

REEXAMINARE LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"

- SESIUNEA SEPTEMBRIE 2017 -

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 int a;
    public static int b;

    public C() {
        b++; b = b + a;
    }
    public void met() {
        a = a + b; a++;
    }
}

public class Test {
    public static void main(String[] args) {
        C ob1 = new C();    b = 1, a = 1
        ob1.met();           a = 2
        C ob2 = new C();    b = 2
        ob2.met();           a = 3
        System.out.println(ob1.a + ob2.b);
    }
}
```

După executarea programului, va fi afișată valoarea:

- a) 11 b) 9 **(c) 4** d) 3

2. După executarea secvenței de cod

```
String s = "academician";
int p = s.lastIndexOf("c");
int q = s.indexOf(s.charAt(p));
s = s.substring(q, p);
System.out.println(s);
```

se va afișa:

- a) ademi **(b) cademic** c) ademic d) cademi

3. Fie următorul program Java:

```
class C {
    public static void met_1() throws Exception {
        try {
            throw new Exception();
        }
        finally {
            System.out.print("A");
        }
    }
    public static void met_2() throws Exception {
        try {
            met_1();
        }
        catch(Exception ex) {
            System.out.print("B");
        }
    }
}
```

ABDE

```
public class Test {  
    public static void main(String args[]) {  
        try {  
            C.met_2();  
        }  
        catch (Exception e) {  
            System.out.print("C");  
        }  
        finally {  
            System.out.print("D");  
        }  
        System.out.println("E");  
    }  
}
```

ABDE

După executarea programului, se va afișa:

a) ABCDE

☒ b) ABDE

☒ c) CBADE

☒ d) CADE

4. Fie următorul program Java:

```
class B {  
    public B() {  
        System.out.print("A");  
    }  
    public void met() {  
        System.out.print("B");  
    }  
}  
  
class C extends B {  
    public C() {  
        System.out.print("C");  
    }  
    public void met() {  
        System.out.print("D");  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        B ob = new C();  
        ob.met();  
    }  
}
```

Cn

După executarea programului, se va afișa:

☒ a) ACBD

☒ b) CB

c) CADB

☒ d) ACD

5. Considerăm următoarele afirmații:

- I. o clasă abstractă poate fi instanțiată
- II. o clasă abstractă trebuie să conțină cel puțin o metodă abstractă
- III. o clasă abstractă nu poate fi declarată ca fiind de tip final
- IV. o clasă abstractă nu poate să aibă constructori

Precizați care dintre afirmațiile de mai sus sunt false:

a) I, II, IV

☒ b) II, III, IV

☒ c) I, II, III

d) I, II, IV

- II. Se consideră definită o clasă *Vacanță* având datele membre *destinație*, *nr_persoane*, *valoare* și *perioada*. Clasa este utilizată pentru a memora informații despre vacanțele pe care le oferă o agenție de turism. Data membră *destinație* este de tip *String*, data membră *nr_persoane* este de tip *int*, data membră *valoare* este de tip *double*, iar data membră *perioada* este un șir de caractere (*String*) de forma "zz/11/aaaa-zz/11/aaaa". 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 *Vacanță* și, folosind *stream-uri* bazate pe lista creată și *lambda expresii*, rezolvați următoarele cerințe:
- afișați vacanțele pentru două persoane, în ordinea descrescătoare a valorilor lor;
 - afișați destinațiile distincte ale vacanțelor;
 - creați o colecție care să conțină vacanțele din luna septembrie a anului curent având destinația Dubai;
 - afișați pentru fiecare destinație o listă a vacanțelor oferite de agenție.
- III. Informațiile despre vacanțele oferite de sucursalele agenției de voiaj *WymarzoneWakacje* sunt păstrate în mai multe fișiere text. Fiecare linie dintr-un astfel de fișier conține informații referitoare la o vacanță, respectiv *destinație*, *nr_persoane*, *valoare* și *perioada*, despărțite prin virgule. Scrieți o clasă *Java* care să calculeze, pe baza informațiilor dintr-un fișier de tipul indicat anterior, valoarea totală a vacanțelor cu o anumită destinație, folosind un fir de executare dedicat. Scrieți un program care, utilizând clasa definită anterior, citește de la tastatură un șir de caractere reprezentând o localitate, după care afișează valoarea totală a vacanțelor având ca destinație localitatea respectivă oferite de două sucursale ale agenției de voiaj, pe baza informațiilor din fișierele text *Bucharest.txt* și *Warsaw.txt*.
- IV. Se consideră baza de date *WymarzoneWakacje*, având următorul URL: *jdbc:derby://localhost:1527/WW*. Baza de date conține tabela *Wakacje*, care păstrează informații despre vacanțele oferite de agenția de voiaj *WymarzoneWakacje*. Tabela *Wakacje* are câmpurile *destinație*, *nr_persoane*, *valoare* și *perioada*. Scrieți un *servlet* care să preia printr-un parametru (transmis folosind metoda *GET*) un șir de caractere reprezentând numele unui oraș, iar apoi să genereze o pagină *HTML* care să conțină informații despre vacanțele oferite de agenție având ca destinație orașul respectiv sau un mesaj corespunzător în cazul în care agenția nu oferă nicio vacanță având ca destinație orașul respectiv.

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)
- Metode ale clasei abstracte *HttpServlet*:
 - `protected void doGet(HttpServletRequest request, HttpServletResponse response)`
 - `protected void doPost(HttpServletRequest request, HttpServletResponse response)`
 - `public void init()`
 - `public void destroy()`