

EXAMEN LA DISCIPLINA "PROGRAMARE AVANSATA PE OBIECTE"
- SESIUNEA MAI/IUNIE 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 static int a=1; }
public class teste_grila {
    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. 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
IllegalArgumentException?

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

3. După executarea sevenței de cod

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

se va afișa:

- a) Progrmare b) rogramare c) Programme d) Progrmre

4. Fie următorul program Java:

```
class A {
    public static int f(int x) { return x+1; }
    public int g(int x) { return x+2; }
}

class B extends A {
    public static int f(int x) { return x+4; }
    public 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));
    }
}
```

statica după tipul declarat
nstatica după tipul real

După executarea programului, se va afișa:

- a) 7 b) 11 c) 8 d) 10

5. Fie următorul program Java:

```
class A {
    String sir = "";
    public A(String sir) { this.sir = this.sir + sir + "A"; }
}

class B extends A {
    public B(String sir) { super(sir); this.sir = this.sir + sir + "B"; }
}

class C extends B {
    public C(String sir) { super(sir); this.sir = this.sir + sir + "C"; }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new C("D").sir);
    }
}
```

Precizați care dintre următoarele afirmații este adevărată pentru programul dat:

- a) există erori de compilare în clasa Test b) există erori de compilare în clasa B
c) va afișa BPBQR după rulare d) va afișa APBQR după rulare

4. Fie următorul program Java:

```
class Persoana implements Serializable {
    String nume;
    int varsta;

    public Persoana(String nume, int varsta) {
        this.nume = nume;
        this.varsta = varsta;
        System.out.println("Constructor");
    }
}

public class Test {
    public static void main(String[] args) throws Exception {
        ObjectOutputStream oos = new ObjectOutputStream(
            new FileOutputStream("persoana.ser"));
        Persoana p = new Persoana("Popescu Ion", 40), q = p;
        oos.writeObject(q);
        oos.close();
        ObjectInputStream ois = new ObjectInputStream(
            new FileInputStream("persoana.ser"));
        Persoana r = (Persoana)ois.readObject();
        ois.close();
    }
}
```

De câte ori va fi afișat mesajul *Constructor*, după executarea programului dat?

- a) niciodată b) o dată c) de două ori d) de trei ori

5. 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"; }
    }
}
```

VERIFICARE LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"
- SESIUNEA MAI/IUNIE 2019 -

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

1. Fie următorul program Java:

```
class Automobil{
    private String marca;
    public Automobil(String marca) { this.marca = marca; }
    public int hashCode() { return 0; }
    public boolean equals(Object obj) { return true; }
}

public class Test {
    public static void main(String[] args) {
        HashSet<Automobil> la = new HashSet<>();
        la.add(new Automobil("Audi"));
        la.add(new Automobil("BMW"));
        la.add(new Automobil("Audi"));
        la.add(new Automobil("Opel"));
        System.out.println(la.size());
    }
}
```

După executarea programului, va fi afișată valoarea:

- a) 4 b) 3 c) 2 d) 1

2. Precizați care dintre următoarele afirmații este adevărată pentru clasele imutabile:

- a) sunt implicit și clase de tip singleton
- b) au toate datele membre de tip final
- c) au un singur constructor, fără parametri
- d) referințele spre instanțele lor nu pot fi modificate

3. Fie următorul program Java:

```
class A {
    public static void metoda(String s) {System.out.print("A"+s); }

class B extends A {
    public static void metoda(String s) {System.out.print("B"+s); }
    public void metoda(String s, String t) { System.out.print("B"+s+t); }

public class Test{
    public static void main(String[] args) {
        A ob = new B();
        ob.metoda("P");
        ob.metoda("Q", "R");
    }
}
```

4. 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 nici suprascrisă și nici supraîncărcată
 - d) nu poate fi suprascrisă, dar poate fi supraîncărcată
5. Fie următorul program Java:

