

Programare orientată pe obiecte

#17 JAVA
Interfața grafică AWT. Applet-uri

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

Curs 17

Interfața grafică AWT. Applet-uri



Interfata grafica AWT. Applet-uri

1. Interfata grafica AWT

2. Applet-uri

2.1. Pagina HTML pentru applet

2.2. Parametrii applet-ului transmisi în pagina HTML

1. Interfata grafica AWT

- Pachetul **AWT** pune la dispozitia utilizatorului o serie de componente grafice:
 - *components*
 - *widgets*
 - *controls*
 - clase pentru evenimente
 - interfete etc.
- Utilizând aceste componente, programatorul le poate combina pentru a construi o anumita interfata grafica (*Graphical User Interface - GUI*).

1. Interfata grafica AWT

- Din punctul de vedere al utilizatorului, componentele grafice AWT sunt caracterizate:
 - printr-o forma
 - printr-un aspect (*look* - ceea ce se vede pe ecran)
 - si prin modul de comportare, ca raspuns la actiunile cu utilizatorului (*feel*)

1. Interfata grafica AWT

- La actiunea utilizatorului asupra unei componente grafice, **JVM(Java Virtual Machine)**, având ca suport sistemul de operare, genereaza asa-numitele *evenimente* (obiecte Java ce contin diferite informatii despre actiunea utilizatorului asupra componentei).
- Evenimentele (obiectele) sunt, mai departe, transmise de JVM unor metode specifice din cadrul programului.

1. Interfata grafica AWT

- Aceste metode sunt scrise de programator, sunt asociate cu componentele grafice si sunt responsabile de actiunea programului la aparitia evenimentelor (la actiunile utilizatorului asupra componentei respective).
- Exista o varietate mare de componente, si fiecare componenta grafica poate genera anumite tipuri de evenimente.

1. Interfata grafica AWT

Tipuri de componente grafice:

- componente de tip **Container** - permit includerea altor componente grafice în interiorul lor
- Pentru crearea de meniuri exista clasele derivate din clasa **MenuComponent**
- Clasa **Component** defineste o serie de metode general valabile pentru componentele grafice **AWT**
- Printre acestea exista metode legate de pozitionarea si dimensiunea componentei, de culorile utilizate pentru afisare, de tipurile de fonturi utilizate, de control al afisarii, etc.

1. Interfata grafica AWT

- Pachetul `java.awt` contine o serie de componente grafice simple, necesare în construirea unei interfete grafice minimale.
- Aceste componente sunt:
 1. `Label` - se foloseste doar pentru **afisarea unei linii de text**, nu interactioneaza cu utilizatorul
 2. `Button` - creaza **imaginea unui buton** asupra caruia se poate actiona prin “apasare”
 3. `Choice` - permite **selectia numai a unui articol** dintr-o lista de articole
 4. `List` - **liste de selectie**
 5. `Checkbox` - **casete de validare** si altele.

1. Interfata grafica AWT

- În vederea **lucrului cu texte**, pachetul `java.awt` contine doua clase instantiabile:
 1. `TextField`, pentru **lucrul cu o singura linie de text**
 2. `TextArea` pentru **lucrul cu texte pe mai multe linii**
- Componentele de tip `container` sunt componentele care furnizeaza cadru de plasare a componentelor grafice.
- Adaugarea de noi componente în interiorul containerelor se face prin metode `add`.

1. Interfata grafica AWT

- Pentru dispunerea componentelor în cadrul unui container se asociază acestuia un **administrator de pozitionare** (**LayoutManager**) care definește regulile de așezare a componentelor în container.
- Clasele disponibile pentru instanțierea containerelor sunt clasele:
 1. **Panel**
 2. **Applet**
 3. **ScrollPane**
 4. **Window**
 5. **Dialog**
 6. **Frame**

Componentele de tip **Panel**, **Applet**, **ScrollPane** pot să fie adăugate la rândul lor în alte containere.
- Cele de tip **Window**, **Dialog**, **Frame** nu permit acest lucru.

