

/\*Plesa Diana lab06

1. Scrieti un program Java care defineste un sir de valori de tip double care vor fi citite de la tastatura. Trati exceptia

data de accesarea unui element cu index negativ sau mai mare decat indexul ultimului element din sir (ArrayIndexOutOfBoundsException). Afisati un mesaj corespunzator in momentul detectarii exceptiei.

\*/

```
package lab06;
```

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.print("Introduceti numarul de valori: ");
            int n = scanner.nextInt();
            double[] array = new double[n];
            System.out.print("Valorile: ");
            for (int i = 0; i < n; i++) {
                array[i] = scanner.nextDouble();
            }

```

```

            System.out.print("Array elements: ");
            for (int i = 0; i <= array.length; i++) {
                System.out.print(array[i] + " ");
            }
        } catch (Exception e) {
            System.out.println("\nException: " + e + "\n");
        }

```

```
    }
```

```
}
```

/\*Plesa Diana lab06

2. Scrie o aplicatie Java care, în cadrul metodei main( ), conține o secvență de cod care poate să lanseze excepții de tip

ArithmeticException, ArrayIndexOutOfBoundsException, NullPointerException, NumberFormatException și altele pe care

considera că e util să le testezi. Afișează în cadrul blocurilor catch mesajul generat de aceste excepții. În blocul finally,

afișează mesajul "Am prins sau nu excepții".\*/

```
package lab06;
```

```
public class Main {
```

```
    public static void main(String[] args) {
        try {
            System.out.println("Impartire cu 0");
            System.out.println("Impartim 5 la 0. Rezultat: ERROARE");

```

```

        System.out.println(5/0);
    }
    catch (ArithmeticException ex) {
        System.out.println("Arithmetic Exception prinsa: " + ex);
    }
    finally {
        System.out.println("Am prins sau nu exceptii");
    }
    System.out.println(" ");

    try {
        System.out.println("Accesarea unui element în afara limitelor");
        int[] array = {10,6,1,5,14,100};

        System.out.println("Tabloul: ");
        showArray(array);
        System.out.println("Accesarea elementului de pe pozitia 7: " + array[7]);
    }
    catch (ArrayIndexOutOfBoundsException ex) {
        System.out.println("ArrayIndexOFB prinsa: " + ex);
    }
    finally {
        System.out.println("Am prins sau nu exceptii");
    }

    System.out.println(" ");

    try {
        System.out.println("Accesarea unui obiect nul si incrementarea");
        Integer integer = null;
        System.out.println("The object is: " + (integer + 1));
    }
    catch (NullPointerException ex) {
        System.out.println("NullPointerException prinsa: " + ex);
    }
    finally {
        System.out.println("Am prins sau nu exceptii");
    }
}

private static void showArray(int[] array) {
    for (Integer it: array) {
        System.out.print(it + " ");
    }
    System.out.println(" ");
}
}

```

/\*Plesa Diana lab06

3. Definiți un pachet care declară interfața Int1 (2 variabile întregi și o metodă numită sum() care returnează suma a doi întregi). În pachet mai există clasa Class1 (2 variabile protected de tip double, constructor, metode acces or și mutator pentru variabilele din clasă). În alt fișier sursă se adaugă în același pachet o nouă interfață numită Int2 (2 var

abile de tip

double i o metodă numită product( ) care returnează produsul a două valori de tip double).

Într-un fișier sursă distinct, importa i tot ce conține pachetul definit anterior i defini i o clasă numită Class2 care este

derivată din Class1 i care implementează ambele interfețe Int1 i Int2. Instanțiază i Class2 i aplică i metodele disponibile

pentru a calcula suma i produsul unor variabile citite de la tastatură.

\*/

```
package lab06;
```

```
import java.util.Scanner;
```

```
import laborator6.*;
```

```
public class Main {  
    static Scanner scanner = new Scanner(System.in);
```

```
    public static void main(String[] args) {  
        System.out.println("Introduceți 2 valori: ");  
        double a, b;  
        a = scanner.nextDouble();  
        b = scanner.nextDouble();  
        Class2 object = new Class2(a, b);  
  
        System.out.println("Suma numerelor este: " + object.sum());  
        System.out.println("Produsul numerelor este: " + object.product());  
    }  
}
```

```
package lab06;
```

```
import laborator6.*;
```

```
public class Class2 extends Class1 {
```

```
    public Class2(double a, double b) {  
        super(a, b);  
    }
```

```
    public int sum() {  
        return (int) (a + b);  
    }
```

```
    public double product() {  
        return a * b;  
    }
```

```
}
```

```
package laborator6;
```

```
public abstract class Class1 implements Int1, Int2 {
```

```
protected double a, b;
```

```
public Class1(double a, double b) {  
    this.a = a;  
    this.b = b;  
}
```

```
public double getA() {  
    return a;  
}
```

```
public void setA(double a) {  
    this.a = a;  
}
```

```
public double getB() {  
    return b;  
}
```

```
public void setB(double b) {  
    this.b = b;  
}
```

```
}
```

```
package laborator6;
```

```
public interface Int1 {  
    int a = 0, b = 0;  
  
    int sum();  
}
```

```
package laborator6;
```

```
public interface Int2 {  
    double a = 0, b = 0;  
  
    double product();  
}
```

```
/*Plesa Diana lab06
```

4. Creați o clasă derivată din Exception, numită SuperExceptie. O a doua clasă ExceptieMaiMica este derivată din

SuperExceptie. În constructorii claselor afișați un mesaj ce indică excepția generată. În cadrul unei alte clase creați o

metodă a() ce semnalizează (aruncă) o excepție de tip ExceptieMaiMica și o metodă b() ce semnalizează o excepție de

tip SuperExceptie.

În cadrul metodei main() apelați aceste metode și încercați să determinați tipul de excepție semnalizată, precum și dacă

blocul catch aferent tipului de excepție ExceptieMaiMica poate prinde o SuperExcepție.

```
*/
```

```
package lab06;
```

```
public class Main {
```

```
    public static void main(String[] args) {  
        Throw object = new Throw();  
        object.a();  
        object.b();  
    }
```

```
}
```

```
package lab06;
```

```
public class Throw {
```

```
    void a() {  
        try {  
            throw new ExceptieMaiMica("Eroare aparuta in metoda a() ");  
        } catch (ExceptieMaiMica ex) {  
            ex.printStackTrace();  
        }  
    }
```

```
    void b() {  
        try {  
            throw new SuperExceptie("Eroare aparuta in metoda b()");  
        } catch (SuperExceptie ex) {  
            ex.printStackTrace();  
        }  
    }  
}
```

```
package lab06;
```

```
public class ExceptieMaiMica extends SuperExceptie {
```

```
    public ExceptieMaiMica(String message) {  
        super("Exceptie mai mica EROARE");  
    }
```

```
}
```

```
package lab06;
```

```
public class SuperExceptie extends Exception {
```

```
    public SuperExceptie(String message) {  
        super("Super exceptie EROARE");  
    }
```

```
}
```

```
/*PlesaDiana lab03
```

```
1. Scrieti o aplicatie Java in care cititi o valoare intreaga. In cazul in care valoarea intreaga este cuprinsa  
intre 1-12
```

```
sa se afiseze luna corespunzatoare. Considerati cazul in care valoarea introdusa este un sir de caractere  
si daca
```

```
aceasta corespunde unei luni din an afisati valoarea numerica a lunii.*/
```

```
package lab3;
```

```
import java.util.Arrays;
```

```
import java.util.Scanner;
```

```
import java.util.List;
```

```
public class lab3 {
```

```
    static Scanner in=new Scanner(System.in);
```

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

```
        Integer[] myIntArray = {1,2,3,4,5,6,7,8,9,10,11,12};
```

```
        String [] LunileAnului = {"Ianuarie","Februarie","Martie","Aprilie","Mai","Iunie","Iulie","August",  
        "Septembrie","Octombrie","Noiembrie","Decembrie"};
```

```
        int n, i, j, ok=0;
```

```
        System.out.println("Introduceti un numar: ");
```

```
        n = in.nextInt();
```

```
        if(n > 12 || n < 1)
```

```
        {
```

```
            System.out.println("Nu poate fi luna!");
```

```
        }
```

```
        else
```

```
        System.out.println("Luna corespunzatoare numarului este: ");
```

```
        for(i = 0; i < myIntArray.length; i++)
```

```
        {
```

```
            if(myIntArray[i] == n)
```

```
                System.out.println(LunileAnului[i]);
```

```
        }
```

```
        Scanner sc= new Scanner(System.in);
```

```
        System.out.print("\nIntroduceti un sir de caractere: \n");
```

```
        String sir= sc.next();
```

```
        for(j = 0; j < LunileAnului.length; j++)
```

```
        {
```

```
            if(sir.equals(LunileAnului[j]))
```

```
            {
```

```
                System.out.println("Numarul corespunzator sirului este: ");
```

```
                System.out.println(myIntArray[j]);
```

```
                ok = 1;
```

```
            }
```

```
        }
```

```
        if(ok == 0)
```

```
        System.out.println("Sirul introdus nu corespunde unei luni! ");
```

```
    }
```

```
}
```

```
/*PlesaDiana lab03
```

```
2. Cititi un sir de caractere de la intrarea standard. Transformati sirul intr-un tablou de caractere. Cautati i  
n acest  
tablou un caracter specificat in program. Afisati numarul de aparitii.*/
```

```
package lab3;
```

```
import java.util.Arrays;  
import java.util.Scanner;
```

```
public class lab3 {
```

```
    private static Scanner sc = new Scanner(System.in);
```

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

```
        int i, ap = 0;
```

```
        System.out.print("Introduceti un sir de caractere: \n");
```

```
        String sir= sc.next();
```

```
        char [] a=sir.toCharArray();
```

```
        System.out.print("\nIntroduceti caracterul de cautat in sir: \n");
```

