

# Curs 4 POO

© Conf. univ dr. Crenguta M. Puchianu

---

- ☐ Obiecte
- ☐ Atribute si operatii in UML
- ☐ Concepte
- ☐ Clase. Declarare
- ☐ Constructori
- ☐ Accesarea variabilelor instantata
- ☐ Apelarea metodelor instantata

# Obiecte

- 
- ❑ Un **obiect** reprezintă un mod simplificat de a identifica un lucru sau o entitate din lumea reală sau imaginată.
  - ❑ Din punctul de vedere al paradigmei pe obiecte, un obiect este o combinație dintre:
    - o *informație de stare* reprezentată de o mulțime de *attribute* sau proprietăți ale obiectului și
    - un *comportament* reprezentat de un set de *operații* ce acționează asupra atributelor obiectului și eventual asupra altor obiecte.
  - ❑ Un **atribut** este o abstractizare a unei proprietăți a unui obiect din lumea reală. De exemplu, magazinul Profi, ca orice alt magazin, se caracterizează prin următoarele proprietăți: denumire, ora de deschidere și ora de închidere.
- In limbajul UML, un atribut se caracterizează prin: nume, tip, valoare și eventual constrângeri. Tipul unui atribut specifică domeniul de valori în care atributul ia valori.
-

# Attribute

- ❑ Valorile curente ale tuturor atributelor unui obiect constituie **starea** obiectului. De exemplu, starea obiectului magazinProfi este următoarea:  
denumire="Profi City"  
oraDeschidere=10  
oraInchidere=18
- ❑ Din punct de vedere conceptual, attributele se împart în două categorii: **attribute intrinseci** și cele **extrinseci**.
- ❑ **Attributele intrinseci** ale unui obiect sunt calități inerente ce pot fi percepute sau măsurate în lumea reală, precum: formă, culoare, dimensiune, sunet, miros, lungime, greutate, și caracterizează obiectul respectiv.  
Attributele intrinseci au proprietatea importantă că depind numai de obiectul în care există și sunt independente de existența altor obiecte.
- ❑ **Attributele extrinseci** ale unei entități nu sunt inerente și depind în totalitate sau parțial de alte obiecte, cu care obiectul respectiv este legat.

# Operatii

- ❑ O operație este un algoritm privat al obiectului, adică se execută în mediul său și care operează asupra valorilor atributelor sale sau ale altor obiecte de obicei pentru a furniza un serviciu unui alt obiect numit client.
- ❑ Operațiile se deduc din responsabilitățile obiectului pe care trebuie să le îndeplinească. De exemplu, obiectul magazinTomis trebuie să furnizeze acces la ora de închidere a magazinului, adică să furnizeze informații de stare ale obiectului, deoarece un alt obiect are nevoie de această informație pentru a o afișa utilizatorului.

În plus, utilizatorul programului poate să modifice ora de închidere a magazinului, modificând astfel starea obiectului magazinTomis. Sau un alt obiect ar vrea să știe care este întregul orar al magazinului. Așadar, obiectul magazinProfi va avea trei operații descrise în UML astfel:

- + obtineOraInchidere(): Integer
- + modificaOraInchidere(nouaOra: Integer)
- + obtineOrar(): String

Interfata naturala a unui obiect = multimea atributelor si operatiilor la care sunt accesibile din exteriorul obiectului respectiv.

# Concepte

- ❑ Dpdv conceptual, un obiect = instanța a unui **concept**

## Magazin

- ❑ Orice concept poate fi descris informal de trei elemente:
  - nume (cuvântul folosit când ne referim la un anumit concept),
  - semantică (înțeles) și
  - Extensie = mulțimea tuturor instanțelor.

