

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

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

1. Fie următorul program Java:

```
class C { public static int a=1; }  
public class Test {  
    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 afișată valoarea:

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

2. Fie următorul program Java:

```
class A {  
    public int f(int x) { return x+1; }  
    public static int g(int x) { return x+2; }  
}  
  
class B extends A {  
    public int f(int x) { return x+4; }  
    public static 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) 8 c) 10 d) 11

3. 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 afișa dacă metoda `met()` va lansa excepția `InputMismatchException`?

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

4. După executarea secvenței de cod

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

se va afișa:

- a) Progrmare
- b) Programre
- c) Progmre
- d) rogramare

5. 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 suprascrisă, dar poate fi supraîncărcată
- d) nu poate fi nici suprascrisă și nici supraîncărcată

II. Se consideră definită complet o clasă `Automobil` având datele membre `marca`, `model`, `capacitate` și `pret`. Clasa încapsulează constructori, 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 `Automobil` și, folosind stream-uri bazate pe lista creată și lambda expresii, rezolvați următoarele cerințe:

- afișați automobilele care costă cel mult 7000€, în ordinea crescătoare a prețurilor;
- afișați mărcile distincte de automobile în ordine alfabetică;
- creați o listă formată din automobilele care au capacitatea cilindrică cuprinsă între 1000 și 2000 cm³;
- afișați prețul minim al unui automobil marca "BMW".

III. Scrieți o clasă Java care să calculeze câte cuvinte sunt într-un fișier text, folosind un fir de executare. Scrieți un program care citește de la tastatură un cuvânt și, utilizând clasa definită anterior, afișează numărul total de cuvinte din fișierele text *exemplu_1.txt* și *exemplu_2.txt*. Cuvintele din fișiere sunt despărțite între ele prin spații și semnele de punctuație uzuale.

IV. Se consideră definită complet o clasă *Angajat* având datele membre *nume*, *vechime* și *salariu*. Data membră *nume* este de tip *String*, *vechime* este de tip *int*, iar *salariu* este de tip *double*. Clasa încapsulează constructori, metode de tip *set/get* pentru toate datele membre, precum și metodele *toString()*, *equals()* și *hashCode()*. Cerințe:

- știind că impozitul unui angajat se calculează doar în funcție de *salariu* și *vechime*, definiți o interfață funcțională corespunzătoare unei metode pentru calculul impozitului unui angajat;
- definiți o clasă care să conțină o metodă statică pentru calculul impozitului maxim al angajaților memorați într-o listă cu elemente de tip *Angajat* folosind o formulă de calcul generică, transmisă utilizând mecanismul de callback;
- folosind o lambda expresie, exemplificați utilizarea clasei utilitare pentru o listă cu elemente de tip *Angajat* și următoarea formulă de calcul a impozitului: $\text{impozit} = \text{salariu}/100 + \text{vechime} \cdot 10$.

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)