

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

REEXAMINĂRI/MĂRIRI LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"
- SESIUNEA SEPTEMBRIE 2022 -

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

1. Fie următorul program:

```
class C {  
    static int x = 1;  
    static int f() { return x++; }  
    static int g() { return --x; }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        System.out.println(C.f() + " " + C.g() + " " + C.x);  
    }  
}
```

După executarea programului de mai sus, pe ecran se vor afișa valorile:

- a) 2 2 2 b) 2 2 1 c) 1 1 2 **(d) 1 1 1**

2. 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) după executarea programului se vor afișa doar numerele impare cuprinse între 1 și 10
b) se generează o excepție la compilare deoarece unei colecții de tip List nu i se poate atașa un iterator
(c) se generează o excepție la executare de tip ConcurrentModificationException în urma apelului metodei remove
d) se generează o excepție la compilare de tip ConcurrentModificationException din cauza apelului metodei remove

3. Fie următorul program Java:

```
class Super {  
    public Super() { print(); }  
    public void print() { System.out.println("Super "); }  
}
```

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

```
class Sub extends Super {
    public Sub() { print(); }
    public void print() { System.out.println("Sub"); }
}

public class Test {
    public static void main(String args[]) {
        Super p = new Sub();
        p.print();
    }
}
```

După executarea programului se va afișa:

- | | | | |
|--------|--------|----------|----------|
| a) Sub | b) Sub | c) Super | d) Super |
| Sub | Sub | Super | Sub |
| Sub | Super | Super | Super |

4. Fie următorul program Java:

```
class A{
    int a;
    public A(int i){a=i;}
    public String toString(){return "A"+(a%3);}
    public int get(){return a/3;}
}

interface Interface1{void f(int x);}
interface Interface2{int f();}
interface Interface3{String f(String s);}
interface Interface4{void f(String s);}

public class Test {
    public static void main(String[] args) {
        int x = ...;

        A obiect = new A(x);
        Interface1 ob1 = i -> System.out.print(i * 3 + " ");
        Interface2 ob2 = () -> obiect.get();
        Interface3 ob3 = s -> s.substring(1);
        Interface4 ob4 = s -> System.out.println(s);

        ob1.f(ob2.f());
        ob4.f(ob3.f(obiect.toString()));
    }
}
```

Dacă programul va afișa 42 4 după executarea sa, precizați care dintre următoarele afirmații este adevărată:

- a) valoarea inițială a variabilei x este multiplu de 3
☒ b) valoarea inițială a variabilei x este multiplu de 4
☐ c) valoarea inițială a variabilei x este multiplu de 5
☐ d) valoarea inițială a variabilei x este multiplu de 7

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

5. Fie următoarea clasă:

```
class A {
    int a;
    public A(int i) { a = i; }
    public int hashCode(){ return a; }
    public boolean equals(Object b) { return a != ((A)b).a; }
}
```

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, 1, 2 d) 1, 2, 2

II. Se consideră definită complet o clasă *ProdusAlimentar* având datele membre *denumire*, *cantitate*, *pretUnitar* și *dataExpirare*. Clasa este utilizată pentru a memora informații despre produsele firmei *MegaSmall*. Datele membre *denumire* și *data* sunt de tip *String* (*data* este un șir de forma "zz-ll-aa"), *cantitate* este de tip *int*, iar *pretUnitar* este de tip *double*. 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 *ProdusAlimentar* și, folosind stream-uri bazate pe lista creată și lambda expresii, rezolvați următoarele cerințe:

- afișați lista produselor a căror *denumire* începe cu litera A, în ordinea crescătoare a prețurilor unitare;
- creați o colecție care să conțină produsele al căror preț unitar este cuprins între 100 RON și 1000 RON;
- considerând faptul că firma *MegaSmall* își va vinde toate produsele în timp util (până la expirarea lor), calculați încasările totale ale firmei;
- afișați, pentru fiecare luna, lista produselor care vor expira în luna respectivă.

III. Informațiile despre produsele firmei *MegaSmall* sunt păstrate în mai multe fișiere text. Fiecare linie dintr-un astfel de fișier conține informații referitoare la un produs alimentară, respectiv *denumire*, *cantitate*, *pretUnitar* și *dataExpirare*, despărțite prin virgule. Scrieți o clasă Java care să calculeze, pe baza informațiilor dintr-un fișier de tipul indicat anterior, valoarea totală a produselor care vor expira într-o anumită lună și au un preț unitar minim, folosind un fir de executare dedicat. Scrieți un program care, utilizând clasa definită anterior, citește de la tastatură o valoare reală *pmín* și un șir de caractere *lexp*, după care afișează valoarea totală a produselor având prețurile unitare cel puțin egale cu *pmín* și care vor expira în luna *lexp*, pe baza informațiilor din fișierele text *produse_1.txt* și *produse_2.txt*.

IV. Se consideră definită complet clasa *Adresa* care permite memorarea unei adrese. Definiți complet o clasă imutabilă *Angajat* care să permită memorarea următoarelor informații despre un angajat: numele (șir de caractere), vârsta (număr natural), venitul 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. (0.5p. + 1p. + 0.5p.) + 1p. (din oficiu)