Categorie	Exemple de concepte
tangibil	persoană, mașină, scaun
intangibil	timp, firmă
ființă din lumea reală	persoană, pisică, câine
lucru din lumea reală	artefact, mașină, casă, floare
descriere a unui alt obiect	marcă de mașină, specificația unei aplicații OO, documentație de proiect
organizație	firmă, agenție de voiaj, bancă, fundație
loc	magazin, port, oraș
rol în scenarii reale sau imaginare	pilot, administrator, profesor, medic, proprietar, autor, student, angajat

# Categorii de concepte

relațional	căsătorie, asociație familială
tranzacție	vânzare, rezervare, cumpărare, contractare, închiriere
elementele ce compun o tranzacție	articol dintr-o vânzare de mai multe produse, rezervarea pentru o persoană în rezervarea unei excursii în grup
eveniment	trimitere mesaj, revizie, vizita medicală, înregistrare automobil
proces	achiziție on-line a unor produse, înscriere la universitate, admitere
agregat	clasă de elevi, fereastră de componente grafice, mașină, masă, roi de albine
entitate conținută într-un agregat	elev într-o clasă, buton într-o fereastră, motor, picior de masă, albină
dispozitiv	vitezometru, modem, înregistrator de casă
sistem extern	sistemul informatic al serviciului personal, sistemul de contabilitate, sistemul de autorizare a utilizării unui card bancar
înregistrări de contracte sau evenimente	factură, log-are, memorarea sosirii unui mesaj în sistem, comandă de achiziție
concepte abstracte	calendar gregorian, integrală, mulțime, ecuație

# Clase de obiecte

- ❑ O **clasă** este o descriere a unei mulțimi de obiecte care au aceleași attribute, aceleași operații, aceleași relații cu alte clase și aceeași semantică. (clasa=concept)
- ❑ Clasele unui program orientat spre obiecte sunt identificate în primul rând din analiza domeniului problemei ce va fi rezolvată de către program. Domeniul problemei este constituit din porțiunea lumii reale pe care analistul o analizează.
- ❑ Exemplu. În cazul proiectului Magazin, obiectele fiecărei clase identificate până acum sunt caracterizate de o serie de attribute intrinseci pe care le enumerăm în continuare:

Magazin: denumire, oraDeschidere, oraInchidere

Factura: numar (unic), dataEmiterii

SpecificatieProdus: denumire, pretUnitar

Articol: cantitate

Vanzare: tva, total

Persoana: nume, prenume, adresa

ListaProduce:

Magazin
- denumire : String - oraDeschidere : Integer - oraInchidere : Integer
+ obtineOraInchidere() : Integer + modificaOraInchidere(nouaOra : Integer) : void + obtineOrar() : String

# Declararea unei clase in Java

---

*DeclaratieClasa ::= [Adnotare] [**public** | **protected** | **private**] [ **abstract** | **static** | **final** | **strictfp**] **class** Identificator [<ListaParametriTip>]  
[**extends** IdentificatorClasa [<ListaParametriTip>]] [**implements**  
ListaIdentificatoriInterfata] Bloc  
*ListaParametriTip ::= ListaParametriTip , ParameterTip | ParameterTip*  
*ListaIdentificatoriInterfata ::= ListaIdentificatoriInterfata,  
IdentificatorInterfata | IdentificatorInterfata**

```
public class Magazin{  
    private String denumire;  
    private int oraDeschidere, oraInchidere;  
    public int obtineOraInchidere(){return oraInchidere; }  
    public void modificaOraInchidere(int nouaOra){ oraInchidere=nouaOra;}  
    public String obtineOrar(){  
        return oraDeschidere+":00 - "+ oraInchidere+":00 ";  
    }  
}
```