1. Interfata grafica AWT

- Pentru **lucrul cu ferestre grafice** package-ul `java.awt` ofera o serie de clase derivate din clasa `Window`:
 1. `Dialog`
 2. `Frame`
 3. `FileDialog`
- Interfata grafica este creata pentru a asigura comunicatia între utilizator si program.
- În **Java** pentru comunicatia între utilizator si program se folosesc evenimente, care sunt generate de componente (**containere** sau **controale**) la actiunile utilizatorului.

1. Interfata grafica AWT

Pentru dezvoltarea interfetelor grafice **AWT** se parcurg mai multe etape, care vor fi evidentiatae în exemplele urmatoare:

1. Se porneste cu crearea unuia sau mai multor containere (obiecte din clase derivate din clasa **Container**).
2. Se creaza controalele (cu operatorul **new**; de exemplu **new Button()**, **new Choice()**).
3. Controalele create se insereaza într-un container, folosind functii **add()** ale containerului.

1. Interfata grafica AWT

4. Modul de inserare a controalelor trebuie sa tina seama de administratorii de pozitionare impliciti sau noi creati ai containerului.
5. Se prevede modul de tratare a evenimentelor generate de componente.
6. Pentru aceasta se creaza clase care implementeaza interfetele care contin metodele necesare tratarii evenimentelor dorite (clase de ascultare, *clase delegat*).

1. Interfata grafica AWT

7. După aceasta, se creează obiecte din clasele de ascultare (delegati) care înregistrează ca ascultatori ai evenimentelor care vor fi tratate.
8. Componentele și evenimentele **AWT** se pot folosi atât pentru **aplicațiile independente** (de sine statatoare - *standalone*) cât și pentru **applet**-uri.

Interfata grafica AWT. Applet-uri

1. Interfata grafica AWT

2. **Applet-uri**

2.1. Pagina HTML pentru applet

2.2. Parametrii applet-ului transmisi în pagina HTML

2. Applet-uri

- Un **applet** este *un program Java care se descarca o data cu o pagina HTML si este executat înauntrul unui browser Web, ca de exemplu Mozilla, Google Chrome, Microsoft Internet Explorer, Opera, Safari, etc.*
- Pentru ca **applet**-urile **Java** sunt programe care se obtin din Internet, se considera ca nu sunt de încredere si ca prezinta numeroase riscuri:
 - virusarea fisierelor sistem ale unui calculator si extinderea virusilor si pe alte discuri,
 - distrugerea sau divulgarea informatiilor existente în calculatorul în care se executa.

2. Applet-uri

- Din motive de securitate, **applet**-urilor li se restrictioneaza operatiile pe care le pot efectua:
 - *Applet-urile nu pot citi sau scrie fisiere pe un disc al unui utilizator.* Daca informatia trebuie salvata pe disc în timp ce un applet este în executie, memorarea informatiei trebuie facuta pe discul de pe care a fost încarcata pagina Web care contine applet-ul.
 - *Applet-urile nu pot realiza o conexiune pe retea spre un computer,* altul decât cel de la care a fost încarcata pagina Web, cu exceptia îndreptarii browser-ului catre o noua locatie.

2. Applet-uri

- *(continuare)* Din motive de securitate, **applet**-urilor li se restrictioneaza operatiile pe care le pot efectua:
 - *Applet-urile nu pot folosi librariile dinamice sau partajate de la orice alt limbaj de programare.* Desi Java poate folosi programe scrise în limbaje ca Visual C++, **applet**-urile nu se pot folosi de aceste caracteristici pentru ca nu exista nici o cale pentru o verificare adecvata a securitatii codului nonJava.