```
class Fir implements Runnable{
    int x;

    public Fir(int x){
        this.x = x;
    }

    public void run(){
        for (int i = 0; i < 10; i++) System.out.print(x);
    }

    public static void main(String args[]) throws InterruptedException{
        Fir obj1 = new Fir(1);
        Fir obj2 = new Fir(2);
        Thread t1 = new Thread(obj1);
        Thread t2 = new Thread(obj2);
        t1.start();
        t2.start();
        t2.join();
        System.out.print(3);
    }
}
```

După executarea programului, poate fi afișat un număr format din:

- a) 10 cifre egale cu 1, 10 cifre egale cu 2 și o cifră egală cu 3, cifrele fiind în orice ordine posibilă
 - b) 10 cifre egale cu 1, 10 cifre egale cu 2 și o cifră egală cu 3, dar toate cifrele egale cu 2 se vor afla înaintea cifrei 3
 - c) 10 cifre egale cu 1, 10 cifre egale cu 2 și o cifră egală cu 3, dar toate cifrele egale cu 1 sau 2 se vor afla înaintea cifrei 3
 - d) 10 cifre egale cu 1, urmate de 10 cifre egale cu 2 și la sfârșit o cifră egală cu 3
- 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()*. Scrieți un program care să permită deserializarea unei liste cu obiecte de tip *Produs*, salvată în fișierul *produse.bin*. Folosind stream-uri bazate pe lista astfel creată și lambda expresii, rezolvării următoarele cerințe:
- afișați produsele firmei "Megajava", în ordinea descrescătoare a prețurilor unitare;
 - afișați o listă care conține denumirile distincte ale produselor, indiferent de firma producătoare;
 - creați o colecție care să conțină produsele având valoarea totală cel puțin 1000 RON;
 - afișați pentru fiecare firmă toate produsele sale aflate în magazin.

```
class Calcul_3 extends Calcul_2{
    void calcul(int a, int b){
        System.out.print(a * b + " ");
    }
}

public class Test{
    public static void main(String[] args){
        Calcul_1 x = new Calcul_3();
        x.calcul(1, 2);

        Calcul_2 y = (Calcul_2) x;
        y.calcul(3, 4);

        Calcul_3 z = (Calcul_3) y;
        z.calcul(5, 6);
    }
}
```

După executarea programului, va fi afișată valoarea:

- a) 2 12 30 b) 3 -1 30 c) 3 7 11 d) 2 7 11

3. După executarea secvenței de cod

```
int x = 0;
int[] a = {1, 2, 3, 4, 5};

try{
    a[1] = a[5] / (a[0]-1);
} catch(ArithmcticException e){
    System.out.print("AE ");
    x += 6;
} catch(IndexOutOfBoundsException e){
    System.out.print("IOBE ");
    x += 5;
} catch(RuntimeException e){
    System.out.print("RE ");
    x += 4;
} catch(Exception e){
    System.out.print("E ");
    x += 3;
} finally{
    System.out.print("F ");
    x += 2;
}
System.out.print(++x);
```

se va afișa:

- a) AE IOBE F 13 b) AE IOBE RE E F c) AE F 9 d) IOBE F 8

EXAMEN LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"
- SESIUNEA MAI 2018 -

- I. Pentru fiecare dintre cele 5 întrebări de mai jos, indicați variantele de răspuns pe care le considerați corecte:
1. Considerăm următorul program Java:

```
class Sir{
    private String sir;

    public Sir(String sir){
        this.sir = sir;
    }
    public void modificaSir(String sir){
        this.sir = sir;
    }
    public void modificaSir(Sir sir){
        sir = new Sir("Mihai");
    }
    public String getSir(){
        return sir;
    }
}

public class Test {
    public static void main(String[] args){
        Sir s = new Sir("Ion");
        Sir t = new Sir("Alex");
        s.modificaSir("Matei");
        t.modificaSir(new Sir("Dan"));
        s.modificaSir(t);
        System.out.println(s.getSir() + " " + t.getSir());
    }
}
```

După executarea programului, va fi afișată valoarea:

- a) Matei Dan b) Dan Dan c) Matei Alex d) Alex Dan

2. Considerăm următorul program Java:

```
class Calcul_1{
    void calcul(int a, int b){
        System.out.print(a + b + " ");
    }
}

class Calcul_2 extends Calcul_1{
    void calcul(int a, int b){
        System.out.print(a - b + " ");
    }
}
```

```
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");  
    }  
}
```

A~~B~~CDE

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();  
    }  
}
```

C~~B~~

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

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 obl = new C();      b=1
        obl.met();           a=2
        C ob2 = new C();     b=2
        ob2.met();           a=3
        System.out.println(obl.a + ob2.b);
    }
}
```

După executarea programului, va fi afișată valoarea:

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

2. După executarea sevenției 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");
        }
    }
}
```

ABCDE

După executarea programului, se va afișa:

a) MCFT

b) CMFT

c) CFMT

d) MCT

3. După executarea secvenței de cod

```
String s = "abracadabra";
s.replace('r', 'R');
int p = s.indexOf("R");
int q = s.lastIndexOf("b");
s = s.substring(0, q - p);
System.out.println(s.length());
```

se va afișa:

a) 10

b) 6

c) 9

d) 7

4. Fie următorul program Java:

```
class B {
    static String x = "";
    public static void met_1() {
        x = x + "A";
    }
    public void met_2() {
        x = x + "B";
    }
}

class C extends B {
    public static void met_1() {
        x = x + "C";
    }
    public void met_2() {
        x = x + "D";
    }
}

public class Test {
    public static void main(String[] args) {
        B ob = new C();
        ob.met_1();
        ob.met_2();
        System.out.println(ob.x);
    }
}
```

După executarea programului, se va afișa:

a) AB

b) CD

c) CB

d) AD

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 poate să aibă constructori

Precizați care dintre afirmațiile de mai sus sunt false:

a) II, III, IV

b) III, IV

c) II, IV

d) I, II, III

RESTANȚĂ 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 static int a = 1;
    public int b = 1;

    public void met() {
        a++;      l l z z z z
        b++;      l l
    }
}

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

După executarea programului, va fi afișată valoarea:

- (a) 4 ~~(b)~~ 3 c) 6 d) 5

2. Fie următorul program Java:

```
class C {
    public static void met() throws Exception {
        try {
            throw new Exception();
        }
        finally {
            System.out.print("M");
        }
    }
}

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

M C FT

3. După executarea se poate afișa de cod

```
String s = new String("Examen");
if (s == "Examen") System.out.print("A");
else System.out.print("B");
if (s.equals("Examen")) System.out.print("C");
else System.out.print("D");
```

se va afișa:

- a) AD b) BC c) AC d) BD

4. Fie următorul program Java:

```
class A{
    public int x = 1;
    void afisare() { System.out.println(x); }
}
class B extends A{
    public int x = 2;
    void afisare() { System.out.println(x); }
}
public class Test{
    public static void main(String[] args){
        A ob = new B();
        System.out.println(++ob.x);
    }
}
```

După executarea programului, se va afișa:

- a) 0 b) 2 c) 1 d) 3

5. Fie următorul program Java:

```
class Tablou{
    static void met(int[] a, int b){
        int aux;
        aux = a[0]; a[0] = b; b = aux;
    }
}
public class Test{
    public static void main(String[] args){
        int a[] = {1, 2, 3, 4, 5}, b = 6;
        Tablou.met(a, b);
        int s = b;
        for (int i = 0; i < a.length; i++) s = s + a[i];
        System.out.println(s);
    }
}
```

După executarea programului, se va afișa:

- a) 15 b) 21 c) 21 d) 26

EXAMEN LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"
- SESIUNEA MAI/IUNIE 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 ABC{
    public static int x;
    public int y;

    public ABC() { met(); }

    public void met(){ x = x + 3; y = y + 1; }

    public class Test{
        public static void main(String[] args){
            ABC t = new ABC();
            ABC u = new ABC();
            System.out.println(t.x + u.y);
        }
    }
}
```

