TEHNOLOGII JAVA
PENTRU DEZVOLTAREA APLICAȚIILOR
LUCRARE DE LABORATOR 5

Clase şi obiecte în Java.
Crearea obiectelor. Constructori. Variabile clasă

I. SCOPUL LUCRĂRII

Lucrarea de față are rolul de a prezenta şi familiariza studentul cu noțiuni de bază ale limbajului Java: clase şi crearea obiectelor, rolul constructorilor, semnificația şi utilizarea variabilelor clasă.

La sfârşitul acestei lucrări, studentul va avea posibilitatea să scrie programe Java în care să folosească noile noțiuni.

II. NOȚIUNI TEORETICE

1. Crearea obiectelor

Vezi Curs 5, secțiunea 4.1.

Să considerăm următorul exemplu:

```
public class Vehicul
{
      public String categoria;
      public String marca;
      public int nrRoti;
      public boolean aerCond;
      public boolean volan;

      public Vehicul(String nume,boolean aerCond) {
        categoria="automobil";
        marca=nume;
        nrRoti=4;
        this.aerCond=aerCond;
        volan=true;
      }
      public Vehicul(String nume) {
        categoria="bicicleta";
        marca=nume;
        nrRoti=2;
        aerCond=false;
        volan=false;
      }
      public void afisare() {
         System.out.println("Categorie vehicul "+contor+": "
                            +categoria);
```

```
            System.out.println("Denumire "+categoria+": "+marca);
            System.out.println("Nr roti: "+nrRoti);
            System.out.print("Aer conditionat: ");
            if(aerCond)
                        System.out.println("da");
            else
                        System.out.println("nu");
            System.out.print("Volan: ");
            if(volan)
                        System.out.println("da");
            else
                        System.out.println("nu");
            System.out.print("\n");
      }

      public static void main(String args[ ]) {

            Vehicul b1,m1,m2;
            b1=new Vehicul("Pegas");
            m1=new Vehicul("Dacia",false);
            m2=new Vehicul("Cielo",true);
            b1.afisare();
            m1.afisare();
            m2.afisare();
      }
}
```

Observații:

Un fişier sursă java care conține o clasă publică trebuie salvat cu numele clasei respective la care se adaugă extensia *.java.*

Într-un fişier sursă java poate exista cel mult o clasă declarată publică.

Astfel programul de mai sus se va salva într-un fişier cu numele **Vehicul.java**. Altfel, la compilare se va semnala eroare.

Clasa **Vehicul** conține doi constructori. Utilizarea mai multor constructori pentru aceeaşi clasă se recomandă atunci când se urmăreşte inițializarea obiectelor clasei în moduri diferite. Astfel, în **main( )** se vor crea obiecte (vehicule) ce reprezintă biciclete şi automobile, în funcție de utilizarea unui constructor sau al altuia.

Unul din contextele în care utilizarea notației bazate pe **this** (**this** este referința către obiectul curent) este necesară, este atunci când se defineşte un parametru sau o variabilă locală cu acelaşi nume ca o dată membru a clasei. Astfel, pentru a se face diferența între variabila locală sau parametru şi data membru a clasei, aceasta din urmă este accesată explicit prin referința **this (**vezi primul constructor al clasei **Vehicul**: `this.aerCond=aerCond`**)**.

Într-un constructor se poate referi ca primă instrucțiune un constructor al clasei curente. Este singura poziție din definiția unui constructor în care poate să apară o referire la un alt constructor. Un astfel de apel apare sub forma:

```
this(listaDeArgumente); // constructor din aceeasi clasa
```

unde **listaDeArgumente** reprezintă lista valorilor parametrilor constructorului invocat.

Exemplu:

```
public class Floare
{
      public String denumire;
      public int nrPetale;
      public String zonaClima="-";

      public Floare(String denum,int nrP) {
        denumire=denum;
        nrPetale=nrP;
      }
      public Floare(String denum,int nrP,String clima) {
        this(denum,nrP);
        zonaClima=clima;
      }
      public void afisare() {
          System.out.println("Denumire: "+denumire);
          System.out.println("Numar petale : "+nrPetale);
          System.out.println("Zona climatica: "+zonaClima+"\n");
      }

      public static void main(String args[ ]) {

          Floare floare1=new Floare("narcisa",7);
          Floare floare2=new Floare("orhidee",30,"tropicala");
          floare1.afisare();
          floare2.afisare();
      }
}
```

Observații:

Variabilele membru se pot inițializa când sunt declarate. Se recomandă ca inițializarea să se facă în constructor (în principal, pentru ca fiecare obiect al clasei să poată avea valori diferite pentru datele membru). În cazul în care o dată membru membru este inițializată la declarare iar apoi şi în constructor, atunci valoarea sa va fi cea pe care o primeşte în constructor. Dacă în constructor nu mai primeşte nici o valoare, atunci valoarea sa va rămâne cea de la declarare.

Astfel, obiectul **floare1** va avea pentru data membru **zonaClima** valoarea "-" deoarece în primul constructor (cel care este utilizat pentru crearea acestui obiect), aceasta dată membru nu primeşte nici o valoare. Obiectul **floare2**, în schimb, este creat cu ajutorul celui de-al doilea constructor, care inițiaziează data membru **zonaClima** la valoarea celui de-al treilea parametru al sau – care este "tropicala".

2. Variabile clasă

Vezi Curs 5, secțiunea 4.2.