2. Applet-uri

- Un **applet** se definește printr-o clasă publică care extinde clasa **Applet** din pachetul **java.applet**, ca de exemplu:

```
public class HelloWorld extends java.applet.Applet {  
    /* to do */  
}
```

- Clasa **Applet** extinde clasa **java.awt.Panel**, care este o fereastră și un container **Java**.
- Această definiție permite unui **applet** să ocupe un loc fizic în pagina **Web** din fereastra browser-ului și să admită includerea altor componente în interiorul lui.

2. Applet-uri

Metodele clasei Applet

- Structura si comportamentul unui **applet** e stabilit prin executia metodelor acestuia, apelate de browser în diferite situatii.
- Exista versiuni implicite ale fiecareia dintre metode în cazul în care nu se opteaza pentru supraîncarcarea lor.

2. Applet-uri

Daca un **applet** trebuie sa execute functii specifice, atunci una sau mai multe din aceste metode trebuie supraîncarcate.

- **init()**: metoda de initializare folosita pentru a seta starea unui **applet**:
 - încarcarea imaginilor
 - initializarea variabilelor
 - crearea obiectelor
- **start()**: porneste executia **applet**-ului sau îl reporneste daca a fost oprit.

2. Applet-uri

- **stop():**

opreste un **applet** întrerupându-i executia dar îi pastreaza resursele intacte astfel încât acesta sa poata porni din nou.

- Un **applet** trebuie oprit înainte de a-l distruge.

- Metoda poate fi folosita si pentru a opri executia unui **applet** când este necesara o pauza în desfasurarea sa.

2. Applet-uri

- **destroy():**
 - distruge un **applet**, eliberând toate resursele (memorie, spatiul de salvare pe disc).
 - Aceasta este ultima metoda care este apelata atunci când utilizatorul paraseste pagina care contine **applet**-ul.

2. Applet-uri

- Alte metode importante pentru un **applet** sunt metodele **paint()** si **repaint()**, mostenite de **applet** din clasa **Component**.
- Un **applet** poate apela direct functia **repaint()** oricând se doreste reactualizarea ferestrei.
- Aceasta metoda apeleaza metoda **paint()**, care trebuie sa fie redefinita daca se doreste o anumita desenare (afisare) în fereastra **applet**-ului.

2. Applet-uri

- Metoda **paint()** mai este apelata si de masina virtuala în mod automat, ori de câte ori este necesara redesenarea **applet**-ului (la redimensionarea ferestrei browserului în care este inclus **applet**-ul, la descoperirea ferestrei dupa ce a fost acoperita sau minimizata etc.).

2. Applet-uri

Ca exemplu, se definește applet-ul HelloWorld astfel:

```
// HelloWorld.java
```

```
import java.awt.*;  
import java.applet.*;  
public class HelloWorld extends Applet {  
    public void paint(Graphics g) {  
        g.drawString("Hello World !", 10, 10);  
    }  
}
```



Interfata grafica AWT. Applet-uri

1. Interfata grafica AWT

2. Applet-uri

2.1. Pagina HTML pentru applet

2.2. Parametrii applet-ului transmisi în pagina HTML

2.1. Pagina HTML pentru applet

- *Limbajul de Marcare HiperText (HTML)* contine un set de coduri speciale (marcaje) care se insereaza într-un text, pentru a adauga informatii despre formatare si despre legaturi.
- Prin conventie, toate marcajele **HTML** încep cu caracterul “<” si se termina cu caracterul “>” (de exemplu **<HTML>**).
- Acest control, sau marcaj **HTML**, comunica unui interpretor **HTML** (program de navigare) faptul ca documentul este scris si formatat în limbajul **HTML**.
- Unele dintre controalele HTML sunt de tip pereche, având controale de început (de deschidere) si de sfârșit (de închidere) (**<HTML>**, **</HTML>**, **<BODY>**, **</BODY>**).

