, cod Java care va:
  - Prelua textul scris in EditText
  - Folosi, intermediar, pentru text obiectele de tip Dog sau Cat
  - Scrie textul ales in TextView-uri cu ajutorul getters de la punctul anterior.

Aplicatia 2

- Folosind clasele definite anterior se va scrie, in MainActivity.class, cod Java care va:
  - Crea un listener pentru un buton care va face operatiile necesare **doar** atunci cand este apasat
  - *Optional: Metoda de tip listener va fi restransa la un lambda expression.

Fin.
The End.
Sfarsit.