

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

1. 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 true; }  
}
```

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:

```
public class Test {  
    public static void main(String[] args) throws FileNotFoundException {  
        for (int i = 1; i <= 5; i++)  
            try (Scanner sc = new Scanner(new File("file" + i + ".txt"))) {  
                System.out.print(sc.nextInt());  
            } catch (FileNotFoundException e) {  
                System.out.print("!");  
            } catch (Exception e) {  
                System.out.print("?");  
            } finally {  
                System.out.print("F");  
            }  
    }  
}
```

Presupunem faptul că în directorul unde se execută programul există doar următoarele fișiere text:

- file1.txt gol
- file2.txt conținând doar numărul 2
- file3.txt gol
- file4.txt conținând doar numărul 4

După executarea programului, se va afișa:

- a) ?F2F?F4F!F, executare terminată cu excepție
- b) ?F2?F4!F, executare terminată cu succes
- c) ?F2F?F4F!F, executare terminată cu succes
- d) ?F2F?F4F!, executare terminată cu excepție

3. Fie următorul program Java:

```
class Persoana {
    String nume;
    int varsta;
    public Persoana(String nume, int varsta) {
        this.nume = nume;
        this.varsta = varsta;
    }
    public String getNume() { return nume; }
    public int getVarsta() { return varsta; }
}

public class Test {
    public static void main(String[] args) throws Exception {
        ArrayList<Persoana> listaPersoane = new ArrayList<>();
        listaPersoane.add(new Persoana("Dan", 27));
        listaPersoane.add(new Persoana("Sorina", 26));
        listaPersoane.add(new Persoana("Ana", 23));
        listaPersoane.add(new Persoana("Bogdan", 22));
        listaPersoane.add(new Persoana("Camelia", 25));
        listaPersoane.add(new Persoana("Robert", 22));

        System.out.println(listaPersoane.stream()
            .filter(p -> p.getNume().endsWith("a"))
            .filter(p -> p.getVarsta() > 24)
            .map(Persoana::getNume)
            .sorted()
            .map(o -> o.substring(0, 1))
            .collect(Collectors.joining(", ")));
    }
}
```

Ce se va afișa după rularea programului?

- a) A,C,S      b) C,A,S      c) S,C      d) C,S

4. După executarea secvenței de cod

```
String s="";
for (int i=0;i<=5;i++)
    (s += i; s += 5-i;)

int p = s.lastIndexOf("2");
s = s.substring(p);
System.out.println(s.length());
```

se va afișa:

- a) 4      b) 5      c) 6      d) 7

5. Fie următorul cod Java:

```
class Fir extends Thread{
    int nivel;
    static int numar = 0;

    public Fir(int n){nivel = n;}
}
```

```

        public void run() {
            System.out.print(nivel + " ");

            if (nivel < 2) {
                Fir fir = new Fir(nivel+1);
                fir.start();
            }

            if (nivel < 10000000) {
                Fir fir = new Fir(nivel+1);
                fir.start();
            }
        }
    }

    public class Test {
        public static void main(String[] args) {
            Fir fir = new Fir(0);
            fir.start();
        }
    }

```

Presupunând faptul că s-ar termina executarea programului, de câte ori va fi afișată valoarea 9999999 ?

- a) de 4 ori
- b) de 6 ori
- c) de 8 ori
- d) de 16 ori

II. Se consideră definită o clasă Suvenir având datele membre denumire, material, cantitate și pret\_unitar. Clasa este utilizată pentru a memora informații despre suvenirurile existente într-un bazar. Datele membre denumire și material sunt de tip String, data membră cantitate este de tip int, iar data membră pret este 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 listă care să conțină cel puțin 3 obiecte de tip Suvenir și, folosind stream-uri bazate pe lista creată și lambda expresii, rezolvați următoarele cerințe:

- afișați suvenirurile confecționate din aur și care au prețul unitar cel mult 100 RON, în ordinea crescătoare a prețurilor lor unitare;
- afișați materiale distincte din care sunt confecționate suvenirurile;
- creați o colecție care să conțină suvenirurile confecționate din plastic având valoarea totală cuprinsă între 500 RON și 1000 RON;
- afișați pentru fiecare material lista suvenirurilor confecționate din el.

III. Informațiile despre suvenirurile existente în bazarul HaiLaNoi din București sunt păstrate în mai multe fișiere text, câte unul pentru fiecare tarabă. Fiecare linie dintr-un astfel de fișier conține informații referitoare la un suvenir, respectiv denumire, material, cantitate și pret\_unitar, despărțite prin virgule. Scrieți o clasă Java care să calculeze, pe baza informațiilor dintr-un fișier de tipul indicat anterior, valoarea totală a suvenirurilor confecționate dintr-un anumit material care sunt vândute la taraba respectivă, folosind un fir de executare dedicat. Scrieți un program care, utilizând clasa definită anterior, citește de la tastatură denumirea unui material sub forma unui șir de caractere, după care afișează valoarea totală a suvenirurilor confecționate din materialul respectiv care sunt vândute la două tarabe, pe baza informațiilor din fișierele text stoisko\_1.txt și stoisko\_2.txt.

- IV. Se consideră definită complet o clasă serializabilă *Persoana* care permite memorarea următoarelor informații despre o persoană: nume (șir de caractere), vârsta în ani (număr natural) și salariul mediu anual (număr real). Definiți complet o clasă singleton denumită *CitireScrierePersoane* care să permită citirea sau scrierea informațiilor despre mai multe persoane într-un fișier text de tip CSV, respectiv informațiile despre o persoană se vor scrie pe o linie, despărțite între ele prin câte o virgulă. Informațiile despre persoane se vor citi din fișierul text într-un obiect `ArrayList<Persoana>`, respectiv se vor scrie

în fișierul text dintr-un obiect de tip `ArrayList<Persoana>`. Numele fișierului text de tip CSV și referința spre obiectul de tip `ArrayList<Persoana>` se vor transmite ca parametrii ai constructorului clasei *CitireScrierePersoane*.

**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)