2.1. Pagina HTML pentru applet

- Compilarea applet-urilor se face tot cu compilatorul **javac** deoarece un applet este compus din clase **Java**.
- Pentru introducerea (de fapt, specificarea) unui applet într-o pagina HTML se folosesc marcajele **<APPLET>** si **</APPLET>**.
- Între aceste marcaje se înscriu mai multe attribute, dintre care obligatorii sunt numele fisierului care contine clasa **applet**-ului (atributul **CODE**) si dimensiunile ferestrei **applet**-ului (**latimea si înaltimea: width, height**).

2.1. Pagina HTML pentru applet

- Ca exemplu, este prezentata pagina HTML care contine `applet`-ul HelloWorld:

```
<html>  
<body>  
  <APPLET code=HelloWorld.class height = 250 width = 250> </APPLET>  
</body>  
</html>
```
- Daca se dau numai attributele de nume si dimensiune ale `applet`-ului, atunci fisierele class ale `applet`-ului trebuie sa se afle în acelasi director cu pagina `HTML` care îl include pentru a fi gasite si descarcate la utilizator (browser).
- Daca `applet`-ul se afla în alt director decât pagina `HTML` care îl include, directorul în care se afla fisierele class ale acestuia se specifica prin proprietatea `CODEBASE` a marcajului `<APPLET>` sub forma unei adrese `URL (Uniform Resource Locator)`.

Interfata grafica AWT. Applet-uri

1. Interfata grafica AWT

2. Applet-uri

2.1. Pagina HTML pentru applet

2.2. Parametrii applet-ului transmisi în pagina HTML

2.2. Parametrii applet-ului transmisi în pagina HTML

- Pentru transmiterea unor date **applet**-urilor se poate folosi marcajul **<PARAM>** din limbajul **HTML**, care definește numele și valoarea unui parametru.
- Acești parametri pot fi cititi în funcția **start()** a **applet**-ului, memorati și apoi folositi în funcționarea acestuia.
- Ca exemplu, se modifica **applet**-ul HelloWorld și se obține **applet**-ul HelloNume, care afiseaza un sir de caractere ce include numele și prenumele unei persoane salutate, iar aceste date se introduc ca parametri în pagina HTML care insereaza **applet**-ul.

```
// HelloNume.java
import java.awt.*;
import java.applet.*;
public class HelloNume extends Applet {
    String nume = null;
    String prenume = null;
    public void start(){
        nume = getParameter("Nume");
        prenume = getParameter("Prenume");
    }
    public void paint(Graphics g) {
        String str = "Hello " + nume + " " + prenume + " !";
        g.drawString(str, 20, 20);
    }
}
```

Pagina HTML contine doi parametri (Nume si Prenume), ale caror valori care sunt citite la pornirea **applet**-ului (în functia start()) a acestuia).

```
<html>
```

```
<body>
```

```
  <applet code=HelloNume.class height = 250  
width = 250>
```

```
  <PARAM NAME="Nume" VALUE="Popescu">
```

```
  <PARAM NAME="Prenume" VALUE="Costica">
```

```
  </applet>
```

```
</body>
```

```
</html>
```

Exemple de grille Java

Ce se va afișa la execuția următorului program Java?

```
public class Grila1{  
public static void main(String args[]){  
    String s1 = "anul" + 20 + 16, s2 = 20 + 16 +  
    "anul";  
    System.out.println("s1 = " + s1 + ", s2 = " +  
    s2);  
    }  
}
```

s1 = anul2016, s2 = 2016anul

Ce se va afișa la execuția următorului program Java?

```
public class Grila2{  
    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);  
    }  
}
```

x = 6, x = 9

Ce se va afișa la execuția următorului program Java?

```
public class Grila4{
    public static void main(String args[]){
        A.B a = new A().new B();
        a.f();
    }
}
class A {
    private int x = 4;
    public class B{
        private int y = 5;
        public void f(){
            System.out.println("x = " + x);
            x = 3;
            System.out.println("x = " + x + " y = " + y);
        }
    }
}
```

x = 4

x = 3 y = 5

Ce se va afișa la execuția următorului program Java?

```
public class Grila4{  
public static void main(String args[]){  
    int a,b;  
    for(a=10, b=1; a-- > ++b; ) ;  
        System.out.println((a--) + "," + (++b));  
    }  
}
```

