

## COLOCVIU LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE" – SESIUNEA MAI/IUNIE 2023 –

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

1. Fie următorul program:

```
public class Test {  
    public static void main(String[] args) {  
        LinkedHashMap m = new LinkedHashMap();  
        m.put("a", null);  
        m.put("b", "JavaSE");  
        m.put("c", "JavaSE");  
        m.put("c", "Python");  
        m.put(null, "PHP");  
        m.put(null, null);  
        System.out.println(m);    }  
}
```

După executarea sa, se va afișa:

- a) {a=null, b=JavaSE, c=Python, null=null}
- b) {a=null, b=JavaSE, c=Python}
- c) {null=PHP, a=null, b=JavaSE, c=Python}
- d) Excepție de tip NullPointerException la executare

2. Fie următorul program Java:

```
class A {  
    int x = 10;  
    static int y = 20;  
}  
  
class B extends A {  
    int x = 30;  
    static int y = 40;  
}  
  
public class Test {  
    public static void main(String[] args) {  
        A ob = new B();  
        System.out.println(ob.x + " " + ob.y);  
    }  
}
```

După executarea programului, se va afișa:

- a) 10 40      b) 30 40      c) 20 30      d) 10 20

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) ABEDC   b) ABCED   c) ABDEC   d) ABCD

4. Fie următorul program Java:

```
public class Test {
    public static void main(String[] args) {
        List<Integer> numere = new ArrayList<Integer>();

        for (int i = 0; i < 11; i++)
            numere.add(i);

        Iterator<Integer> itr = numere.iterator();
        while (itr.hasNext()) {
            Integer nr = itr.next();
            if (nr % 2 == 0)
                numere.remove(nr);
        }
        System.out.println(numere);
    }
}
```

Care dintre următoarele afirmații este adevărată?

- a) Programul va afișa doar numere impare.
- b) Unei colecții de tip `List` nu i se poate atașa un iterator.
- c) Apelul metodei `remove` va genera o excepție de tip `ConcurrentModificationException` la executare.
- d) Apelul metodei `remove` va genera o excepție de tip `ConcurrentModificationException` la compilare.

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 `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()`. Creați o listă care să conțină cel puțin 3 obiecte de tip `Produs` și, folosind stream-uri bazate pe lista creată și lambda expresii, rezolvați următoarele cerințe:

- afișați produsele al căror preț unitar este cel puțin 100 RON, în ordinea descrescătoare a cantităților;
- afișați firmele producătoare distincte, în ordine alfabetică;
- creați o colecție care să conțină produsele având valoarea totală cel mult 1000 RON;
- afișați prețul unitar mediu al tuturor produselor.

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 produselor unei anumite firme, folosind un fir de executare dedicat. Scrieți un program care, utilizând clasa definită anterior, citește de la tastatură numele unei firme producătoare, după care afișează valoarea totală a produselor sale existente în două magazine, pe baza informațiilor din fișierele text `magazin_1.txt` și `magazin_2.txt`. Datele de intrare firma și produs sunt de tip `String`, iar cantitate și pret\_unitar sunt de tip `float`.

- IV. Se consideră definită complet clasa mutabilă `Adresa` care permite memorarea unei adrese din România: *stradă, număr, bloc, scară, etaj, apartament, județ și localitate*. Definiți complet o clasă imutabilă `Facultate` care să permită memorarea următoarelor informații despre o facultate: *denumirea facultății* (șir de caractere), *numărul de studenți* (număr natural), *specializări* (listă cu elemente de tip `String`) 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. + 1p. (din oficiu)