

Programare orientată pe obiecte

#18 JAVA
Curs final

<https://www.runceanu.ro/adrian/activitate-didactica/>

Curs 18

Curs final JAVA



Structura biletelor de examen:

1. Subiect 1 - Grile cu alegere multiplă. Identificați litera care corespunde răspunsului corect.
2. Subiect 2 - Enunțurile unor probleme având un exemplu specificat. Implementarea obligatorie în JAVA.

Structura biletelor de examen:

A. Proba teoretică

1. Subiect 1 - Grile cu alegere multiplă. Identificați litera care corespunde răspunsului corect.



1. Ce se poate spune despre programul alaturat?

```
public class Program {  
    static int x = 6;  
    public static void main(String[] args) {  
        System.out.print("x = " + x);  
        int x = (x = 3) * x;  
        System.out.print(", x = " + x);  
    }  
}
```

- a) Eroare la compilare: variabila x este declarată de doua ori
- b) Programul se compilează si la executie afisează x = 6, x = 3
- c) Programul se compilează si la executie afisează x = 6, x = 9
- d) Programul se compilează si la executie afisează x = 6, x = 18

2. Ce va afisa la executie programul alaturat?

```
public class Program
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        int a = 3;
```

```
        int b = (a = 2) * a;
```

```
        int c = b * (b = 5);
```

```
        System.out.println( "a = " + a + ", b = " + b + ", c = " + c);
```

```
    }
```

```
}
```

a) a = 2, b = 4, c = 20

b) a = 2, b = 5, c = 20

c) a = 2, b = 5, c = 25

d) a = 3, b = 6, c = 30

3. Ce va afisa urmatoarea secventa de instructiuni?

```
StudyPage.java X
1 package studytool;
2 public class StudyPage{
3     public static void main(String args[]) {
4         int x=9,y=4,z;
5         z=++x-y++;
6         y+=z++;
7         System.out.println("x="+x+"y="+y+"z="+z);
8     }
9 }
```

- a) x=10y=10z=7
- b) x=10y=11z=7
- c) x=11y=10z=6
- d) x=11y=11z=7

4. Ce va afisa urmatoarea secventa de instructiuni?

```
StudyPage.java X
1 package studytool;
2 public class StudyPage{
3     public static void main(String args[]) {
4         int[] v=new int[10];
5         for(int i=0;i<10;i++) v[i]=i+1;
6         System.out.println(v[6]);
7     }
8 }
```

- a) 6
- b) 0
- c) 5
- d) 7

5. Ce se afiseaza dupa executia urmatorului program?

```
StudyPage.java X
1 package studytool;
2 public class StudyPage{
3     public static void main(String args[]) {
4         int[] v=new int[10];
5         int n=7, k=3;
6         for(int i=0;i<n-k;i++)
7             for(int j=0;j<k;j++) v[i+j]=k+i;
8         for(int i=0;i<n;i++) System.out.print(v[i]+" ");
9     }
10 }
```

- a) 1 2 3 4 5 6 0
- b) 2 3 4 5 6 6 0
- c) 3 4 5 6 6 6 0
- d) 3 4 5 7 7 7 0

6. Ce se afiseaza dupa executia urmatorului program?

```
StudyPage.java X
1 package studytool;
2 public class StudyPage{
3     public static void main(String args[]) {
4         int[] v=new int[30];
5         for(int i=0;i<30;i++) v[i]=v.length+i;
6         System.out.print(v[10]);
7     }
8 }
```

- a) 10
- b) 39
- c) 40
- d) 30

7. Ce se afiseaza dupa executia urmatorului program?

```
StudyPage.java X
1 package studytool;
2 public class StudyPage{
3     public static void main(String args[]) {
4         int[] a=new int[10];
5         for(int i=0;i<5;i++) a[i]=i+1;
6         a[1+a[2]]=5;
7         a[a[1]+a[0]]=10;
8         for(int i=0;i<5;i++)System.out.print(a[i]+" ");
9     }
10 }
```

- a) 1 2 3 4 5
- b) 1 2 3 10 5
- c) 2 3 4 5 10
- d) 2 3 4 5 6

8. Ce se afiseaza dupa executia urmatorului program?

```
*grila10.java
1 package grile;
2
3 public class grila10 {
4
5     public static void main(String[] args) {
6         for(int i=0;i<5;i++){
7             switch(i){
8                 case 1: System.out.print("a");
9                 case 2: System.out.print("b");
10                case 3: System.out.print("c");
11                case 4: System.out.print("d");
12                case 5: System.out.print("e");
13                default: System.out.print("x");
14            }
15        }
16    }
17 }
```

- a) Programul afiseaza "xabcd";
- b) Programul afiseaza "abcdex";
- c) Programul afiseaza "xabcdexabcdexabcdexabcde";
- d) Programul afiseaza "xabcdexbcdexcdexdex";