5,7

Proba practica:

Se considera un vector

$$\mathbf{x} = \{ x_0, x_1, x_2, \dots, x_{n-1} \}.$$

Sa se scrie un program **Java** care sa modifice vectorul astfel incat in final sa avem:

$$\mathbf{x} = \{ x_1, x_2, \dots, x_{n-1}, x_0 \}$$

```
import java.util.*;
public class vector7 {
public static void main(String[] args) throws IOException
{
    int n;
    Scanner input = new Scanner(System.in);
    System.out.print("dati numarul de linii: ");
    n = input.nextInt();
    int X[] = new int[n];
    int i;
    for(i=0;i<n;i++)
    {
        System.out.print("X["+i+"]=");
        X[i]=input.nextInt();
    }
}
```

```
System.out.print("Vectorul initial : ");  
for(i=0;i<n;i++)  
    System.out.print(" " + X[i]);
```

```
int a=X[0];  
for(i=0;i<n-1;i++)  
    X[i]=X[i+1];  
X[n-1]=a;
```

```
System.out.println(" ");  
System.out.print("Vectorul final : ");  
for(i=0;i<n;i++)  
    System.out.print(" " + X[i]);  
}  
}
```

Rezultatul executiei:

Introduceti numarul de elemente $n = 5$

$X[0]=1$

$X[1]=2$

$X[2]=3$

$X[3]=4$

$X[4]=5$

Vectorul initial: 1 2 3 4 5

Vectorul final: 2 3 4 5 1

Probleme propuse spre rezolvare

Probleme propuse spre rezolvare:

1) Să se afișeze toate numerele palindroame mai mari decăt 10 și mai mici decăt un număr dat, n .

Exemplu:

Date de intrare: $n=110$

Date de ieșire:

11 22 33 44 55 66 77 88 99 101

Probleme propuse spre rezolvare:

2) Să se determine toate tripletele de numere a, b, c cu proprietățile: $1 < a < b < c < 100$; $a+b+c$ se divide cu 10.

Exemplu:

Date de intrare: -

Date de ieșire: 95 96 99 și 95 97 99 sunt ultimele două triplete

Probleme propuse spre rezolvare:

3) Să se afișeze toate numerele de două cifre care adunate cu răsturnatul lor dau 55.

Exemplu:

Date de intrare: -

Date de ieșire: 14 41; 23 32; 32 23; 41 14

Probleme propuse spre rezolvare:

4) Se introduce un număr natural cu maxim 9 cifre. Să se determine și să se afișeze numărul de cifre, cea mai mare cifră, cea mai mică cifră și suma tuturor cifrelor acestui număr.

Exemplu:

Date de intrare: 24356103

Date de ieșire:

Numarul de cifre 8

Cea mai mare cifra 6

Cea mai mica cifra 0

Suma cifrelor 24

Probleme propuse spre rezolvare:

5) Dat un număr întreg de maxim 9 cifre, să se afișeze numărul de apariții al fiecărei cifre.

Exemplu:

Date de intrare 364901211

Date de ieșire:

0 apare de 1 ori

1 apare de 3 ori

2 apare de 1 ori

3 apare de 1 ori

4 apare de 1 ori

5 apare de 0 ori

6 apare de 1 ori

7 apare de 0 ori

8 apare de 0 ori

9 apare de 1 ori

Probleme propuse spre rezolvare:

6) Intersectia a doua multimi

Sa se scrie un program care sa calculeze intersectia a doua multimi de cate n , respectiv m numere intregi memorate cu ajutorul vectorilor.

Exemplu:

Date de intrare:

$n = 5$ $a = \{1, 2, 5, 7, 12\}$ si

$m = 4$ $b = \{2, 4, 8, 12\}$

Date de iesire:

se obtine: $k = 2$ $d = \{2, 12\}$

Probleme propuse spre rezolvare:

7) Diferența a două mulțimi

Sa se scrie un program care sa calculeze diferenta a doua multimi de cate n, respectiv m numere intregi memorate cu ajutorul vectorilor.

