

## TEHNICI AVANSATE DE PROGRAMARE

### Referat 1

#### TEMA 1.

Să se scrie un program pentru un medic veterinar care dorește să țină înregistrările unor rase de câini pe care îi tratează precum și bolile de care suferă aceștia. În mod concret, veterinarul dorește să vadă ce tipuri de boli afectează câinii cu pete și pe cei fără pete. Un mod de abordare ar fi crearea a două subclase de câini, o clasă pentru câinii cu pete (**CaineCuPete**) și alta pentru cei fără pete (**CaineFaraPete**), care să moștenească clasa părinte **Caine**.

Aplicatia va avea urmatorul schelet :

```
class Caine{
    protected String rasa;
    protected String culoare;
    protected int inaltime;
    protected int greutate;
    public Caine(String rasa, String culoare, int inaltime, int greutate);
    public void arataGreutatea();
    public String arataGreutatea();
}
```

Clasele derivate adauga noi noi campuri de date si metode respectiv specializeaza anumite metode ale clasei de baza. Astfel clasa *CaineCuPete* va avea cinci campuri de date (...., CuloarePata) si doua metode noi. In cazul clasei *CaineFaraPete* apare o data in plus fata de clasa parinte (*static int NrCainiFaraPete=0*).

```
class CaineCuPete extends Caine{
    public String CuloarePata;
    // constructori
    .....
    public String arataGreutatea();
    public void arataRasa();
    void informatiiPata();
}

class extends CaineFaraPete{
    static int NrCainiFaraPete=0;
    // constructori
    .....
}
```

Sa se afiseze intr-o noua clasa numita **TestCaini** numarul cainilor care nu au pete.

**TEMA 2.**

Să se definească clasele **NumarReal**, **NumarIntreg** și **NumarComplex** derivate din clasa abstractă **Numar** ținând cont de următorii membrii ai claselor:

```
abstract class Numar {
    static int NumarInstante=0;
    public abstract void vizualizare();
    .....
}
class NumarReal extends Numar {
    public float valoare;
    .....
}
class NumarIntreg extends Numar {
    public int valoare;
    .....
}
class NumarComplex extends Numar {
    public double re, im;
    .....
    public NumarComplex suma(Complex z);
    public NumarComplex produs(Complex z);
    public double modul();
    public String toString();
}
```

Definiți constructorii corespunzatori pentru clasele de mai sus. Clasa care descrie corpul numerelor complexe va avea trei constructori: implicit, cu un parametru și cu doi. Implementați metodele pentru suma, produsul a două numere complexe și modulul unui număr complex. Să se testeze aplicația într-o clasă nouă cu numele **TestNumere** creând câteva instanțe pentru fiecare clasă definită. Verificați toate metodele implementate și afișați numărul de obiecte create în program.

**TEMA 3.**

Realizați o aplicație Java care simulează operațiunile asupra unui cont bancar. Implementarea trebuie să conțină cel puțin următoarele *attribute*:

- numărul contului;
- tipul contului;
- numele și adresa clientului;
- suma de bani existentă;
- dobânda.

Definiți *constructori* cu parametrii pentru inițializarea:

- numărului contului;
- tipului contului;
- numele și adresa clientului.

Ca metode, descrieți:

- depunerea unei sume de bani;
- retragerea unei sume de bani;
- aflarea unei sume de bani curente(sold curent);
- aflarea dobânzii curente.

Banca are două tipuri de conturi pentru persoane fizice:

- cu dobândă la vedere(cu posibilitate de operațiuni zilnice)
- la termen ( are loc reținerea sumei de bani până la termenul stabilit)

Suma de bani dintr-un cont la vedere poate chiar să scadă( există posibilitatea dobânzii negative), dar nu și suma de bani dintr-un cont la termen.

La sfârșitul fiecărei luni/an, la suma inițială se adaugă suma de bani obținută din dobânda contului de la scadență.

Se vor scrie clase Java corespunzătoare fiecărui tip de cont bancar. Utilizând un atribut static, numărați câte persoane fizice au cont deschis în cadrul băncii respective.