```
        String c = sc.next();
```

```
        for(i = 0; i < a.length; i++)
```

```
            if (c.charAt(0) == a[i]) ap++;
```

```
        System.out.println("Numarul de aparitii a carcterului in sir este: \n" + ap);
```

```
    }
```

```
}
```

```
/*PlesaDiana lab03
```

```
3. Pornind de la problema precedenta, copiatu primele 3 caractere din sir intr-un alt tablou si afisati rezult  
atul  
obtinut (folositi metoda arraycopy() din clasa System).  
*/
```

```
package lab3;
```

```
import java.util.Arrays;  
import java.util.Scanner;
```

```
public class lab3 {
```

```
    private static Scanner sc = new Scanner(System.in);
```

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

```
        int i;
```

```
        System.out.print("Introduceti un sir de caractere: \n");
```

```
        String sir= sc.next();
```

```
        char [] a=sir.toCharArray();
```

```
        char[] b = new char[3];
```

```
        System.arraycopy(a,0,b,0,3);
```

```
        System.out.print("\nPrimele 3 caractere din sir sunt: \n");
```

```
        for(i = 0; i < b.length; i++)
```

```
            {
```

```
                System.out.print(b[i] + " ");
```

```
    }  
}  
}
```

/\*PlesaDiana lab03

4. Scrieti o aplicatie Java care defineste o valoare intreaga si apoi afiseaza reprezentarea ei binara, octala si hexazecimala. Scrieti metode de conversie a valorii citite in diferite baze de numeratie.\*/

```
package lab3;
```

```
import java.util.Scanner;
```

```
public class lab3 {
```

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

```
        Scanner scanner = new Scanner(System.in);
```

```
        int number;
```

```
        System.out.print("Conversia folosind integer.toTypeString!\n");
```

```
        System.out.print("Introduceti o valoare: ");
```

```
        number = scanner.nextInt();
```

```
        System.out.println(number + " in Baza 2: " + Integer.toBinaryString(number));
```

```
        System.out.println(number + " in Baza 8: " + Integer.toOctalString(number));
```

```
        System.out.println(number + " in Baza 16: " + Integer.toHexString(number));
```

```
    }
```

```
}
```

/\*PlesaDiana lab03



5. Implementati algoritmi cunoscuti de ordonare a sirurilor numerice (bubble sort, insertion sort, quick sort, etc.)

si aplicati-i asupra unui sir de valori intregi citite de la tastatura.\*/

```
package lab3;
```

```
import java.util.Arrays;
import java.util.Scanner;
```

```
public class lab3 {
```

```
    private static Scanner scan = new Scanner(System.in);
```

```
    public static void main(String[] args) {
        System.out.println("Introduceti cate valori se citesc: ");
        int nr = scan.nextInt();
        int[] arr = new int[nr];
        System.out.println("Introduceti sirul de numere: ");
        for (int i = 0; i < nr; i++)
        {
            arr[i] = scan.nextInt();
        }
    }
```

```
    System.out.println("\nSirul initial este: " + Arrays.toString(arr));
```

```
    int[] aux = arr;
    bubblesort(aux);
    System.out.println("Bubblesort: " + Arrays.toString(aux));
```

```
    aux = arr;
    insertionsort(aux);
    System.out.println("Insertionsort: " + Arrays.toString(aux));
}
```

```
static void bubblesort(int array[]) {
    for (int i = 0; i < array.length; i++)
    {
        for (int j = 0; j < array.length - 1; j++)
        {
            if (array[j + 1] < array[j])
            {
                int aux = array[j];
                array[j] = array[j + 1];
                array[j + 1] = aux;
            }
        }
    }
}
```

```
static void insertionsort(int array[]) {
    for (int i = 1; i < array.length; i++)
    {
        int arrayr = array[i], j = i - 1;
        while (arrayr < array[j] && j >= 0)
```

```

    {
        array[j + 1] = array[j];
        --j;
    }
    array[j + 1] = arrayr;
}
}
}

```

/\*PlesaDiana lab03

6. Scrieti un program Java care genereaza 2 valori aleatoare si efectueaza asupra lor cateva operatii matematice.\*/

```
package lab3;
```

```
import java.util.Random;
```

```

public class lab3 {
    public static void main(String[] args) {
        Random rand = new Random();
        int randNo1 = rand.nextInt(100);
        int randNo2 = rand.nextInt(100);

        System.out.println("Numerele sunt: " + randNo1 + " " + randNo2);
        System.out.println("Adunare: " + (randNo1 + randNo2));
        System.out.println("Scadere: " + (randNo1 - randNo2));
        System.out.println("Inmultire: " + (randNo1 * randNo2));
        System.out.println("Impartire: " + (randNo1 / randNo2));
        System.out.println("Radicalul numerelor: " + Math.sqrt(randNo1) + " " + Math.sqrt(randNo2));
    }
}

```

/\*Plesa Diana lab9

1. Să se creeze o aplicație Java în cadrul căreia există o clasă ce implementează interfața Runnable. Constructorul clasei permite definirea unui nume asociat fiecărui obiect instanțiat din clasa respectivă și de asemenea clasa are un atribut static ce contorizează numărul de obiecte instanțiate. Metoda run() a clasei va afișa numele obiectului de un număr de ori egal cu valoarea contorului și cu o întârziere de 1000 msec între afișări.\*/

```
package lab9;
```

```
import java.util.Scanner;
```

```
public class Main {
```

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

```

        Scanner sc=new Scanner(System.in);
        System.out.print("Introduceti un nume: ");
        String sir=sc.next();
        Thread sir1 =new Thread(new MyRunnable(sir));
        System.out.print("Introduceti un nume: ");
        sir=sc.next();
        Thread sir2 =new Thread(new MyRunnable(sir));
        System.out.print("Introduceti un nume: ");
    }
}

```

```

        sir=sc.next();
        Thread sir3 =new Thread(new MyRunnable(sir));
        sir1.start();
        sir2.start();
        sir3.start();
    }
}

```

```
package lab9;
```

```

public class MyRunnable implements Runnable{
    String nume;
    static int contor=0;
    public MyRunnable(String nume){
        this.nume=nume;
        contor++;
    }
    public void run() {
        for(int i=0;i<contor;i++)
        {
            try {
                System.out.println(nume);
                Thread.sleep(1000);

            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```
/*Plesa Diana lab9 grupa 2127
```

2. Scrieti o aplicatie Java in care un fir de executie genereaza 30 numere aleatoare intre 0 si 30 si un alt fir de executie

afiseaza aria cercului cu raza generata de primul fir de executie, daca aceasta raza este divizibila cu 3.

```
*/
```

```
package lab9;
```

```
public class main {
```

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

```

        try {
            Numar numar = new Numar();
            Thread nrThread = new Thread(numar);
            nrThread.start();

```

```

            Cerc cerc = new Cerc(numar);
            Thread cercThread = new Thread(cerc);
            cercThread.start();

```

```
        } catch (Exception e) {
```

```
    e.printStackTrace();
}
}
```

```
package lab9;
```

```
import java.util.Arrays;
import java.util.Random;
```

```
public class Numar implements Runnable {
    int[] numere = new int[30];
```

```
    public Numar() {}
```

```
    public void run() {
        generateNumere();
        displayNumere();
    }
```

```
    public void generateNumere() {
        Random rand = new Random();
        for (int i = 0; i < 30; i++) {
            numere[i] = rand.nextInt(30) + 1;
        }
    }
```

```
    public void displayNumere() {
        System.out.println("Tablou: " + Arrays.toString(numere));
    }
```

```
}
```

```
package lab9;
```

```
public class Cerc implements Runnable {
    Numar numar;
```

```
    public Cerc(Numar numar) {
        this.numar = numar;
    }
```

```
    public void Aria(int[] sir) {
        int numarCerc = 0;
        for (int i = 0; i < sir.length; i++) {
            if (sir[i] % 3 == 0) {
                numarCerc++;
                System.out.print("\nCercul[" + numarCerc + "] are aria: " + Math.PI * Math.pow(sir[i], 2));
            }
        }
    }
```

```
}
```

```

public void run() {
    Aria(numar.numere);
}
}

```

/\*Plesa Diana lab9 grupa 2127

3. Scrieti o clasa pentru determinarea unui element din sirul lui Fibonacci. In clasa vor fi 2 metode, una care calculeaza valoarea elementului si alta care afiseaza acea valoare. Folositi mecanismul de multithreading cu sincronizare, astfel incat in timp ce un fir de executie realizeaza calculul, celalalt sa afiseze valorile intermediare (toate componentele sirului pana la elementul calculat).

