

**EXAMEN LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"**  
**- SESIUNEA MAI 2018 -**

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

1. Considerăm următorul program Java:

```
class Sir{
    private String sir;

    public Sir(String sir){
        this.sir = sir;
    }
    public void modificaSir(String sir){
        this.sir = sir;
    }
    public void modificaSir(Sir sir){
        sir = new Sir("Mihai");
    }
    public String getSir(){
        return sir;
    }
}

public class Test {
    public static void main(String[] args){
        Sir s = new Sir("Ion");
        Sir t = new Sir("Alex");
        s.modificaSir("Matei");
        t.modificaSir(new Sir("Dan"));
        s.modificaSir(t);
        System.out.println(s.getSir() + " " + t.getSir());
    }
}
```

După executarea programului, va fi afișată valoarea:

a) Matei Dan    b) Dan Dan    c) Matei Alex    d) Alex Dan

2. Considerăm următorul program Java:

```
class Calcul_1{
    void calcul(int a, int b){
        System.out.print(a + b + " ");
    }
}

class Calcul_2 extends Calcul_1{
    void calcul(int a, int b){
        System.out.print(a - b + " ");
    }
}
```

```
class Calcul_3 extends Calcul_2{
    void calcul(int a, int b){
        System.out.print(a * b + " ");
    }
}

public class Test{
    public static void main(String[] args){
        Calcul_1 x = new Calcul_3();
        x.calcul(1, 2);

        Calcul_2 y = (Calcul_2) x;
        y.calcul(3, 4);

        Calcul_3 z = (Calcul_3) y;
        z.calcul(5, 6);
    }
}
```

După executarea programului, va fi afișată valoarea:

- a) 2 12 30    b) 3 -1 30    c) 3 7 11    d) 2 7 11

3. După executarea secvenței de cod

```
int x = 0;
int[] a = {1, 2, 3, 4, 5};

try{
    a[1] = a[5] / (a[0]-1);
} catch(ArithmeticException e){
    System.out.print("AE ");
    x += 6;
} catch(IndexOutOfBoundsException e){
    System.out.print("IOBE ");
    x += 5;
} catch(RuntimeException e){
    System.out.print("RE ");
    x += 4;
} catch(Exception e){
    System.out.print("E ");
    x += 3;
} finally{
    System.out.print("F ");
    x += 2;
}
System.out.print(++x);
```

se va afișa:

- a) AE IOBE F 13    b) AE IOBE RE E F    c) AE F 9    d) IOBE F 8

4. Care dintre următoarele afirmații este adevărată pentru o metodă de tip `final`?

- a) poate fi și suprascrisă și supraîncărcată
- b) poate fi suprascrisă, dar nu poate fi supraîncărcată
- c) nu poate fi nici suprascrisă și nici supraîncărcată
- d) nu poate fi suprascrisă, dar poate fi supraîncărcată

5. Fie următorul program Java:

```
class Fir implements Runnable{
    int x;

    public Fir(int x){
        this.x = x;
    }

    public void run(){
        for (int i = 0; i < 10; i++) System.out.print(x);
    }

    public static void main(String args[]) throws InterruptedException{
        Fir obj1 = new Fir(1);
        Fir obj2 = new Fir(2);
        Thread t1 = new Thread(obj1);
        Thread t2 = new Thread(obj2);
        t1.start();
        t2.start();
        t2.join();
        System.out.print(3);
    }
}
```

După executarea programului, poate fi afișat un număr format din:

- a) 10 cifre egale cu 1, 10 cifre egale cu 2 și o cifră egală cu 3, cifrele fiind în orice ordine posibilă
- b) 10 cifre egale cu 1, 10 cifre egale cu 2 și o cifră egală cu 3, dar toate cifrele egale cu 2 se vor afla înaintea cifrei 3
- c) 10 cifre egale cu 1, 10 cifre egale cu 2 și o cifră egală cu 3, dar toate cifrele egale cu 1 sau 2 se vor afla înaintea cifrei 3
- d) 10 cifre egale cu 1, urmate de 10 cifre egale cu 2 și la sfârșit o cifră egală cu 3

II. Se consideră definită o clasă `Produs` având datele membre `firma`, `produs`, `cantitate` și `pret_unitar`. Clasa este utilizată pentru a memora informații despre produsele existente într-un magazin. Datele membre `firma` și `produs` sunt de tip `String`, iar `cantitate` și `pret_unitar` sunt de tip `float`. Clasa încapsulează metode de tip `set/get` pentru toate datele membre, precum și metodele `toString()`, `equals()` și `hashCode()`. Scrieți un program care să permită deserializarea unei liste cu obiecte de tip `Produs`, salvată în fișierul `produse.bin`. Folosind stream-uri bazate pe lista astfel creată și lambda expresii, rezolvați următoarele cerințe:

- afișați produsele firmei "MegaJava", în ordinea descrescătoare a prețurilor unitare;
- afișați o listă care conține denumirile distincte ale produselor, indiferent de firma producătoare;
- creați o colecție care să conțină produsele având valoarea totală cel puțin 1000 RON;
- afișați pentru fiecare firmă toate produsele sale aflate în magazin.



- III. Informațiile despre produsele existente într-un magazin sunt memorate într-un fișier text. Fiecare linie din fișier conține informații referitoare la un produs, sub forma `firma,produs,cantitate,pret_unitar`. Scrieți o clasă Java care să calculeze, pe baza informațiilor dintr-un fișier de tipul indicat anterior, valoarea totală a unui anumit produs, indiferent de firma producătoare, folosind un fir de executare dedicat. Scrieți un program care, utilizând clasa definită anterior, citește de la tastatură numele unui produs, după care afișează valoarea totală a produsului respectiv din două magazine, pe baza informațiilor din fișierele text `magazin_1.txt` și `magazin_2.txt`.
- IV. Se consideră baza de date `Produse`, având următorul URL: `jdbc:derby://localhost:1527/Produse`. Baza de date conține tabela `Stoc`, care păstrează informații despre produsele existente în mai multe magazine. Tabela `Stoc` are câmpurile `firma`, `produs`, `cantitate`, `pret_unitar` și `cod_magazin`. Scrieți un servlet/JSP care să preia prin 2 parametri (transmiși folosind metoda GET) numele unui firme și codul unui magazin, după care să generează o pagină HTML care să conțină informații despre produsele firmei respective existente în magazinul indicat sau un mesaj corespunzător în cazul în care magazinul nu are pe stoc nici un produs al firmei respective.

**NOTĂ:**

- Datele de intrare se consideră corecte.
- Nu se vor trata excepțiile.
- Punctaj: 2.5p. (5 x 0.5p.) + 2.5p. + 2p. + 2p. + 1p. (din oficiu)
- Metode ale clasei abstracte `HttpServlet`:
  - `protected void doGet(HttpServletRequest request, HttpServletResponse response)`
  - `protected void doPost(HttpServletRequest request, HttpServletResponse response)`
  - `public void init()`
  - `public void destroy()`