#### TEMA 4.

Realizați o aplicație Java care simulează următorul joc:

*Presupunem că avem doi jucători care folosesc mâna dreaptă pentru reprezentarea a trei obiecte:*

- *hârtie: înseamnă palma întinsă;*
- *pumn: înseamnă mana strânsă;*
- *foarfece: înseamnă două degete depărtate(semnul victoriei)*

*Jucătorii își arată simultan mâna dreaptă în una din aceste configurații(de mai multe ori). Dacă ei arată același lucru, este remiză(nu câștigă). Dacă nu, se aplică una din următoarele trei reguli:*

- *Hârtia acoperă pumnul(deci palma întinsă câștigă față de palma strânsă);*
- *Pumnul sparge foarfecele;*
- *Foarfecele taie hârtia.*

Să se simuleze acest joc, făcând un număr arbitrar de evenimente și precizând scorul final. Se cere să se joace persoană-calculator și varianta a doua, calculator-calculator.

#### TEMA 5 .

Realizați un program Java care să modeleze conceptul de telefon mobil. Considerăm că un telefon mobil este caracterizat prin următoarele atribute (variabile de instanță):

- firmă producătoare (string);
- tip telefon (string);
- rețea (string);
- polifonic/monofonic (int) – număr canale;
- display color/ monocrom (int) – număr culori;
- tip display (string) – (e.g. OLED, TFT etc.)

- bluetooth/infrared /port USB(boolean);
- timp standby (int) – număr ore;
- timp convorbire(int) – număr ore;
- mp3 player (boolean);
- mesagerie/jocuri (string);
- ceas/alarmă/vibrații (boolean), etc.

Deasemenea, orice potențial cumpărător al unui telefon mobil este interesat dacă acesta este dotat sau nu cu cameră foto/video. Atributele unei camere foto/video sunt:

- rezoluție (float) - mp;
- blitz (boolean);
- încorporată/atașabilă (boolean);
- zoom digital (float);
- zoom optic (float);

Definiți constructorii pentru clasele *TelefonMobil* și *Cameră*. Utilizați variabile de instanță private, cu acces prin metodele publice set și get. Să se implementeze metode de afișare a caracteristicilor telefonului. Folosind o variabilă statică atașată clasei *TelefonMobil*, să se afișeze numărul de telefoane disponibile. Să se creeze o clasă care să implementeze o metodă ce preia datele de intrare introduse de utilizator de la tastatură și afișează telefoanele care îndeplinesc dorințele acestora.

#### TEMA 6.

Modelați cu ajutorul limbajului Java următoarea problemă:

*O persoana este interesată în achiziționarea unui frigider. Considerăm că frigiderele de pe piața sunt caracterizate de următoarele elemente: marcă, tip, an\_fabricație, dimensiuni(lungime, lățime, inaltime), greutate, numar compartimente, garantie și preț.*

*Exemplu: Electrolux, Q14, 2004, 48 cm, 40 cm, 204 cm, 130 Kg, 6 compartimente, 3 ani, 625 \$.*

*Cumpărătorul mai este interesat și:*

- a) *dacă frigiderul are sau nu afisaj electronic;*
- b) *dacă frigiderul are sau nu compartiment special pentru racirea sticlelor;*
- c) *dacă frigiderul are sau nu capacitatea de a nu amesteca mirosurile;*

*Presupunem că avem în discuție șapte frigidere.*

*Trebuie definită o metodă care să stabilească, pentru fiecare frigider, gradul de performanță al acestuia (luând în calcul elementele caracteristice definite mai sus).*

#### TEMA 7.

Următorul program Java implementează un sistem de închiriere a diferitelor articole multimedia (casete audio/video, radio/casetofoane, etc). Fiecare articol este caracterizat prin: denumire, preț de închiriere pe zi, disponibil sau nu, data la care a fost închiriat, data când trebuie returnat. Se dorește să se știe în orice moment al programului câte articole există disponibile pentru închiriat și câte sunt deja închiriate.

```
public class Articol {

    int contor_disponibil=0;
    int contor_inchiriat=0;
    String denumire;
    double pret;
```

```

boolean disponibil;
String data_inchiriere
String data_returnare;

public Articol(String nume, boolean dispon, String data1,
                String data2)
{
    denumire=nume;
    pret=10000;
    disponibil=dispon;
    data_inchiriere=data1;
    data_returnare=data2;
    if (disponibil) contor_disponibil++;
    else contor_inchiriat++;
}

public void afisare_contor()
{
    System.out.println("Nr articole inchiriate = "
                       +contor_inchiriat);
    System.out.println("Nr articole disponibile = "
                       +contor_disponibil);
}

public static void main(String arg[])
{
    Articol.afisare_contor();
    Articol a1=new Articol("caseta audio Madonna", true,"12 mai",
                           "20 mai");
    Articol a2=new Articol("caseta video Mortal Weapon", false,
                           "1 iunie","2 iunie");
    Articol a3=new Articol("boxe audio",true,"1 august",
                           "3 august");
    Articol.afisare_contor();
}
}