```

package lab9;

```

```

import java.util.Scanner;

```

```

public class main {

```

```

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Introduceti numarul dorit: ");
        int nr = scanner.nextInt();
        scanner.close();

```

```

        if (nr < 0)
            System.out.println("Numar gresit! Program terminat!"); // invalid index
        else {
            try {
                Thread fibonacciThread = new Thread(new Fibonacci(nr));
                Thread outputThread = new Clasa();

```

```

                fibonacciThread.start();
                outputThread.start();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

```

```

    }

```

```

}

```

```

package lab9;

```

```

public class Fibonacci implements Runnable {
    public static volatile int fn = 0;
    public static volatile boolean isPrinted = false;
    boolean firstTime = true;
    int index;

```

```

    public Fibonacci(int index) {

```

```

    this.index = index;
}

public void run() {
    calcNumar();
}

public synchronized void calcNumar() {
    if (index == 1) {
        afisareNumere(0);
    } else if (index == 2) {
        afisareNumere(1);
    } else {

        if (firstTime) {
            try {
                Thread.sleep(500);
                System.out.println(1);

            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        int n1 = 0, n2 = 1;
        for (int i = 2; i < index; i++) {
            isPrinted = false;
            fn = n1 + n2;
            n1 = n2;
            n2 = fn;

            while (!isPrinted) {
                try {
                    Thread.sleep(500);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }

            afisareNumere(fn);
        }

    }

    public void afisareNumere(int numar) {
        System.out.println("Numarul dorit este: " + numar);
    }

}

package lab9;

public class Clasa extends Thread {

```

```

public Clasa() {
    setDaemon(true);
}

public void run() {
    while (true) {
        while (Fibonacci.isPrinted) {
            try {
                Thread.sleep(500);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        System.out.println(Fibonacci.fn);
        Fibonacci.isPrinted = true;
    }
}
}
}

```

/\*Plesa Diana lab9 grupa 2127

4. Crea i o aplica ie Java ce folose te accesul sincronizat la metode pentru excludere mutuală. Crea i 3 fire d e execu ie ce apelează simultan metode de incrementare i decrementare a unei variabile dintr-o altă clasă. Verifica i dacă rezultatele sunt cele a teptate. Elimina i blocurile sincronizate i re-evalua i rezultatele.  
\*/

```

package lab9;

```

```

public class main {

    public static void main(String[] args) {
        Numar nr = new Numar(10);

        Incrementrare first = new Incrementrare(nr);
        Decrementare second = new Decrementare(nr);
        Incrementrare third = new Incrementrare(nr);

        first.start();
        second.start();
        third.start(); // 10+8-5 = 6

        System.out.println("Valoarea este: " + nr.nr);
    }

}

```

```

package lab9;

```

```

public class Decrementare extends Thread {
    Numar numar;

    public Decrementare(Numar nr) {

```

```

    this.numar = nr;
}

public void run() {
    decrement();
}

public void decrement() {
    synchronized (numar.nr) {
        try {
            numar.nr -= 5;
        } catch (Exception e) {
            System.out.println("Thread intrerupt.");
        }
    }
}

}

package lab9;

public class Incrementrare extends Thread {
    Numar numar;

    public Incrementrare(Numar nr) {
        this.numar = nr;
    }

    public void run() {
        increment();
    }

    public void increment() {
        synchronized (numar.nr) {
            try {
                numar.nr += 4;
            } catch (Exception e) {
                System.out.println("Thread intrerupt.");
            }
        }
    }
}

package lab9;

public class Numar {
    Integer nr;

    public Numar(Integer nr) {
        this.nr = nr;
    }
}

```



/\*Plesa Diana lab08

1. Folosind mecanismul preluării datelor de la tastatură cu un `BufferedReader` și `InputStreamReader`, preluăm: un mesaj de tip `String`, ziua ca întreg, luna de tip `String` și anul ca un întreg. Procesul se va încheia în momentul trecerii la o nouă linie, când se va afișa fiecare câmp pe o linie separată. Afișați de asemenea toate câmpurile în ordinea în care au apărut.

Recomandare: Folosiți clasa `StreamTokenizer`, atributele `sval`, `nval` respectiv constanta `TT_EOL`. Considerați în cazul în care nu cunoașteți natura câmpurilor introduse (numeric, cuvânt) folosind constantele

`TT_NUMBER`, `TT_WORD`.\*/

```
package lab8;
```

```
import java.io.*;
```

```
import static java.io.StreamTokenizer.*;
```

```
public class main {
```

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

```
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
```

```
        StreamTokenizer streamTokenizer = new StreamTokenizer(bufferedReader);
```

```
        streamTokenizer.eolIsSignificant(true);
```

```
        try {
```

```
            System.out.print("Introduceti data: ");
```

```
            while(streamTokenizer.nextToken() != TT_EOF)
```

```
            {
                if(streamTokenizer.ttype == TT_WORD)
```

```
                {
```

```
                    System.out.println("Luna: " + streamTokenizer.sval);
```

```

        } else if(streamTokenizer.ttype == TT_NUMBER) {

            System.out.println("Numar: " + (int)streamTokenizer.nval);

        } else if(streamTokenizer.ttype == TT_EOL) {

            System.out.println("Sfarsitul liniei");

            break;

        }

    } catch(Exception ex) {

        ex.printStackTrace();

    }

}

}

}

/*Plesa Diana lab08
2. Implementa i aplica ia anterioara folosind un fi ier de intrare care con ine un anumit numar de inregistrari
cu
formatul specificat mai sus.*/

package lab8;

import java.io.*;
import static java.io.StreamTokenizer.*;

public class main {

    public static void main(String[] args) {
        BufferedReader bufferedReader;
        try {
            bufferedReader = new BufferedReader(new FileReader("fisier.txt"));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            return;
        }

        StreamTokenizer streamTokenizer = new StreamTokenizer(bufferedReader);
        streamTokenizer.eollsSignificant(true);
        try {
            while (streamTokenizer.nextToken() != TT_EOF) {
                if (streamTokenizer.ttype == TT_WORD) {
                    System.out.println("Sir: " + streamTokenizer.sval);
                } else if (streamTokenizer.ttype == TT_NUMBER) {
                    System.out.println("Numar: " + (int) streamTokenizer.nval);
                }
            }
        }
    }
}

```

```

    } else if(streamTokenizer.ttype == TT_EOL) {

        System.out.println("Sfarsitul liniei");
        break;
    }
}
} catch (Exception ex) {
    ex.printStackTrace();
}
}
}
}

```

/\*Plesa Diana lab08

3. Se citesc de la tastatură date formatate sub forma DD/MM/YYYY. Să se afi eze sub forma DD luna YY YY, unde luna este forma expandată a MM i de asemenea să se afi eze i dacă anul este bisect. Programul se opre te prin apăsarea tastei X sau x. Puteti folosi si clasa DateFormatSymbols pentru conversia lunii.\*/

```
package lab8;
```

```

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.StreamTokenizer;
import java.text.NumberFormat;
import java.util.Scanner;

```

```
public class main {
```

```

    public static void main(String[] args) throws FileNotFoundException {
        try {
            BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(new FileInputStream("
fisier.in")));
            StreamTokenizer streamTokenizer = new StreamTokenizer(bufferedReader);
            streamTokenizer.whitespaceChars('/', '/');
            streamTokenizer.wordChars(' ', ' ');
            int iterator = 0;

```

```

            NumberFormat nf = NumberFormat.getInstance();
            nf.setGroupingUsed(false);

```

```

            while (streamTokenizer.ttype != StreamTokenizer.TT_EOF) {
                int token = streamTokenizer.nextToken();
                if (token == StreamTokenizer.TT_NUMBER) {
                    if (iterator == 1) {
                        int x = (int) streamTokenizer.nval;
                        getYourMonth(x);
                    } else {
                        if (iterator == 0) { // ziua
                            if (streamTokenizer.nval < 10) {
                                System.out.print("0" + nf.format(streamTokenizer.nval) + " ");
                            } else

```

```

        System.out.print(nf.format(streamTokenizer.nval) + " ");
    } else { // anul
        System.out.print(nf.format(streamTokenizer.nval));
        leapYear((int)streamTokenizer.nval);
    }
}
iterator++;
if (iterator > 2) {
    iterator = 0;
    System.out.print("\n");
}
}
}
System.out.println("\nSfarsitul fisierului (EOF)");

} catch (IOException e) {
    e.printStackTrace();
}

exitProgram();
}

```

```

public static void getYourMonth(int x) {
    switch (x) {
    case 1:
        System.out.println("Ianuarie ");
        break;
    case 2:
        System.out.println("Februarie ");
        break;
    case 3:
        System.out.println("Martie ");
        break;
    case 4:
        System.out.println("Aprilie ");
        break;
    case 5:
        System.out.println("Mai ");
        break;
    case 6:
        System.out.println("Iunie ");
        break;
    case 7:
        System.out.println("Iulie ");
        break;
    case 8:
        System.out.println("August ");
        break;
    case 9:
        System.out.println("Septembrie ");
        break;
    case 10:
        System.out.println("Octombrie ");
        break;
    case 11:

```

```

    System.out.println("Noiembrie ");
    break;
case 12:
    System.out.print("Decembrie ");
    break;
default:
    System.out.print("Luna incorecta!");
    break;
}
}

public static void exitProgram() {
    System.out.println("\nApasati X/x pentru a iesi din program!");
    Scanner scanner = new Scanner(System.in);
    System.out.print("Tasta: ");
    String key = scanner.next();
    while (!key.equalsIgnoreCase("X") || !key.equalsIgnoreCase("x")) {
        System.out.print("\nIncercati din nou!\nTasta: ");
        key = scanner.next();
    }
    System.out.print("<<<<<Program terminat>>>>>");
}

public static void leapYear(int year) {
    boolean isLeap = false;
    if (year % 4 == 0) {
        if (year % 100 == 0) {
            if (year % 400 == 0)
                isLeap = true;
            else
                isLeap = false;
        } else
            isLeap = true;
    } else {
        isLeap = false;
    }

    if (isLeap == true)
        System.out.print("\n"+year + " este an bisect.\n");
}
}

```

/\*Plesa Diana lab08

4. Se da un fiier binar ce contine o secventa de caractere ce reprezinta o cheie privata pentru Bitcoin. De la tastatura se citeste o secventa de caractere ce reprezinta cheia publica a unui Bitcoin. Sa se genereze identificatorul de tranzactie corespunzator acestor informatii folosind operatia de sau exclusiv (XOR) pe biile aplicate intre cheile publica si privata. Scrie informatia rezultata intr-un fiier binar.\*//

