UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ - DEPARTAMENTUL DE INFORMATICĂ
DOMENIUL DE STUDII: INFORMATICĂ
SERIILE 23, 24 ȘI 25
VARIANTA 1

COLOCVIU LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"
– SESIUNEA MAI/IUNIE 2022 –

I. Pentru fiecare dintre cele 5 întrebări de mai jos indicați varianta de răspuns pe care o considerați corectă:

1. Fie următoarea clasă:

```
class A {
  int a;
  public A(int i) { a = i; }
  public int hashCode(){ return a; }
  public boolean equals(Object other) { return false; }
}
```

Să presupunem faptul că se va executa următorul cod:

```
HashMap<A,Integer> H = new HashMap<>();
H.put(new A(1), 1);
H.put(new A(1), 1);
H.put(new A(2), 2);
H.put(new A(2), 2);
```

Atunci tabela de dispersie `H` va conține valorile:

a) 1, 1, 2, 2 b) 1, 2 c) 1 d) 2

2. Fie următorul program Java:

```
class Persoana implements Serializable {
    String nume;
    int varsta;

    public Persoana(String nume, int varsta) {
        this.nume = nume;
        this.varsta = varsta;
        System.out.println("Constructor");
    }
}

public class Test {
    public static void main(String[] args) throws Exception {
        ObjectOutputStream oos = new ObjectOutputStream(
                                        new FileOutputStream("persoana.ser"));
        Persoana p = new Persoana("Popescu Ion", 40), q = p;
        oos.writeObject(q);
        oos.close();

        ObjectInputStream ois = new ObjectInputStream(
                                        new FileInputStream("persoana.ser"));
        Persoana r = (Persoana)ois.readObject();
        ois.close();
    }
}
```

De câte ori va fi afișat mesajul *Constructor* după executarea programului dat?

a) niciodată b) de trei ori c) de două ori d) o dată

3. Fie următorul program Java:

```
class Test {
    static String sir = "A";

    void A() {
        try {
            sir = sir + "B";
            B();
        } catch (Exception e) { sir = sir + "C"; }
    }

    void B() throws Exception {
        try {
            sir = sir + "D";
            C();
        }
        catch (Exception e) { throw new Exception(); }
        finally { sir = sir + "E"; }
    }

    void C() throws Exception { throw new Exception(); }

    public static void main(String[] args) {
        Test ob = new Test();
        ob.A();
        System.out.println(sir);
    }
}
```

După executarea programului, se va afișa:

a) ABCDE b) ABDE c) ABDEC d) ABCD

4. După executarea secvenței de cod

```
String s = "abracadabra";
s.toUpperCase();

int p = s.indexOf("B");
int q = s.lastIndexOf("B");
s = s.substring(0, s.length() + p - q);
System.out.println(s.length());
```

se va afișa:

a) 11 b) 4 c) 5 d) 10

5. Fie următorul cod Java:

```
class A {

    int intA;

    public A met1() { return new A(); }

    final void met2() { }
```

```
        public void met3() { }

        public static void met4() { }

        private int met5(int i) { return 5; }
    }

    class B extends A {

        int intB;

        public B met1() { return new B(); }

        public void met2() { }

        private void met3() { }

        static void met4() { }

        private int met5() { return 5; }
    }
```

Câte suprascrieri (overriding), supraîncărcări (overloading), respectiv erori conține codul de mai sus?

a) o suprascriere, două supraîncărcări, două erori
b) o suprascriere, o supraîncărcare, trei erori
c) două suprascrieri, o supraîncărcare, două erori
d) trei erori

II. Se consideră definită o clasă `Imobil` având datele membre `tip`, `localitate`, `nrCamere`, `suprafata` și `pret`. Clasa este utilizată pentru a memora informații despre imobilele gestionate de o agenție imobiliară. Datele membre `tip` și `localitate` sunt de tip `String`, data membră `nrCamere` este de tip `int`, iar datele membre `suprafata` și `pret` sunt de tip `double`. Clasa încapsulează constructor cu argumente, metode de tip `set/get` pentru toate datele membre, precum și metodele `toString()`, `equals()` și `hashCode()`. Creați o lista care să conțină cel puțin 3 obiecte de tip `Imobil` și, folosind stream-uri bazate pe lista creată și lambda expresii, rezolvați următoarele cerințe:

- afișați imobilele care au cel puțin 3 camere și nu costă mai mult de 100000 RON, în ordinea crescătoare a suprafețelor lor;
- afișați localitățile distincte;
- creați o colecție care să conțină imobilele aflate în București cu prețul cuprins între 200000 RON și 500000 RON;
- afișați pentru fiecare localitate distinctă lista imobilelor.

III. Informațiile despre imobilele gestionate de către lanțul agențiilor imobiliare `RentSale` sunt păstrate în mai multe fișiere text. Fiecare linie dintr-un astfel de fișier conține informații referitoare la un imobil, respectiv `tip`, `localitate`, `nrCamere`, `suprafata` și `pret`, despărțite prin virgule. Scrieți o clasă Java care să calculeze, pe baza informațiilor dintr-un fișier de tipul indicat anterior, numărul imobilelor care au o suprafață mai mare decât o valoare dată, precum și un număr minim de camere, folosind un fir de executare dedicat. Scrieți un program care, utilizând clasa definită anterior, citește de la tastatură o valoare reală `smin` și un număr natural `cmin`, după care afișează numărul total al imobilelor având suprafețele mai mari decât `smin` și cel puțin `cmin` camere existente în două agenții, pe baza informațiilor din fișierele text `agentieRentSale_1.txt` și `agentieRentSale_2.txt`.

IV. Se consideră definită complet clasa `Adresa` care permite memorarea unei adrese. Definiți complet o clasă imutabilă `Firma` care să permită memorarea următoarelor informații despre o firmă: denumirea firmei (șir de caractere), numărul de angajați (număr natural), profitul mediu anual (număr real) și adresa (referință spre un obiect de tip `Adresa`).

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)