```

a) Corectați eventualele erori din programul de mai sus. Argumentați modul de corectare.

b) Completați programul de mai sus astfel:

- implementați în clasa *Articol* două funcții care să simuleze returnarea unui articol închiriat și respectiv închirierea unui articol disponibil.
- creați o nouă clasă *Radio* care extinde clasa *Articol*. În clasa *Radio* definiți o dată membru *double pret* și un constructor corespunzător.
- creați o nouă clasă *RadioCasetofon* care extinde clasa *Radio*. În clasa *RadioCasetofon* definiți o dată membru *double pret*, un constructor corespunzător, și o funcție membru care să afișeze pretul de închiriere pe zi pentru un radio-casetofon, pentru un radio și pentru un articol generic.
- Completați metoda *main* astfel încât să se utilizeze codul nou creat.

**TEMA 8.**

Modelați cu ajutorul limbajului Java următoarea problemă:

O persoană este interesată de achiziționarea unui automobil. Știm că orice pasionat de automobile își îndreaptă atenția asupra motorului, fiind interesat de :

- număr cilindri (int),
- număr supape(int),
- putere maximă-cp(int),
- tip carburant(string),
- tip injecție(string),
- catalizator(string),
- sistem de alimentare(string).

Cumpărătorul mai este interesat de culoarea mașinii(string), firma producătoare (string) și de alte dotări :

- vopsea metalizată (boolean);
- aer condiționat (boolean);
- sistem de frânare(string);
- airbag șofer (boolean);
- faruri dublu optice (boolean);

Implementați clasele *Motor* și *Automobil* știind că aceasta din urmă moștenește clasa *Vehicul* care are următoarele atribute: număr roți (int), număr uși (int), număr locuri(int); capacitate rezervor (int) /portbagaj (int) u.m. : litru; dimensiuni: lungime (int)/lățime(int); înălțime(int) u.m.: mm; consum: urban/extraurban/mixt (int) u.m.:litri/100km.Să se implementeze constructorii corespunzători claselor definite anterior, să se declare metodele get și set pentru variabilele private ale claselor, să se realizeze o metodă de afișare a caracteristicilor fiecărui automobil. Într-o nouă clasă numită *Achiziție* declarați 7 obiecte de tip *Automobil*. Realizați o metodă care pe baza unor date de intrare citite de la tastatură să afișeze automobilul/automobilele care corespund cererii utilizatorului(potențialului cumpărător).

**TEMA 9.**

Să se realizeze un program de gestiune a unei biblioteci ținând cont de următoarele atribute (variabile de instanță) ale claselor:

```
public class Biblioteca{
    private Client[] ob1;
    private Carte[] ob2;
    .....
}
```

```
public class Client{
    private String nume;
    private String prenume;
    private String cnp;
    private int varsta;
    private int cod_unic_client;
    private int [] lista_coduri_carti;
```

```

.....
}
public class Carte{
    private final String titlu;
    private final String editura;
    private final String autor;
    private final int an_apariție;
    private int cod_unic_carte;
    private int nr_total_exemplare;
    private int nr_curent_exemplare;
    .....
}

```

**Cerințe:**

- Definiți constructorii corespunzători pentru clasele de mai sus;
- Utilizați pentru accesul la variabilele de instanță private metodele set și get;
- Implementați metode pentru :
  - împrumutarea unei cărți;
  - restituirea unei cărți;
  - afișare cărți existente în bibliotecă;
  - căutare carte după: nume/autor/editură/an\_apariție/cod\_unic;
  - căutare client după: nume\_client/cod\_client;

**TEMA 10.**

Să se realizeze o aplicație Java prin care este modelată modalitatea de plată a angajaților din cadrul unei firme. Considerăm următoarele date: firma are trei departamente:

*Secretariat, Conducere și Relații Cu Publicul.*

Pentru fiecare dintre aceste trei departamente se consideră caracteristicile: *nume\_angajat (String), prenume\_angajat (String), vârstă (int), cod\_angajat (int), vechime (int)*. Se știe că pentru secretariat, fiecare angajat are un spor salarial de 0.5% pentru fiecare an de vechime. Pentru conducere, sporul este de 1%/an, iar pentru relații cu publicul de 0.7%/an. În plus, departamentul conducere mai are un spor salarial de 0.1% pentru fiecare angajat din departamentele secretariat și relații cu publicul.