9. Ce se afiseaza dupa executia urmatorului program?

```
grila10.java grila11.java ✕
1 package grile;
2 class Baza {
3     public void functie(int i) {System.out.print(" i = " + i);}
4 }
5
6 public class grila11 extends Baza {
7
8     public void functie(int j) {System.out.print(" j = " + j);}
9     public void functie(String s) {System.out.print(" s = " + s);}
10
11     public static void main(String args[]) {
12         Baza b1 = new Baza();
13         Baza b2 = new grila11();
14         b1.functie(5);
15         b2.functie(6);
16     }
17 }
```

- a) i = 5 i = 6
- b) j = 5 j = 6
- c) i = 5 j = 6
- d) s = 5 s = 6
- e) eroare la compilare deoarece lipseste cuvântul virtual din metoda functie() a clasei Baza

10. Ce se afiseaza dupa executia urmatorului program?

```
grila10.java grila11.java grila12.java grila13.java ✕
1 package grile;
2
3 public class grila13 {
4
5     public static void main(String[] args) {
6         C1 obiect = new C2();
7         System.out.println(
8             obiect.f() + ", " + obiect.g());
9     }
10 }
11
12 class C1 {
13     static String f() {
14         return "Mesajul Unu din C1"; }
15     String g() {return "Mesajul Doi din C1";}
16 }
17 class C2 extends C1 {
18     static String f() {
19         return "Mesajul Unu din C2"; }
20     String g() {return "Mesajul Doi din C2";}
21 }
22
```

- a) programul este corect și va afișa la execuție: Mesajul Unu din C1, Mesajul Doi din C1
- b) programul este corect și va afișa la execuție: Mesajul Unu din C1, Mesajul Doi din C2
- c) programul este corect și va afișa la execuție: Mesajul Unu din C2, Mesajul Doi din C1
- d) programul este corect și va afișa la execuție: Mesajul Unu din C2, Mesajul Doi din C2
- e) va apare eroare la compilare deoarece în clasa grila13 variabila obiect nu aparține clasei C2

11. Ce se afiseaza dupa executia urmatorului program?

```
grila10.java grila11.java grila12.java grila13.java grila14.java ✕
1 package grile;
2
3 public class grila14 {
4
5     public static void main(String[] args) {
6         B bobj = new B();
7         System.out.println(bobj.b);
8     }
9 }
10 class A {
11     int a=1;
12     A() { a=2;}
13 }
14 class B extends A {
15     int b=3;
16     B() { b=a;}
17 }
18 }
```

- a) 1
- b) 2
- c) 3
- d) Nu va afisa nimic deoarece codul este incorect

Structura biletelor de examen:

2. Subiect 2 - Enunțurile unor probleme având un exemplu specificat. Implementarea obligatorie în JAVA.



Problema 1:

Se da urmatorul tablou de cuvinte (siruri de caractere): "Martie", "Aprilie", "Luna", "An".

Sa se scrie un program Java care sa afiseze câte un mesaj care sa specifice cu ce litera începe fiecare cuvânt dat, folosind instructiunea *switch*.

Problema 2:

Să se scrie un program Java prin care să se afișeze numărul de vocale dintr-un text scris cu litere mici, memorat într-o variabilă de tip șir de caractere.

Exemplu:

Date de intrare: Programarea calculatoarelor

Date de ieșire: Exista 12 vocale in text

Problema 3:

Fiind dat un tablou bidimensional $A(n,m)$ de numere intregi, să se scrie un program Java prin care să se afișeze:

- a) produsul elementelor pozitive aflate pe linii pare și coloane impare.
- b) suma elementelor pozitive impare din matrice.
- c) numărul elementelor pozitive, negative și respectiv nule din matrice.
- d) elementele aflate pe linii impare și coloane divizibile cu 3.

Problema 4:

Fiind dat un tablou bidimensional $A(n,m)$ de numere intregi. Sa se scrie un program Java care sa determine linia (liniile) din matrice care contine cele mai multe elemente nenule.

Problema 5:

Clasa Complex. Să se scrie un program Java prin care să se definească o clasă "Complex" pentru operații cu numere complexe și să se testeze metodele implementate. Clasa va avea doi constructori astfel:

- cu doi parametri (parte reală și parte imaginară)
- fără parametri (constructor implicit)

Metodele necesare sunt: adunare, înmulțire, ridicare la putere naturală și afișare (șir de caractere de forma (real,imaginar)).

Problema 6:

Clasa Multime. Să se scrie un program Java prin care să se definească o clasă Multime pentru o multime de numere întregi, care conține un vector numere întregi.

Metodele clasei:

- Constructor cu un parametru care reprezintă dimensiunea multimii
- void add(int): adăugare element, dacă nu există
- boolean contains(int): verifică dacă un număr dat se află sau nu în multime
- afisare: multimea de elemente transformată în șir de caractere

Să se scrie un program care creează o multime folosind clasa Multime și testează operațiile de mai sus prin adăugări succesive și afisări.

Succes la examen!