```

package lab8;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;

```

```

import java.io.IOException;

public class main {

    public static void main(String[] args) throws FileNotFoundException {

        FileInputStream in = new FileInputStream("public.dat");
        FileInputStream inPrivat = new FileInputStream("privat.dat");
        FileOutputStream out = new FileOutputStream("out.dat");

        byte buf[] = new byte[256];
        byte buf2[] = new byte[256];

        try {

            while (in.read(buf) != -1 && inPrivat.read(buf2) != -1) {
                for (int i = 0; i < 64; i++) {
                    Byte buf3 = new Byte((byte) (buf[i] ^ buf2[i]));
                    out.write(buf3);
                }
            }

            in.close();
            out.close();
            inPrivat.close();

        } catch (IOException e) {
            System.out.println("Eroare la citirea din fisier!");
        }
    }
}

```

/\*PlesaDiana lab05

\* 2. Defini i o interfa ă denumită FormaGeometrica care are 2 metode care primesc ca parametru o valoare integer i returnează o valoare double reprezentînd aria, respectiv perimetrul unei forme geometrice echilaterale. Implementa i interfa a în 3 clase distincte pentru forme concrete de tip pătrat, triunghi echilateral i hexagon regulat.  
Se preiau datele pentru un număr N de forme geometrice de fiecare tip.  
Să se calculeze aria i perimetrul total al formelor geometrice, înînd cont că acestea nu se suprapun.\*/

```

package lab05;

import java.util.Scanner;

public class Main {
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.println("Introduceti numarul de forme: ");
        int N = scanner.nextInt();
        Forma object = new Forma();
        object.read(N);
    }
}

```

```
    object.display();  
}  
}
```

```
package lab05;
```

```
public interface FormaGeometrica {  
  
    default double ariaPatrat(double l) {  
        return l * l;  
    }  
    default double perimetruPatrat(double l) {  
        return 4 * (l);  
    }  
    default double ariaTriunghiEchi(double l) {  
        return l * Math.sqrt(3) / 4;  
    }  
  
    default double perimetruTriunghiEchi(double l) {  
        return 3 * l;  
    }  
    default double perimetruHexagon(double l) {  
        return 6 * l;  
    }  
    default double ariaHexagon(double l) {  
        return (3*Math.sqrt(3)/2)*l*l;  
    }  
}
```

```
package lab05;
```

```
import java.util.LinkedList;  
import java.util.List;  
import java.util.Scanner;
```

```
public class Forma implements FormaGeometrica{  
  
    static Scanner scanner = new Scanner(System.in);  
    List<Patrat> patrat = new LinkedList<>();  
    List<TriunghiEchi> triunghiEchi = new LinkedList<>();  
    List<Hexagon> hexagon = new LinkedList<>();  
    public void read(int n) {  
        System.out.print("1.Patrat\n2.Triunghi echilateral\n3.Hexagon\n\n");  
        int option;  
        for (int i = 0; i < n; i++) {  
            System.out.println("\nOptiunea pentru figura " + (i + 1) + " : ");  
            option = scanner.nextInt();  
            switch (option) {  
                case 1:  
                    addPatrat();  
                    break;  
                case 2:  
                    addTriunghiEchi();  
                    break;  
                case 3:  
                    addHexagon();  
                    break;  
            }  
        }  
    }  
}
```

```

        break;
    case 3:
        addHexagon();
        break;
    }
}

private void addTriunghiEchi() {
    System.out.println("Introduceti lungimea laturii triunghiului: ");
    double length = scanner.nextDouble();
    TriunghiEchi object = new TriunghiEchi(length);
    triunghiEchi.add(object);
}

private void addPatrat() {
    System.out.println("Introduceti lungimea laturii patratului: ");
    double length = scanner.nextDouble();
    Patrat object = new Patrat(length);
    patrat.add(object);
}

private void addHexagon() {
    System.out.println("Introduceti lungimea laturii hexagonului: ");
    double length = scanner.nextDouble();
    Hexagon object = new Hexagon(length);
    hexagon.add(object);
}

public void display() {
    System.out.println("\nCe forma doriti: "
        + "\n1.Patrat\n2.Triunghi echilateral\n3.Hexagon" + "\n\nOptiunea: ");
    int option = scanner.nextInt();
    System.out.println();

    switch (option) {
        case 1:
            for (int i = 0; i < patrat.size(); i++) {
                System.out
                    .println("Patratul " + (i + 1) + " aria: " + String.format("%.2f", ariaPatrat(patrat.get(i).l))
                        + " si perimetrul: " + String.format("%.2f", perimetruPatrat(patrat.get(i).l)));
            }
            break;
        case 2:
            for (int i = 0; i < triunghiEchi.size(); i++) {
                System.out.println("Triunghiul echilateral " + (i + 1) + " aria: "
                    + String.format("%.2f", ariaTriunghiEchi(triunghiEchi.get(i).l)) + " si perimetrul: "
                    + String.format("%.2f", perimetruTriunghiEchi(triunghiEchi.get(i).l)));
            }
            break;
        case 3:
            for (int i = 0; i < hexagon.size(); i++) {
                System.out.println("Hexagonul " + (i + 1) + " aria: "
                    + String.format("%.2f", ariaTriunghiEchi(hexagon.get(i).l)) + " si perimetrul: "
                    + String.format("%.2f", perimetruHexagon(hexagon.get(i).l)));
            }
            break;
        default:
            System.out.println("\nWrong input!");
    }
}

```



```
break;
}
}
}
```

```
package lab05;
```

```
public class Patrat {
    double l;

    public Patrat(double length) {
        this.l = length;
    }
}
```

```
package lab05;
```

```
public class TriunghiEchi {
    double l;

    public TriunghiEchi(double l) {
        this.l = l;
    }
}
```

```
package lab05;
```

```
public class Hexagon {
    double l;

    public Hexagon(double l) {
        this.l = l;
    }
}
```

```
/*Plesa Diana lab05
```

1. Să se implementeze o interfață Java care conține prototipurile metodelor de adunare, scădere, înmulțire, împărțire, radical și ridicare la putere. Toate metodele vor avea doi parametri de tip double și vor specifica tipul returnat double.

Implementați interfața astfel încât operațiile să fie definite în cadrul unei clase. Instanțiați clasa și verificați operațiile implementate. \*/

```
package lab5;
```

```
import java.util.Scanner;
```

```
public class Main {

    private static Scanner scanner = new Scanner(System.in);
```

```

public static void main(String[] args) {

    System.out.print("Introduceti doua numere: ");
    double a, b;
    a = scanner.nextDouble();
    b = scanner.nextDouble();

    Operatii object = new Operatii();

    System.out.print("\nSuma numerelor este: " + object.adunare(a, b));
    System.out.print("\nDiferenta numerelor este: " + object.scadere(a, b));
    System.out.print("\nProdusul numerelor este: " + object.inmultire(a, b));
    System.out.print("\nCatul numerelor este: " + object.impartire(a, b));
    System.out.print("\nRadical de ordinul a din b este: " + object.radical(a, b));
    System.out.print("\na la puterea b este: " + object.putere(a, b));

}

}

```

```

package lab5;

```

```

import java.lang.Math;

```

```

public class Operatii implements Interfata {

    public double adunare(double a, double b) {
        return a + b;
    }

    public double scadere(double a, double b) {
        return a - b;
    }

    public double inmultire(double a, double b) {
        return a * b;
    }

    public double impartire(double a, double b) {
        return a / b;
    }

    public double putere(double a, double b) {
        return Math.pow(a , b);
    }

    public double radical(double a, double b) {
        return Math.pow(b, 1/a);
    }
}

```

```

package lab5;

```

```

public interface Interfata {

```

```
public double adunare(double a, double b);
public double scadere(double a, double b);
public double inmultire(double a, double b);
public double impartire(double a, double b);
public double radical(double a, double b);
public double putere(double a, double b);
```

```
}
```

/\*Plesa Diana lab11 grupa 2127

1. Implementa i o aplica ie Java care folose te GridLayout pentru aranjarea în pagină a componentelor unei aplica ii de

calcul (TextField pt. intrare/ie ire, butoane pentru numere, opera ii matematice, tergere a ecranului, etc.).

\*/

```
package lab11;
```

```
public class main {
    public static void main(String[] args) {
        new Clacul();
    }
}
```

```
package lab11;
```

```
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JTextArea;
```

```
public class Clacul {
    JFrame frame;
```

```
    public Clacul() {
        initUI();
    }
```

```
    public void initUI() {
        frame = new JFrame("Aplicatie de calcul GridLayout");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 400);
```

```
        GridLayout grid = new GridLayout(5, 3, 10, 10);
```

```
        frame.setLayout(grid);
        JTextArea text = new JTextArea();
        text.setText("Rezultat: ");
        text.setEditable(false);
        frame.add(text);
```

```
        addOperatii();
```

```
        addNumere();
```

```

frame.setVisible(true);
}

private void addNumere() {
    JButton[] tablouBtn = new JButton[10];

    for (int i = 0; i < tablouBtn.length; i++) {
        tablouBtn[i] = new JButton(Integer.toString(i));
        frame.add(tablouBtn[i]);
    }
}

private void addOperatii() {
    frame.add(new JButton("-"));
    frame.add(new JButton("+"));
    frame.add(new JButton("="));
    frame.add(new JButton("<-"));
}
}

```

/\*Plesa Diana lab11 grupa 2127

2. Implementa i o aplica ie Java care folose te GridBagLayout pentru aranjarea în pagină a componentelor unei aplica ii de calcul (TextField pt. intrare/ie ire, butoane pentru numere, opera ii matematice, tergere a ecranului, etc.).

\*/

```

package lab11;

public class main {
    public static void main(String[] args) {
        new Clacul();
    }
}

package lab11;

import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import javax.swing.*;

public class Clacul extends JFrame {

    private static final long serialVersionUID = 1L;

