UNIVERSITATEA DIN BUCUREŞTI
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
DEPARTAMENTUL DE INFORMATICĂ
DOMENIUL DE STUDII INFORMATICĂ
SERIA 23

MODEL DE SUBIECT LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"

I. Pentru fiecare dintre cele 5 întrebări de mai jos, indicaţi variantele de răspuns pe care le consideraţi corecte:

1. Fie următorul program Java:

```
class C { public static int a=1; }
public class teste_grila {
   public static void main(String[] args) {
       C ob1 = new C();
       C ob2 = new C();
       ob1.a++;
       System.out.println(++ob2.a);
  }
}
```

După executarea programului, va fi afişată valoarea:

a) 2 b) 3 c) 1 d) nicio valoare, deoarece programul este incorect sintactic şi nu va putea fi executat

2. Considerăm următoarea metodă:

```
void test(){
   try{
       met();
   }
   catch (NullPointerException ex){
       System.out.print("NPE ");
   }
   catch (Exception ex){
       System.out.print("EX ");
   }
   finally{
       System.out.print("FIN ");
   }
   System.out.println("END");
}
```

După apelarea metodei `test()`, ce se va afişa dacă metoda `met()` va lansa excepţia `IllegalArgumentException`?

a) NPE FIN END
b) EX END
c) NPE EX FIN END
d) EX FIN END

3. După executarea secvenței de cod

```
String s = "abcbd";
String t = "Programare";
int p = t.indexOf(s.charAt(0));
t = t.substring(0, p) + t.substring(p+1);
System.out.println(t);
```

se va afișa:

a) Progrmare b) rogramare c) Programre d) Progrmre

4. Fie următorul program Java:

```
class A {
    public static int f(int x) { return x+1; }
    public int g(int x) { return x+2; }
}

class B extends A {
    public static int f(int x) { return x+4; }
    public int g(int x) { return x+3; }
}

public class Test {
    public static void main(String[] args) {
        A a = new B();
        System.out.println(a.f(1) + a.g(3));
    }
}
```

După executarea programului, se va afișa:

a) 7 b) 11 c) 8 d) 10

5. Fie următorul program Java:

```
class A {
    String sir = "";
    public A(String sir) { this.sir = this.sir + sir + "A"; }
}

class B extends A {
    public B(String sir) { super(sir); this.sir = this.sir + sir + "B"; }
}

class C extends B {
    public C(String sir) { super(sir); this.sir = this.sir + sir + "C"; }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new C("D").sir);
    }
}
```

După executarea programului, se va afişa:

a) DCDBDA b) DCBA c) DADBDC d) DADABDABC

II. Se consideră definită o clasă `Automobil` având datele membre `marca, model, capacitate` şi `pret`. Clasa încapsulează constructori, metode de tip set/get pentru toate datele membre, precum şi metodele `toString(), equals()` şi `hashCode()`. Creaţi o lista care să conţină cel puţin 3 obiecte de tip `Automobil` şi, folosind stream-uri bazate pe lista creată şi lambda expresii, rezolvaţi următoarele cerinţe:

- afişaţi automobilele care costă cel puţin 5000€, în ordinea descrescătoare a preţurilor;
- afişaţi mărcile distincte de automobile;
- creaţi o listă formată din automobilele care au capacitatea cilindrică cuprinsă între 2000 şi 3000 cm^3;
- afişaţi preţul maxim al unui automobil marca "Audi".

III. Scrieţi o clasă Java care să calculeze de câte ori apare un cuvânt dat într-un fişier text, folosind un fir de executare. Scrieţi un program care citeşte de la tastatură un cuvânt şi, utilizând clasa definită anterior, afişează numărul total al apariţiilor cuvântului respectiv în fişierele text *exemplu_1.txt*, *exemplu_2.txt* şi *exemplu_3.txt*. Cuvintele din fişierele text de intrare sunt despărţite între ele prin spaţii şi semnele de punctuaţie uzuale.

IV. Se consideră baza de date *Angajati*, având următorul URL: *jdbc:derby://localhost:1527/Angajati* . Baza de date conţine tabela *DateAngajati*, având câmpurile *CNP*, *Nume*, *Varsta* şi *Salariu*. Scrieţi un program care să citească de la tastatură o valoare reală *s* şi valoare întreagă *v*, după care afişează informaţiile despre angajaţii având cel mult *v* ani şi salariul cel puţin *s* RON.

NOTĂ:

- Datele de intrare se consideră corecte.
- Nu se vor trata excepţiile.
- Punctaj: 2.5p. (5 x 0.5p.) + 2.5p. + 2p. + 2p. + 1p. (din oficiu)