Să considerăm clasa **Vehicul** prezentată în secțiunea anterioară. O vom modifica adăugându-i o dată membru statică numită **contor** care numără obiectele clasei ce sunt create în program. Această nouă variabilă, fiind declarată **static**, este asociată clasei, ci nu instanțelor clasei. O variabilă clasă poate fi accesată prin numele clasei, deci nu este nevoie de crearea unei instanțe a clasei doar pentru a o accesa. O variabilă statică este creată o singură dată, la încărcarea clasei.

```
public class Vehicul
{
      static int contor=0;
      public String categoria;
      public String marca;
      public int nrRoti;
      public boolean aerCond;
      public boolean volan;

      public Vehicul(String nume,boolean aerCond)
      {
        categoria="automobil";
        marca=nume;
        nrRoti=4;
        this.aerCond=aerCond;
        volan=true;
        contor++;
      }
      public Vehicul(String nume)
      {
        categoria="bicicleta";
        marca=nume;
        nrRoti=2;
        aerCond=false;
        volan=false;
        contor++;
      }
      public void afisare()
      {
          System.out.println("Categorie vehicul "+contor+": "
                              +categoria);
          System.out.println("Denumire "+categoria+": "+marca);
          System.out.println("Nr roti: "+nrRoti);
          System.out.print("Aer conditionat: ");
          if(aerCond)
                    System.out.println("da");
          else
                    System.out.println("nu");
          System.out.print("Volan: ");
          if(volan)
                    System.out.println("da");
          else
                    System.out.println("nu");
          System.out.print("\n");
      }

      public static void main(String args[ ]) {

          Vehicul b1,m1,m2;
          b1=new Vehicul("Pegas");
          b1.afisare();
          m1=new Vehicul("Dacia",false);
          m1.afisare();
          m2=new Vehicul("Cielo",true);
```

```
            m2.afisare();
            System.out.println("\nExista "+Vehicul.contor+" obiecte
                                ale clasei Vehicul");
      }
}
```

Să considerăm un alt exemplu:

```
class Salariat
{
      public static final double impozit=0.15;
      public String nume;
      public String functia;
      public int varsta;
      public double salariuBrut;

      public Salariat(String num,String func,int ani,double sal)
      {
        nume=num;
        functia=func;
        varsta=ani;
        salariuBrut=sal;
      }
      public double salariuNet()
      {
          return salariuBrut*(1-impozit);
      }
      public void afisare()
      {
       System.out.println("Nume salariat: "+nume);
       System.out.println("Functia: "+functia);
       System.out.println("Varsta: "+varsta);
       System.out.println("SalariuBrut: "+salariuBrut+" lei");
       System.out.println("SalariuNet: "+salariuNet()+" lei \n");
      }
}

public class UtilSal
{
  public static void main(String args[ ])
  {
    Salariat s1=new Salariat("Popa Ion","contabil",36,7300000);
    Salariat s2=new Salariat("Rada Alina","secretara",28,6000000);
    System.out.println("Impozit: "+(int)(Salariat.impozit*100)+"% \n");
    s1.afisare();
    s2.afisare();
  }
}
```

Observații:

Am utilizat în clasa **Salariat** o dată membru **impozit** care este declarată şi static şi final, ceea ce înseamnă că este considerată o constantă de către compilator (valoarea sa nu poate fi modificată).

Programul de mai sus trebuie salvat într-un fişier sursă cu numele *UtilSal.java.*

III. MODUL DE LUCRU

1. Se editează codul sursă al programului Java folosind un editor de text disponibil(de ex., se poate utiliza Notepad).
2. Se salvează fişierul cu extensia **.java**. Fişierul trebuie salvat la următoarea locaţie: **c:\JBulider7\jdk1.3.1\bin**
3. Compilarea mini-aplicaţiei Java se va face din linia de comandă. Se poate proceda astfel. Se deschide o fereastră MS-Dos: Start ->Run, se tipăreşte **command** în căsuţa de text şi se apasă butonul OK. Printr-o schimbare de directoare şi subdirectoare se trece la locaţia: **c:\JBulider7\jdk1.3.1\bin.** Sau, se lansează WindowsCommander. Se trece la locaţia **c:\JBulider7\jdk1.3.1\bin.** Se deschide o fereastră MS-Dos: Commander ->Run Dos.
4. Pentru compilare, se tipăreşte la prompter **javac nume_fişier_sursă.java** şi se apasă Enter. De ex., dacă fişierul se numeşte Test.java, se va scrie **javac Test.java**. În cazul în care programul conţine erori acestea vor fi semnalate şi afişate în fereastra MS-Dos, după care va apare iar prompter-ul. Dacă programul nu conţine erori şi compilarea se face cu succes, atunci va apare numai prompter-ul.
5. Pentru rularea aplicaţiei Java, se lansează interpretorul Java. Se tipăreşte la prompter următoarea comandă **java nume_clasă_care_conţine_main** şi se apasă Enter. De ex., dacă clasa din program care conţine metoda *main( )* se numeşte Test, se va scrie **java Test**.

IV. TEMĂ

1. Se vor parcurge toate exemplele prezentate în platforma de laborator testându-se practic.
2. Modificaţi clasa Salariat astfel încât salariul fiecărui angajat să poată fi exprimat şi în dolari. De asemenea, programul să afişeze şi cursul de schimb al dolarului utilizat în calcule.
3. Scrieţi clasa Student. Fiecare student are un nume, an, grupă, şi două note obţinute la o anumită materie - una pe semestrul 1 iar cealaltă pe semestrul 2. Clasa conţine un constructor ce iniţializează datele membru ale clasei la valorile parametrilor săi, o funcţie care calculează şi returnează media celor două note, şi o funcţie de afişare ce afişează valorile datelor membru şi valoarea mediei. Scrieţi un program complet în care să utilizaţi obiecte ale clasei Salariat.