Exemplu:

Date de intrare:

$n = 5$ $a = \{1, 2, 5, 7, 12\}$ si

$m = 4$ $b = \{2, 4, 8, 12\}$

Date de iesire:

se obtine: $k = 3$ $e = \{1, 5, 7\}$

Probleme propuse spre rezolvare:

8) Se dă un șir de n numere naturale. Să se afișeze pe două randuri, pe primul rand cele pare și pe al doilea cele impare.

Exemplu:

Date de intrare:

Pentru $n = 10$

și elementele 4 3 2 5 6 8 9 0 1 5

Date de ieșire:

4 2 6 8 0

3 5 9 1 5

Probleme propuse spre rezolvare:

9) Sa se scrie un program care sa introducă n numere intr-un vector și să citească un număr d. Să se afișeze acele numere din șirul dat care sunt divizibile cu d.

Exemplu:

Date de intrare:

n=5 si valorile: 5 7 10 23 15, d=5

Date de iesire:

5 10 15

Probleme propuse spre rezolvare:

10) Se introduc temperaturile măsurate în n zile. Să se scrie un program care să afișeze media temperaturilor negative și media celor pozitive.

Exemplu:

Date de intrare:

$n = 5$ și temperaturile: 23 24 23 25 22

Date de ieșire:

23.40

Probleme propuse spre rezolvare:

11) Dintr-un șir de numere întregi, să se afișeze suma elementelor de pe pozițiile pare și suma celor de pe poziții impare.

Exemplu:

Date de intrare:

$n=6$ și numerele 3 1 5 4 6 7

Date de ieșire:

$s_{\text{imp}} = 14$ (am însumat elem. 3+5+6)

$s_{\text{par}} = 12$ (am însumat elem. 1+4+7)

Probleme propuse spre rezolvare:

12) Se introduc în calculator și se memorează vârstele a n bărbați. Afișați câți au vârsta mai mare decât 50 și mai mică decât 70 de ani.

Exemplu:

Date de intrare:

$n=4$ și vârstele 56 45 76 65

Date de ieșire:

2 barbati

Probleme propuse spre rezolvare:

12) Fiind date n numere naturale, câte sunt prime cu 7? Dar cu 10?

Exemplu:

Date de intrare:

$n=6$ si numerele 5 7 14 20 12 15

Date de ieşire:

prime cu 7: 4 numere (5,20,12,15)

prime cu 10: 1 numar (7)

Doua numere sunt prime între ele dacă cmmdc al lor este 1.

Probleme propuse spre rezolvare:

13) Se dă un tablou bidimensional cu n linii și m coloane, $1 \leq n, m \leq 50$, cu componente întregi și un număr întreg k . Se cere să se afișeze tabloul cu componentele mărite cu k .

Exemplu:

Date de intrare:

$n = 2, m = 3, k = 5$ și elementele:

1 2 3

6 7 8

Date de ieșire:

6 7 8

11 12 13

Probleme propuse spre rezolvare:

14) Fiind dat un tablou bidimensional cu n linii și n coloane, afișați cea mai mare componentă și pozițiile pe care le ocupă.

Exemplu:

Date de intrare:

$n = 2$ și matricea:

4 7

5 7

Date de ieșire:

max = 7 se afla pe pozițiile 1,2 și 2,2

Probleme propuse spre rezolvare:

15) Să se afișeze suma elementelor de pe coloana k a unei matrici cu n linii și m coloane, $k \leq m$.

Exemplu:

Date de intrare:

$n = 2, m = 4, k = 3$ și matricea

4 3 6 8

7 4 9 0

Date de ieșire:

$s = 15$

Probleme propuse spre rezolvare:

16) Să se afișeze toate **prefixele** și **sufixe** unui cuvânt citit de la tastatură.