    JPanel jPanel;

    public Clacul() {
        initUI();
    }
}

```

```

public void initUI() {
    this.setSize(400, 400);
    this.setLocationRelativeTo(null);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setTitle("Calculator");

    JPanel = new JPanel();
    JPanel.setLayout(new GridBagLayout());

    GridBagConstraints gridConstraints = new GridBagConstraints();
    gridConstraints.gridx = 1;
    gridConstraints.gridy = 1;

    gridConstraints.gridwidth = 1;
    gridConstraints.gridheight = 1;

    gridConstraints.weightx = 50;
    gridConstraints.weighty = 100;

    gridConstraints.insets = new Insets(5, 5, 5, 5);
    gridConstraints.anchor = GridBagConstraints.CENTER;
    gridConstraints.fill = GridBagConstraints.BOTH;

    JTextField textResult = new JTextField("0", 20);
    Font font = new Font("Helvetica", Font.PLAIN, 18);
    textResult.setFont(font);

    JButton clearAll = new JButton("C");
    JPanel.add(clearAll, gridConstraints);
    gridConstraints.gridwidth = 20;
    gridConstraints.gridx = 5;
    JPanel.add(textResult, gridConstraints);

    addButtons(gridConstraints);

    add(JPanel);

    setVisible(true);
}

private void addButtons(GridBagConstraints gridConstraints) {
    JButton but1, but2, but3, but4, but5, but6, but7, but8, but9, but0, butPlus, butEgal;
    but1 = new JButton("1");
    but2 = new JButton("2");
    but3 = new JButton("3");
    but4 = new JButton("4");
    but5 = new JButton("5");
    but6 = new JButton("6");
    but7 = new JButton("7");
    but8 = new JButton("8");
    but9 = new JButton("9");
    butPlus = new JButton("+");
    but0 = new JButton("0");
    butEgal = new JButton("=");

```

```

gridConstraints.gridwidth = 1;
gridConstraints.gridx = 1;
gridConstraints.gridy = 2;
jPanel.add(but1, gridConstraints);

gridConstraints.gridx = 5;
jPanel.add(but2, gridConstraints);

gridConstraints.gridx = 9;
jPanel.add(but3, gridConstraints);

gridConstraints.gridx = 1;
gridConstraints.gridy = 3;
jPanel.add(but4, gridConstraints);

gridConstraints.gridx = 5;
jPanel.add(but5, gridConstraints);

gridConstraints.gridx = 9;
jPanel.add(but6, gridConstraints);

gridConstraints.gridx = 1;
gridConstraints.gridy = 4;
jPanel.add(but7, gridConstraints);

gridConstraints.gridx = 5;
jPanel.add(but8, gridConstraints);

gridConstraints.gridx = 9;
jPanel.add(but9, gridConstraints);

gridConstraints.gridx = 1;
gridConstraints.gridy = 5;
jPanel.add(butPlus, gridConstraints);

gridConstraints.gridx = 5;
jPanel.add(but0, gridConstraints);

gridConstraints.gridx = 9;
jPanel.add(butEgal, gridConstraints);
}

}

```

/\*Plesa Diana lab11 grupa 2127

4. Crea i o aplica ie ce preia din două câmpuri text (TextField) numele vostru i grupa din care face i parte i afi ează

această informa ie într-o etichetă de culoare RGB (122,123,129).

\*/

```
package lab11;
```

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

```

```
    Aplicatie app = new Aplicatie();  
    app.setVisible(true);  
}  
}
```

```
package lab11;
```

```
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import javax.swing.*;
```

```
public class Aplicatie extends JFrame {  
    private JTextField TextFieldstanga;  
    private JTextField TextFielddreapta;  
    private JLabel eticheta;  
    JButton rezultat;
```

```
    private static final long serialVersionUID = 1L;
```

```
    public Aplicatie() {  
        initUI();  
    }
```

```
    private void initUI() {  
        setSize(new Dimension(350, 450));  
        setTitle("Aplicatie date");  
        setBackground(Color.GREEN);  
        setLayout(null);
```

```
        TextFieldstanga = new JTextField("Nume");  
        TextFieldstanga.setBounds(50, 25, 230, 30);  
        TextFieldstanga.addMouseListener(new MouseAdapter() {  
            public void mousePressed(MouseEvent e) {  
                if (TextFieldstanga.getText().equals("Nume"))  
                    TextFieldstanga.setText("");  
            }  
        });
```

```
        public void mouseExited(MouseEvent e) {  
            if (TextFieldstanga.getText().trim().length() == 0) {  
                TextFieldstanga.setText("Nume");  
            }  
        }  
    }  
};  
add(TextFieldstanga);
```

```
        TextFielddreapta = new JTextField("Grupa");  
        TextFielddreapta.setBounds(50, 55, 230, 30);  
        TextFielddreapta.addMouseListener(new MouseAdapter() {  
            public void mousePressed(MouseEvent e) {  
                if (TextFielddreapta.getText().equals("Grupa"))  
                    TextFielddreapta.setText("");  
            }  
        });
```

```

public void mouseExited(MouseEvent e) {
    if (TextFieIddreapta.getText().trim().length() == 0) {
        TextFieIddreapta.setText("Grupa");
    }
}
});
add(TextFieIddreapta);

eticheta = new JLabel("Asteptare date!", SwingConstants.CENTER);
eticheta.setBounds(35, 225, 250, 30);
add(eticheta);

```

```

rezultat = new JButton("Rezultat");
rezultat.setBounds(115, 350, 100, 30);
add(rezultat);

```

```

rezultat.addActionListener(e -> handle(e));
setVisible(true);
}

```

```

private void handle(ActionEvent e) {
    String nume = TextFieIdstanga.getText();
    String grupa = TextFieIddreapta.getText();

    if ((nume.equals("Nume") || grupa.equals("Grupa"))) {
        eticheta.setForeground(Color.red);
        eticheta.setText("DATE INCORECTE!");
    } else if (nume.trim().length() == 0 || grupa.trim().length() == 0) {
        eticheta.setForeground(Color.red);
        eticheta.setText("Adugati numele/grupa!");
    } else {
        eticheta.setForeground(new Color(122, 123, 129));
        eticheta.setText(nume + " is from group: " + grupa);
    }
}
}
}

```

/\*Plesa Diana lab11 grupa 2127

5. Implementa i o aplica ie graficã în care utilizatorul poate scrie (în componente de tip TextField) cantitã ile R, G, B ale unei culori. După apãsarea unui buton, o componentã de tip Label va afi a un text care reprezintã cantitã il e specificate, cu culoarea rezultatã, dacã valorile sunt corecte (numerice, între 0-255). În caz contrar, eticheta va afi a cu ro u un mesaj de eroare. Încerca i sã folosi i excep ii pentru verificarea valorilor.

\*/

```

package lab11;

```

```

public class main {
    public static void main(String[] args) {
        Aplicatie app = new Aplicatie();
        app.setVisible(true);
    }
}

```



```

package lab11;

import java.awt.*;
import java.awt.event.ActionEvent;
import javax.swing.*;

public class Aplicatie extends JFrame {

    private static final long serialVersionUID = 1L;

    private JTextField jTextField1;
    private JTextField jTextField2;
    private JTextField jTextField3;
    private JLabel eticheta;
    private JButton buton;

    public Aplicatie() {
        init();
    }

    public void init() {
        setSize(new Dimension(400, 200));
        setTitle("Aplicatie RGB");
        setLayout(null);

        JLabel eticheta1 = new JLabel("Introduceti o culoare RGB (0-255,0-255,0-255)");
        eticheta1.setBounds(80, 5, 270, 30);
        add(eticheta1);

        jTextField1 = new JTextField("Color");
        jTextField1.setBounds(50, 40, 100, 30);
        add(jTextField1);

        jTextField2 = new JTextField("Color");
        jTextField2.setBounds(50, 70, 100, 30);
        add(jTextField2);

        jTextField3 = new JTextField("Color");
        jTextField3.setBounds(50, 100, 100, 30);
        add(jTextField3);

        buton = new JButton("Rezultat: ");
        buton.setBounds(50, 140, 100, 30);
        add(buton);

        eticheta = new JLabel("Rezultat");
        eticheta.setBounds(200, 140, 150, 30);
        add(eticheta);

        buton.addActionListener(e -> actionPerformed(e));
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {

```

```

try {
    String rosuVal = jTextField1.getText();
    String verdeVal = jTextField2.getText();
    String albastruVal = jTextField3.getText();

    if ((Integer.parseInt(rosuVal) < 0) || (Integer.parseInt(rosuVal) > 255)
        || (Integer.parseInt(verdeVal) < 0) || (Integer.parseInt(verdeVal) > 255)
        || (Integer.parseInt(albastruVal) < 0) || (Integer.parseInt(albastruVal) > 255))
        throw new RGBIndexOutOfBoundsException("O valoare este introdusa incorect!");

    eticheta.setText(rosuVal + " " + verdeVal + " " + albastruVal);
    eticheta.setForeground(
        new Color(Integer.parseInt(rosuVal), Integer.parseInt(verdeVal), Integer.parseInt(albastruVal)));
} catch (RGBIndexOutOfBoundsException ex) {
    eticheta.setText("Una sau mai multe valori introduse incorect!");
    eticheta.setForeground(Color.RED);
}
}
}

```

```
package lab11;
```

```
public class RGBIndexOutOfBoundsException extends Exception {
```

```
    private static final long serialVersionUID = 1L;
```

```
    public RGBIndexOutOfBoundsException(String message) {
        super(message);
    }
}