După executarea programului, va fi afișată valoarea:

- a) 4 b) 5 c) 7 d) 8

2. Fie următorul program Java:

```
class Test{
    String str = "a";

    void A(){
        try {
            str += "b"; B();
        } catch (Exception e){ str += "c"; }
    }
    void B() throws Exception{
        try{
            str += "d"; C();
        } catch (Exception e){ throw new Exception(); }
        finally{ str += "e"; }
        str += "f";
    }
    void C() throws Exception { throw new Exception(); }

    public static void main(String[] args) throws Exception{
        Test ob = new Test();
        ob.A();
        System.out.println(ob.str);
    }
}
```

abcdef

După executarea programului, va fi afișată valoarea:

- a) abdec b) abdefc c) abdef d) abcdef

Precizați care dintre următoarele afirmații este adevărată pentru programul dat:

(a) există erori de compilare în clasa Test (b) există erori de compilare în clasa B
(c) va afișa BPBQR după rulare *(d)* va afișa APBQR după rulare

4. Fie următorul program Java:

```
class Persoana implements Serializable {  
    String nume;  
    int varsta;  
  
    public Persoana(String nume, int varsta) {  
        this.nume = nume;  
        this.varsta = varsta;  
        System.out.println("Constructor");  
    }  
}  
  
public class Test {  
    public static void main(String[] args) throws Exception {  
        ObjectOutputStream oos = new ObjectOutputStream(  
            new FileOutputStream("persoana.ser"));  
        Persoana p = new Persoana("Popescu Ion", 40), q = p;  
        oos.writeObject(q);  
        oos.close();  
        ObjectInputStream ois = new ObjectInputStream(  
            new FileInputStream("persoana.ser"));  
        Persoana r = (Persoana)ois.readObject();  
        ois.close();  
    }  
}
```

De câte ori va fi afișat mesajul *Constructor*, după executarea programului dat?

- a) niciodată b) o dată c) de două ori d) de trei ori

5. 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"; }  
    }  
}
```

VERIFICARE LA DISCIPLINA "PROGRAMARE AVANSATĂ PE OBIECTE"
- SESIUNEA MAI/IUNIE 2019 -

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

1. Fie următorul program Java:

```
class Automobil{
    private String marca;
    public Automobil(String marca) { this.marca = marca; }
    public int hashCode() { return 0; }
    public boolean equals(Object obj) { return true; }
}

public class Test {
    public static void main(String[] args) {
        HashSet<Automobil> la = new HashSet<>();
        la.add(new Automobil("Audi"));
        la.add(new Automobil("BMW"));
        la.add(new Automobil("Audi"));
        la.add(new Automobil("Opel"));
        System.out.println(la.size());
    }
}
```

După executarea programului, va fi afișată valoarea:

- (a) 4 (b) 3 (c) 2 (d) 1

2. Precizați care dintre următoarele afirmații este adeverată pentru clasele imutabile:

- (a) sunt implicit și clase de tip singleton
(b) au toate datele membre de tip final
(c) au un singur constructor, fără parametri
(d) referințele spre instanțele lor nu pot fi modificate

3. Fie următorul program Java:

```
class A {
    public static void metoda(String s) {System.out.print("A"+s); }
}

class B extends A {
    public static void metoda(String s) {System.out.print("B"+s); }
    public void metoda(String s, String t) { System.out.print("B"+s+t); }
}

public class Test{
    public static void main(String[] args) {
        A ob = new B();
        ob.metoda("P");
        ob.metoda("Q", "R");
    }
}
```