**Cerințe:** Să se realizeze statele de plată ale angajaților celor trei departamente, folosindu-se un număr oarecare de instanțe pentru fiecare dintre cele trei clase (e.g. *nume; departament; salariu*). Pentru descrierea claselor se vor folosi constructori, variabile de instanță private, după caz statice și/sau finale, metodele publice set și get precum și alte metode care se consideră a fi necesare.

**Indicație:** pentru a număra angajații din fiecare departament se pot folosi variabile statice care sunt incrementate la fiecare instanță. Pentru a stabili procentele salariale se pot utiliza variabile finale și statice.

**TEMA 11.**

Să se scrie un applet Java care să permită utilizatorului desenarea cu mouse-ul a diferitelor forme geometrice. Applet-ul va conține următoarele componente vizuale:

- o listă cu următoarele intrări: “linie”, “arc”, “cerc”, “dreptunghi”, “dreptunghi cu marginile rotunjite”, “poligon”. Din listă se va putea selecta o singură opțiune la un moment dat.

Lista va fi etichetată cu eticheta **Alegeti forma geometrica.**

- o componentă checkbox etichetată cu eticheta **Coloreaza (sau Fill)**
- un buton “Sterge ultimul desenat”
- un buton “Sterge tot”

Dacă din listă se va alege opțiunea “linie”, utilizatorul va putea desena o linie de **orice** dimensiune dorește, oriunde pe suprafața applet-ului alocată desenării.

Dacă din listă se va alege una din celelalte opțiuni, atunci prin efectuarea unui click, oriunde pe suprafața applet-ului alocată desenării, utilizatorul va desena forma geometrică aleasă din listă (de o dimensiune predefinită de programator). Dacă componenta checkbox este bifată în momentul desenării formei geometrice, atunci forma respectivă va fi colorată (altfel va avea doar contur).

La acționarea butonului “Sterge ultimul desenat”, se va șterge ultima formă geometrică desenată (indiferent dacă a fost linie sau altceva).

La acționarea butonului “Sterge tot”, se va șterge tot ceea ce s-a desenat până la acel moment.

## TEMA 12.

Creați un applet Java care să simuleze un joc puzzle cu numere sau cu imagini, de dimensiune 4x4 sau 3x3. Applet-ul va conține un buton *Start* (de pornire a jocului) un buton *Scramble* (de amestecare a numerelor/imaginilor) și un buton *Revenire* (revenirea la imaginea inițială). Applet-ul permite logarea fiecărui utilizator prin intermediul unui username și va reține pentru fiecare utilizator în parte timpul minim de rezolvare a unui puzzle.

## CERINȚE GENERALE :

1. Fiecare student va rezolva, “la alegere” (vezi punctul 2), numai **unul** din enunțurile propuse mai sus.
2. **Maxim** 3 studenți dintr-o subgrupă pot alege același enunț. Fiecare student **are obligația** până la data de **24 Martie 2007** să contacteze cadrul didactic care predă laboratorul și să anunțe/stabilească tema aleasă. Altfel, studentul riscă să rezolve o temă care nu îi va fi luată în considerare pentru acordarea notei la laborator.
3. Referatul trebuie să conțină:
  - o parte scrisă (2-3 pagini): descrierea claselor și metodelor, comentariu cod, Layout Managers utilizați (pentru applet-uri), ce componente grafice s-au utilizat (pentru applet-uri), cum s-au tratat evenimentele generate de componentele grafice (pentru applet-uri);
  - codul sursă al aplicației (pe suport magnetic).

## Termen pentru predarea și prezentarea proiectului:

- săptămâna 26-31 Martie 2007 (anul II Informatică, grupele 221A , 222A , 223A, 224A și 225A; anul III Matematică-Informatica, grupele 331A și 332 A)
- săptămâna 02-07 Aprilie 2007 (anul II Informatică, grupele 221B , 222B , 223B, 224B și 225B; anul III Matematică-Informatica, grupele 331B și 332 B)
- proiectele se vor preda și prezenta pe subgrupe la orele de laborator din săptămâna respectivă.