```

```
/*PlesaDiana lab04
```

```

1. Scrie i o clasă Methods in care să ave i 3 metode: o metodă cu număr variabil de parametri de tip intreg,
care
returnează media aritmetică a parametrilor; o altă metodă în care să defini i un tablou unidimensional de n
numere de tip double (n fiind parametrul metodei i care va fi citit în main() de la tastatură), elementele ace
stuia
vor primi valori generate aleator i metoda va returna acest tablou; o metodă statică ce prime te ca paramet
ru
un tablou de caractere, converte te caracterele respective în majuscule i formează din acest ir un obiect
String, pe care îl returnează. Din metoda main(), definită într-o altă clasă, apela i metodele clasei Methods
i
afi a i rezultatele pe ecran (valorile de apel pentru prima metodă i irul de caractere cu care se apelează a 3
-a
metodă se vor defini în main()).
*/

```

```
package lab4;
```

```
import java.util.Arrays;
import java.util.Scanner;
```

```
public class lab4 {
    static Scanner scanner = new Scanner(System.in);

```

```

public static void main(String[] args) {
    int a, b, c;
    System.out.print("Introduceti 3 valori: ");
    a = scanner.nextInt();
    b = scanner.nextInt();
    c = scanner.nextInt();

    Methods object = new Methods();
    System.out.println("Media aritmetica este: " + object.MediaAritmetica(a, b, c));

    System.out.print("\nIntroduceti numarul de elemente pentru tablou: ");
    int n = scanner.nextInt();
    double[] array = new double[n];
    array = object.createArray(n);
    System.out.println("Noul tablou este: " + Arrays.toString(array));

    System.out.print("\nIntroduceti numarul de caractere: ");
    n = scanner.nextInt();
    char[] string = new char[n];
    System.out.print("Introduceti caracterele: ");
    for (int i=0;i<n;i++) {
        string[i]=scanner.next().charAt(0);
    }
    System.out.println("Sirul este: " + object.majuscule(string));
}
}

```

```

package lab4;

```

```

import java.util.Random;

```

```

public class Methods {

    public float MediaAritmetica(int... numbers) {
        float result = 0;
        for (int i : numbers)
            result += i;

        return result / numbers.length;
    }

    public double[] createArray(int n) {
        double[] array = new double[n];
        Random random = new Random();
        for (int i = 0; i < n; i++) {
            array[i] = random.nextDouble();
        }
        return array;
    }

    static public String majuscule(char[] string) {
        for (int i = 0; i < string.length; i++) {
            string[i] = Character.toUpperCase(string[i]);
        }
    }
}

```

```

    }
    return new String(string);
}
}

```

/\*Plesa Diana lab04

2. Definiți o clasă X care va avea ca atribute un șir de caractere privat, un obiect de tip String cu specificat  
 orul  
 protected și un întreg fără specificator de acces. Clasa va conține un constructor explicit cu parametri, care va  
 inițializa atributele, metode accesori și mutator pentru câmpurile care nu pot fi accesate din afara clasei și va  
 redefini metoda finalize() moștenită din clasa Object, care va afișa un mesaj corespunzător.  
 În metoda main(), plasată în altă clasă, creați un obiect din clasa X, afișați valorile atributelor, apoi pregătiți  
 obiectul pentru a fi eliminat de garbage collector. Creați apoi un nou obiect X, modificați valorile atributelor și  
 afișați pe ecran valorile acestora.\*/

```

package lab4;

```

```

import java.util.Arrays;
import java.util.Scanner;

```

```

public class lab4 {

```

```

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Introduceți un șir: ");
        String string = scanner.next();

```

```

        System.out.print("Introduceți 5 caractere: ");
        char[] array = new char[5];
        for (int i = 0; i < 5; i++) {
            array[i] = scanner.next().charAt(0);
        }

```

```

        System.out.print("Introduceți un număr: ");
        int number = scanner.nextInt();
        X object = new X(string, array, number);
        System.out.println("Valorile sunt: " + object.number + ", " + object.string + ", " + Arrays.toString(object.ge
tArray()));
        object=null;
        System.gc();

```

```

        System.out.print("Introduceți un șir: ");
        String string1 = scanner.next();

```

```

        System.out.print("Introduceți 5 caractere: ");
        char[] array1 = new char[5];
        for (int i = 0; i < 5; i++) {
            array[i] = scanner.next().charAt(0);
        }
    }
}

```

```

System.out.print("Introduceti un numar: ");
int number1 = scanner.nextInt();
X objectnew = new X(string, array, number);
System.out.println("Valorile sunt: " + objectnew.number + ", " + objectnew.string + ", " + Arrays.toString(o
bjectnew.getArray()));
}
}

```

```

package lab4;

```

```

public class X {
    protected String string;
    private char[] array;
    int number;

    public X(String string, char[] array, int number) {
        this.string = string;
        this.array = array;
        this.number = number;
    }

    protected void finalize() throws Throwable {
        System.out.println("\nObject garbage collected : " + this);
    }

    public char[] getArray() {
        return array;
    }

    public void setArray(char[] array) {
        this.array = array;
    }
}

```

```

/*Plesa Diana lab04

```

4. Scrie i o clasă Java numită Persoană cu atributele private: nume (String), latitudine i longitudine (float).

Clasa

con ine un constructor explicit fără parametri care ini ializează numele cu null i latitudinea i longitudinea cu 0.

Metoda finalize() va fi implementată pentru a afi a un mesaj la distrugerea obiectelor. Atributele clasei se p ot

modifica prin metodele mutator specifice setNume(), setLatitudine() i setLongitudine(). Atributele vor fi returnate prin metode accesori specifice. Instan ia i n obiecte, citi i i setati valorile atributelor, apoi afi a i datele tuturor obiectelor. \*/

```

package lab4;

```

```

import java.util.ArrayList;
import java.util.Scanner;

```

```

public class lab4 {

```

```
private static Scanner scan = new Scanner(System.in);
```

```
public static void main(String[] args) {  
    System.out.println("Introduceti numarul de persoane: ");  
    int noPersoane = scan.nextInt();  
    if (noPersoane > 0) {  
        ArrayList<Persoana> persoane = new ArrayList<>();  
        readInfo(persoane, noPersoane);  
        displayInfo(persoane, noPersoane);  
    }  
    else {  
        System.out.println("\nINVALID INPUT!");  
    }  
}
```

```
public static void readInfo(ArrayList<Persoana> persoane, int n) {  
    System.out.println("Introduceti datele: ");  
    String Nume;  
    Float lat, longi;  
    for (int i = 0; i < n; i++) {  
        Persoana object = new Persoana();  
        System.out.println("\nNumele: ");  
        Nume = scan.next();  
  
        System.out.println("Latitudine: ");  
        lat = scan.nextFloat();  
  
        System.out.println("Longitudine: ");  
        longi = scan.nextFloat();  
  
        object.setName(Nume);  
        object.setLatitudine(lat);  
        object.setLongitudine(longi);  
  
        persoane.add(object);  
    }  
}
```

```
public static void displayInfo(ArrayList<Persoana> persoane, int n) {  
    for (int i = 0; i < n; i++) {  
        System.out.println("\nNumele pentru persoana " + (i + 1) + " este " + persoane.get(i).getName());  
        System.out.println("Latitudinea pentru persoana " + (i + 1) + " este " + persoane.get(i).getLatitudine());  
        System.out.println("Longitudinea pentru persoana " + (i + 1) + " este " + persoane.get(i).getLongitudine());  
    }  
}
```

```
package lab4;
```

```
public class Persoana {  
    private String nume;  
    private float latitudine;  
    private float longitudine;
```

```

public Persoana() {
    latitudine = 0;
    longitudine = 0;
}

public Persoana(String nume, float latitudine, float longitudine) {
    this.nume = nume;
    this.latitudine = latitudine;
    this.longitudine = longitudine;
}

public String getName() {
    return nume;
}

public void setName(String nume) {
    this.nume = nume;
}

public float getLatitudine() {
    return latitudine;
}

public void setLatitudine(float latitudine) {
    this.latitudine = latitudine;
}

public float getLongitudine() {
    return longitudine;
}

public void setLongitudine(float longitudine) {
    this.longitudine = longitudine;
}

@Override
protected void finalize() throws Throwable {
    System.out.println("\nOBJECT DESTROYED!");
}
}

```

/\*Plesa Diana lab04

6. Definiți un vector unidimensional de valori de tip char, având numărul mai mic sau cel mult egal cu un n citit de la tastatură. Fiecare valoare conține valori alfa-numerice generate aleator. Transformați vectorul de caractere într-un obiect de tip String. Afișați valoarea inițială și apoi valoarea obținută prin înlocuirea fiecărui caracter numeric cu '0'.

package lab4;

```

import java.util.Scanner;

public class lab4 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Introduceti dimensiunea sirului: ");
        int dimensiune = scanner.nextInt();
        String string = randomString(dimensiune);
        System.out.println("Sirul initial este: " + string);

        string = string.replaceAll("[0-9]", "");
        System.out.println("Sirul prelucrat este: " + string);
    }

    private static String randomString(int length) {
        String letterNumber = "abcdefghijklmnopqrstuvwxyz1234567890";
        char[] string = new char[length];
        for (int i = 0; i < length; i++) {
            int rand = (int) (Math.random() * letterNumber.length());
            string[i] = letterNumber.charAt(rand);
        }
        return new String(string);
    }
}

```

/\*Plesa Diana lab10 grupa 2127

1. Scrie i o aplica ie de tip Frame în care afi a i într-un Panel numele vostru i grupa din care face i parte. Text ul va fi de culoare albastră i centrat atât pe orizontală, cât i pe verticală, inând cont de dimensiunea panelului.\* /

```

package lab10;

import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.util.Scanner;

public class JPanelClass extends JPanel {

    private static final long serialVersionUID = 1L;
    static String nume1;
    static String nume2;
    static String grupa;

    public JPanelClass( ) { }

    public void paintComponent(Graphics g) {
        int width = getWidth();
        int height = getHeight();
        g.setColor(Color.blue);
        g.drawString(nume1 + " " + nume2 + " din grupa " + grupa, (getSize().width) / 2, (getSize().height - 1) / 2);
    }
}