Exemplu:

Date de intrare: informatica

Date de ieșire:

i
in
inf
info
infor
inform
informa
informat
informati
informatic
informatica

a
ca
ica
tica
atica
matica
rmatica
ormatica
formatica
nformatica
informatica

Probleme propuse spre rezolvare:

17) Se citește de la tastatură un caracter c și un text de maxim 100 de caractere. Afișați **de câte ori apare caracterul c** în cadrul textului. Literele mari se vor considera diferite de literele mici.

Exemplu:

Date de intrare:

Acesta este un simplu exemplu

Caracterul e

Date de ieșire:

Litera 'e' apare de 5 ori

Probleme propuse spre rezolvare:

18) Se citește de la tastatură un șir de maxim 100 de caractere format numai din litere și cifre. Afișați **numărul literelor mari, numărul literelor mici și numărul caracterelor de tip cifră** din textul dat.

Exemplu:

Date de intrare:

S-a nascut la Targu Jiu in 1968 pe 21 martie

Date de ieșire: Se gasesc 25 de litere mici, 2 litere mari si 6 cifre

Probleme propuse spre rezolvare:

19) Se da un vector v cu n elemente numere intregi.

Sa se scrie un program care sa calculeze:

- Suma elementelor din vectorul v
- Numarul componentelor impare din vectorul v
- Elementul maxim din vectorul v

Exemplu:

Date de intrare:

$n = 9$

$v = \{2, 9, 15, 6, 3, 12, 66, 8, 5\}$

Date de iesire:

a) 126

b) 4

c) 66

Probleme propuse spre rezolvare:

20) Se dau n numere intregi. Sa se scrie un program care sa gaseasca numarul mai mic sau egal decat n , care sa aiba cei mai multi divizori. Se va folosi o functie care calculeaza numarul de divizori ai unui numar.

Exemplu:

Date de intrare: $n = 100$

**Date de iesire:
60 (are 12 divizori)**

Probleme propuse spre rezolvare:

21) Se da un numar natural n . Sa se scrie un program care sa stabileasca raportul dintre numarul divizorilor pari si numarul divizorilor impari ai numarului n .

Exemplu:

Date de intrare: $n = 246$

Date de iesire:

1 (4 divizori pari si 4 divizori impari)

Probleme propuse spre rezolvare:

22) Pentru o matrice formată din n linii și m coloane ($1 < n$, $m < 10$) să se scrie un program care afișează liniile conținând k elemente nule ($0 < k \leq m$).

Se va afișa un mesaj în situația în care nici o linie nu conține exact k elemente nule.

Exemplu:

Date de intrare:

$n=2, m=4, k=2$

1 0 2 3

0 3 0 5

Date de iesire:

Linia 2

Probleme propuse spre rezolvare:

23) Se dă o matrice pătratică de dimensiune $n \times n$ precizată, scrieți un program care stabilește dacă:

- matricea este simetrică față de diagonala principală
- matricea este inferior triunghiulară, adică toate elementele situate deasupra diagonalei principale sunt nule

Exemplu:

Date de intrare:

$n=4$

1 11 22 33

11 1 44 55

22 44 1 66

33 55 66 1

Date de iesire:

Matricea este

simetrica

Probleme propuse spre rezolvare:

24) Scrieți o funcție care să stabilească dacă un număr dat n conține în reprezentarea sa zecimală o anumită cifră precizată, notată, de exemplu, c .

Se va utiliza apoi această funcție pentru a afișa toți întregii cu valori cuprinse între 1 și p (p citit de la tastatură) pentru care numărul, pătratul și cubul reprezentării sale conțin aceeași cifră.

Probleme propuse spre rezolvare:

Exemplu:

Date de intrare:

$c=5$

$p=200$

Date de iesire:

5 (valorile 5, $25=5^2$, $125=5^3$ contin cifra 5)

Întrebări?