---



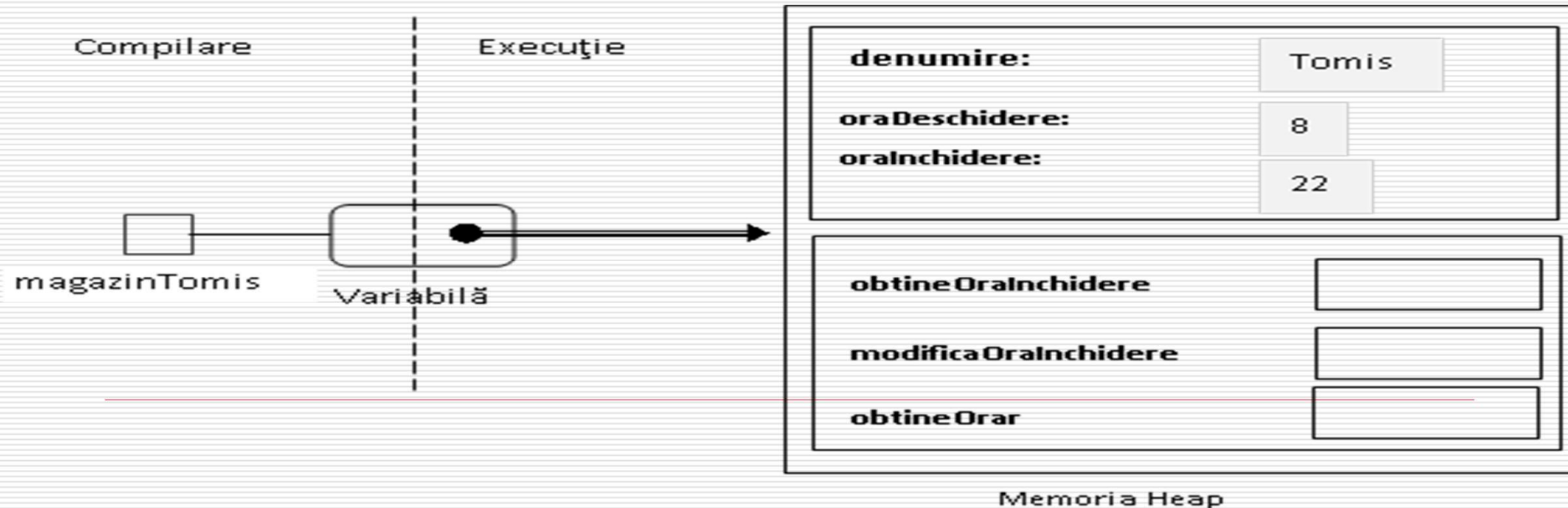
# Construirea obiectelor in Java

---

- ❑ Obiect = instanta a unei clase
- ❑ Un **constructor** al unei clase este o metodă cu același nume cu cel al clasei din care face parte și căreia îi lipsește din declarație tipul rezultatului.
- ❑ Un constructor descrie cum se inițializează variabilele instanță ale noului obiect.
- ❑ O clasă poate să nu aibă un constructor definit de programator. În acest caz, compilatorul oferă automat clasei un **constructor predefinit** fără parametri care inițializează variabilele instanță ale clasei astfel: variabilele primitive numerice cu zero, variabilele booleene cu false și cele referință cu null.
- ❑ Constructorii, ca orice altă metodă, pot fi supraîncărcați, ei diferind prin numărul și tipul parametrilor.

# Construirea obiectelor in Java (cont.)

- ❑ Într-un constructor, putem apela un alt constructor al aceleiași clase.  
**this([parametriActuali])**
- ❑ Constructorul este apelat atunci când dorim să creăm un obiect al clasei în care este definit constructorul:  
**new NumeClasa(parametriActuali);**  
Exemplu: Magazin magazinTomis=new Magazin("Profi", 8, 22);



# Accesarea/Apelarea variabilelor/metodelor instanța

---

- ❑ Accesul la o variabilă instanță (adica, a unui obiect) se face cu ajutorul operatorului binar punct ( "." ).  
Primul operand trebuie să fie o variabilă referință sau o expresie a cărei valoare este de tip referință. Al doilea operand este chiar numele variabilei-instanță.

Exemplu: magazinTomis.oraDeschidere=8;

- ❑ Apelarea unei metode instanță se face cu ajutorul operatorului binar "." .  
Primul operand al apelului trebuie să fie o variabilă referință sau o expresie a cărei valoare să fie de tip referință către un obiect în memorie. Al doilea operand este numele metodei urmat de lista parametrilor actuali.

Exemplu: magazinTomis.modificaOraInchidere(23);

---