```



```

public static void main(String args[ ]) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Introduceti numele de familie: ");
    nume1 = scanner.next();

    System.out.print("Introduceti prenumele: ");
    nume2 = scanner.next();

    System.out.print("Introduceti grupa: ");
    grupa = scanner.next();
    scanner.close();

    JFrame frame = new JFrame("Numele si grupa");
    frame.add(new JPanelClass( ));
    frame.setSize(300, 200);
    frame.setVisible(true);

}
}

```

/\*Plesa Diana lab10 grupa 2127

2. Scrie i o aplica ie de tip Frame care va desena într-un Canvas un cerc de culoare ro ie tangent la dimen  
siunea cea mai  
mică (înăl ime, lă ime) a componentei Canvas\*/

```

package lab10;

```

```

public class Test {
    public static void main(String...strings){

        FrameClass frame = new FrameClass("Painting on a Canvas");

    }
}

```

```

package lab10;

```

```

import java.awt.Canvas;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Dimension;
import java.awt.event.WindowEvent;

```

```

class MyCanvas extends Canvas{
    Graphics g;
    String output = "** *";

    MyCanvas(){
        this.setSize(new Dimension(200, 200));
        this.setBackground(new Color( 205, 255, 255));

    }
    public void paint(Graphics g) {

```

```
g.drawRect(0, 0, getSize().width - 1, getSize().height - 1);
g.setColor(new Color( 204, 0, 0));
g.fillOval(0, 0, getSize().width, getSize().height);
}
}
```

```
package lab10;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
```

```
public class FrameClass extends Frame implements WindowListener {
    MyCanvas canvas;
```

```
    FrameClass(String title){
        super(title);
```

```
        setLayout(new FlowLayout());
        canvas = new MyCanvas();
        add(canvas);
```

```
        this.addWindowListener(this);
```

```
        setSize(300, 300);
        setVisible(true);
```

```
    }

    @Override
    public void windowActivated(WindowEvent arg0) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowClosed(WindowEvent arg0) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowClosing(WindowEvent arg0) {
        System.exit(1);
    }

    @Override
    public void windowDeactivated(WindowEvent arg0) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowDeiconified(WindowEvent arg0) {
        // TODO Auto-generated method stub
    }
}
```

```
}  
@Override  
public void windowIconified(WindowEvent arg0) {  
    // TODO Auto-generated method stub  
  
}  
@Override  
public void windowOpened(WindowEvent arg0) {  
    // TODO Auto-generated method stub  
  
}  
}
```

/\*Plesa Diana lab7

\* 1. Crea i o interfa a denumitã Generator<T> cu singura metodã next(T var). Implementa i interfa a pentru a permite generarea valorilor urmãtoare în cazul aplicãrii asupra unor variabile de tipuri de date concrete (Integer, Character, etc.).  
Instan ierea clasei se va face în cadrul metodei main( ), care va fi pozi ionatã într-o clasã distinctã. \*/

```
package lab7;
```

```
public class main {
```

```
    public static void main(String[] args) {  
        Object a = 10, b = 'd', c = "Laborator";  
        Generare generare = new Generare();  
        System.out.println("Obiectul " + generare.next(a) + " face parte din " + generare.next(a).getClass());  
        System.out.println("Obiectul " + generare.next(b) + " face parte din " + generare.next(b).getClass());  
        System.out.println("Obiectul " + generare.next(c) + " face parte din " + generare.next(c).getClass());  
    }  
}
```

```
package lab7;
```

```
public class Generare implements Generator {  
    public Object next(Object var) {  
        return var;  
    }  
}
```

```
package lab7;
```

```
public interface Generator<T> {  
    T next(T var);  
}
```

/\*Plesa Diana lab7

\*2. Crea i o clasã Calculator ce are implementate metode de adunare, scãdere, înmul ire i împãr ire. Metodele vor avea ca intrare tipuri de date generice i vor returna rezultatul conform acestor date. De exemplu, suma a doi întregi va returna tot un întreg, pentru douã numere de tip float se va returna tot un float. Aplicarea metodelor de adunare i scãdere este permisã i pe variabile de tip String, restul opera iilor fiind interzise pt. acest tip de date (mesaj de eroare). \*/

```
package lab7;
```

```
public class main {
```

```
    public static void main(String[] args) {  
        Calculator object = new Calculator();  
        System.out.println("Adunare: " + object.adunare("Laborator", "Java"));  
        System.out.println("Scadere: " + object.scadere(3.5, 1.5));  
    }  
}
```

```

try {
    System.out.println("Impartire: " + object.impartire("Ana", "are"));
} catch (Exception e) {
    System.out.println(e.getMessage());
}
try {
    System.out.println("Inmultire: " + object.inmultire(6, 5));
} catch (Exception ex) {
    System.out.println(ex.getMessage());
}

}

}

```

package lab7;

```

public class Calculator implements Interface {

}

```

package lab7;

```

public interface Interface<T> {
    default T adunare(T v1, T v2) {
        if ((v1 == null) || (v2 == null)) return null;

        if ((v1 instanceof Integer) && (v2 instanceof Integer))
            return (T) new Integer(((Integer) v1) + ((Integer) v2));
        else if ((v1 instanceof Float) && (v2 instanceof Float))
            return (T) new Float(((Float) v1) + ((Float) v2));
        else if ((v1 instanceof Double) && (v2 instanceof Double))
            return (T) new Double(((Double) v1) + ((Double) v2));
        else
            return (T) new String(v1.toString() + v2.toString());
    }

    default T scadere(T v1, T v2) throws IllegalArgumentException {
        if ((v1 == null) || (v2 == null)) return null;

        if ((v1 instanceof Integer) && (v2 instanceof Integer))
            return (T) new Integer(((Integer) v1) - ((Integer) v2));
        else if ((v1 instanceof Float) && (v2 instanceof Float))
            return (T) new Float(((Float) v1) - ((Float) v2));
        else if ((v1 instanceof Double) && (v2 instanceof Double))
            return (T) new Double(((Double) v1) - ((Double) v2));
        else {
            if (v1.toString().contains(v2.toString())) {
                return (T) new String(v1.toString().replaceAll(v2.toString(), ""));
            } else {
                throw new IllegalArgumentException("ERROR: String 1 nu contine string 2!");
            }
        }
    }
}

```

```
}  
}  
}
```

```
default T inmultire(T v1, T v2) throws IllegalArgumentException {  
    if ((v1 == null) || (v2 == null)) return null;  
  
    if ((v1 instanceof Integer) && (v2 instanceof Integer))  
        return (T) new Integer(((Integer) v1) * ((Integer) v2));  
    else if ((v1 instanceof Float) && (v2 instanceof Float))  
        return (T) new Float(((Float) v1) * ((Float) v2));  
    else if ((v1 instanceof Double) && (v2 instanceof Double))  
        return (T) new Double(((Double) v1) * ((Double) v2));  
    else {  
        throw new IllegalArgumentException("EROARE: Strings nu pot fi inmultite!");  
    }  
}
```

```
default T impartire(T v1, T v2) throws IllegalArgumentException {  
    if ((v1 == null) || (v2 == null)) return null;  
  
    if ((v1 instanceof Integer) && (v2 instanceof Integer))  
        return (T) new Integer(((Integer) v1) / ((Integer) v2).intValue());  
    else if ((v1 instanceof Float) && (v2 instanceof Float))  
        return (T) new Float(((Float) v1) / ((Float) v2).floatValue());  
    else if ((v1 instanceof Double) && (v2 instanceof Double))  
        return (T) new Double(((Double) v1) / ((Double) v2).doubleValue());  
    else {  
        throw new IllegalArgumentException("EROARE: Strings nu pot fi impartite!");  
    }  
}  
}
```

/\*Plesa Diana lab7

3. Implementa i o aplica ie în cadrul căreia să ave i o clasă SetterGetter generică ce permite setarea i retur narea valorilor atributelor pentru mai multe tipuri de obiecte. De exemplu, având clasele Copil, Adult i Pensionar, să se poată seta i returna numele i vârsta acestora. Crea i colec ii cu intrări unice de obiecte de tip Copil, Adult, Pensio nar pe care să le popula i cu date citite din consolă. Afi a i datele preluate în diferite moduri.  
\*/

package lab7;

```
import java.util.HashSet;  
import java.util.Scanner;  
import java.util.Set;
```

```
public class main {  
    private static Scanner scanner = new Scanner(System.in);
```

```
    public static void main(String[] args) {  
        Set<Copil> copii = addCopil();  
        Set<Adult> adulti=addAdulti();
```

```

Set<Pensionar> pensionar= addPensionar();

System.out.println("Datele introduse sunt: ");
show(copii);
show(adulti);
show(pensionar);
}

private static Set<Copil> addCopil() {
try {
    System.out.print("Introduceti numarul de copii: ");
    int number = scanner.nextInt();
    if (number == 0) {
        return null;
    }

    SetterGetter object = new SetterGetter();
    Set<Copil> copii = new HashSet<>();

    System.out.println("Introduceti numele si varsta: ");
    String nume;
    int varsta;
    for (int i = 0; i < number; i++) {
        System.out.print("\nCopil[" + (i + 1) + "]\nNume: ");
        nume = scanner.next();
        System.out.print("Varsta: ");
        varsta = scanner.nextInt();

        Copil aux = new Copil();
        object.setVarsta(aux, varsta);
        object.setNume(aux, nume);
        copii.add(aux);
    }
    return copii;

} catch (Exception e) {
    e.printStackTrace();
}
return null;
}

private static Set<Adult> addAdulti() {
try {
    System.out.print("\nIntroduceti numarul de adulti: ");
    int numar = scanner.nextInt();
    if (numar == 0) {
        return null;
    }

    SetterGetter object = new SetterGetter();
    Set<Adult> adulti = new HashSet<>();

    System.out.println("Introduceti numele si varsta: ");
    String nume;
    int varsta;

```

```

for (int i = 0; i < numar; i++) {
    System.out.print("\nAdult[" + (i + 1) + "]\nNume: ");
    nume = scanner.next();
    System.out.print("Varsta: ");
    varsta = scanner.nextInt();

    Adult aux = new Adult();
    object.setVarsta(aux, varsta);
    object.setNume(aux, nume);
    adulti.add(aux);
}
return adulti;

} catch (Exception e) {
    e.printStackTrace();
}
return null;
}

private static Set<Pensionar> addPensionar() {
    try {
        System.out.print("\nIntroduceti numarul de pensionari: ");
        int numar = scanner.nextInt();
        if (numar == 0) {
            return null;
        }

        SetterGetter object = new SetterGetter();
        Set<Pensionar> pensionar = new HashSet<>();

        System.out.println("Introduceti numele si varsta: ");
        String nume;
        int varsta;
        for (int i = 0; i < numar; i++) {
            System.out.print("\nPensionar[" + (i + 1) + "]\nNume: ");
            nume = scanner.next();
            System.out.print("Varsta: ");
            varsta = scanner.nextInt();

            Pensionar aux = new Pensionar();
            object.setVarsta(aux, varsta);
            object.setNume(aux, nume);
            pensionar.add(aux);
        }
        return pensionar;

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

private static void show(Set<? extends Persoane> set) {

    if (set instanceof Copil) System.out.println("Copii care sunt in baza de date:"); //hashset

```



```
else if (set instanceof Adult) System.out.println("Adulti care sunt in baza de date:");
else if (set instanceof Pensionar) System.out.println("Pensionarii care sunt in baza de date:");
```

```
try {
    if (set.size() == 0)
        throw new Exception("NULL");
    SetterGetter object = new SetterGetter();

    for (Persoane instanceOfSet : set) {
        System.out.println("\nNume: " + object.getNuma(instanceOfSet));
        System.out.println("Varsta: " + object.getVarsta(instanceOfSet));
    }

} catch (Exception ex) {
    System.out.println("\nEROARE!");
}
}
```

```
package lab7;
```

```
public class SetterGetter<T extends Persoane> {
```

```
    void setNume(T object, String nume) {
        object.numa = nume;
    }
```

```
    String getNume(T object) {
        return object.numa;
    }
```

```
    void setVarsta(T object, int varsta) {
        object.varsta = varsta;
    }
```

```
    int getVarsta(T object) {
        return object.varsta;
    }
```

```
}
```

```
package lab7;
```

```
public class Persoane {
    protected String nume;
    protected int varsta;
```

```
    public Persoane() {}
```

```
    public Persoane(String nume, int varsta) {
        this.nume = nume;
        this.varsta = varsta;
    }
}
```

```
package lab7;

public class Pensionar extends Persoane {

    public Pensionar() {}

    public Pensionar(String nume, int varsta) {
        super(nume, varsta);
    }

}
```

```
package lab7;

public class Copil extends Persoane {
    public Copil() {}

    public Copil(String nume, int varsta) {
        super(nume, varsta);
    }
}
```

```
package lab7;

public class Adult extends Persoane {

    public Adult() {}

    public Adult(String nume, int varsta) {
        super(nume, varsta);
    }

}
```

/\*Plesa Diana lab7

5. Scrieti o clasa Student cu campurile private nume, grupa, media, si metode getter/setter pentru campuri. In main(), plasata in alta clasa, creati o colectie de tip SortedSet, cu obiecte de tip Student, initializate cu valori citite de la tastatura, care sa pastreze elementele descrescator dupa medie si crescator dupa nume (cei care au aceeasi medie sa fie in ordine alfabetica). Parcurgeti cu for-loop colectia si afisati toate elementele. Parcurgeti apoi colectia cu un iterator si afisati toti studentii cu media >=8. Parcurgeti colectia cu forEach() si afisati toate datele studentilor dintr-o anumita grupa.

\*/

```
package lab7;

import java.util.HashSet;
import java.util.Iterator;
import java.util.Scanner;
import java.util.Set;
```

```

public class main {
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        Set<Student> studenti = addStudenti();
        show(studenti);
    }

    private static Set<Student> addStudenti() {
        try {
            System.out.print("Introduceti numarul de studenti: ");
            int numar = scanner.nextInt();
            if (numar == 0)
                return null;

            Set<Student> stud = new HashSet<>();
            System.out.println("Introduceti numele, grupa, media: ");
            String nume;
            int grupa;
            double media;

            for (int i = 0; i < numar; i++) {
                System.out.print("\nStudent[" + (i + 1) + "]\nNume: ");
                nume = scanner.next();
                System.out.print("Grupa: ");
                grupa = scanner.nextInt();
                System.out.print("Media: ");
                media = scanner.nextDouble();

                Student aux = new Student();
                aux.setNume(nume);
                aux.setGrupa(grupa);
                aux.setMedia(media);
                stud.add(aux);
            }
            return stud;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    private static void show(Set<Student> set) {
        try {
            if (set.size() == 0)
                throw new Exception("Niciun studenti!");

            for (Student i : set) {
                System.out.println("\nNume: " + i.getNume());
                System.out.println("Grupa: " + i.getGrupa());
                System.out.println("Media: " + i.getMedia());
            }
        }
    }
}

```

```

Iterator<Student> itr = set.iterator();
System.out.println("\nStudenti cu media >=8 ");
int ap = 0;
while (itr.hasNext()) {
    Student x = itr.next();

    if (x.getMedia() >= 8) {
        System.out.print(x.getNume()+" ");
        ap++;
    }
}
if (ap ==0)
    System.out.print("\nNu sunt studenti cu media >=8 ");
}
catch (Exception e) {
    e.printStackTrace();
}
}
}

```

package lab7;

```

public class Student implements Comparable<Student> {
    private String nume;
    private int grupa;
    private double media;

    public Student() {}

    public Student(String nume, int grupa, double media) {
        this.nume = nume;
        this.grupa = grupa;
        this.media = media;
    }

    public String getNume() {
        return nume;
    }

    public void setNume(String nume) {
        this.nume = nume;
    }

    public int getGrupa() {
        return grupa;
    }

    public void setGrupa(int grupa) {
        this.grupa = grupa;
    }

    public double getMedia() {
        return media;
    }
}

```

```
}
```

```
public void setMedia(double media) {  
    this.media = media;  
}
```

```
public int compareTo(Student o) {  
    if (this.getMedia() != o.getMedia()) {  
        return (int) (o.getMedia() - this.getMedia());  
    } else {  
        return o.getNume().compareTo(this.getNume());  
    }  
}
```

```
}
```

/\*Plesa Diana lab12 grupa 2127

2. Crea i o aplica ie Java ce include 3 elemente de tip slider, prin intermediul cărora se poate seta culoare a unui pătrat

de dimensiunea 100x100. Pătratul va fi desenat într-o componentă specifică (Canvas, etc.).\*/

```
package lab12;
```

```
public class main {  
    public static void main(String[] args) {  
        new Slider();  
    }  
}
```

```
package lab12;
```

```
import java.awt.Color;  
import java.awt.Dimension;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.JSlider;  
import javax.swing.event.ChangeEvent;  
import javax.swing.event.ChangeListener;
```

```
public class Slider extends JFrame {
```

```
    private static final long serialVersionUID = 1L;
```

```
    JSlider slider1;  
    JSlider slider2;  
    JSlider slider3;  
    int rosu = 0;  
    int verde = 0;  
    int albastru = 0;  
    private JPanel panel;
```

```
    public static void main(String[] args) {  
        new Slider();  
    }
```

```
    public Slider() {  
        setLayout(null);  
        setSize(400, 350);  
        setTitle("Culoare patrat");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setVisible(true);
```

```
        panel = new JPanel();  
        panel.setBounds(100, 50, 100, 100);  
        panel.setBackground(new Color(rosu, verde, albastru));  
        add(panel);
```

```
        slider1 = new JSlider(JSlider.HORIZONTAL, 0, 255, 1);  
        slider1.setBounds(20, 40, 50, 20);  
        slider1.setPreferredSize(new Dimension(50, 50));  
        slider1.setMajorTickSpacing(255);
```

```
slider1.addChangeListener(new ChangeListener() {  
    public void stateChanged(ChangeEvent event) {  
        rosu = slider1.getValue();  
        panel.setBackground(new Color(rosu, verde, albastru));  
    }  
});
```

```
add(slider1);
```

```
slider2 = new JSlider(JSlider.HORIZONTAL, 0, 255, 1);  
slider2.setBounds(20, 90, 50, 20);  
slider2.setMajorTickSpacing(255);  
slider2.setPreferredSize(new Dimension(50, 50));  
slider2.addChangeListener(new ChangeListener() {  
    public void stateChanged(ChangeEvent event) {  
        verde = slider2.getValue();  
        panel.setBackground(new Color(rosu, verde, albastru));  
    }  
});  
add(slider2);
```

```
slider3 = new JSlider(JSlider.HORIZONTAL, 0, 255, 1);  
slider3.setBounds(20, 140, 50, 20);  
slider3.setPreferredSize(new Dimension(50, 50));  
slider3.setMajorTickSpacing(255);  
slider3.addChangeListener(new ChangeListener() {  
    public void stateChanged(ChangeEvent event) {  
        albastru = slider3.getValue();  
        panel.setBackground(new Color(rosu, verde, albastru));  
    }  
});  
add(slider3);
```

```
}  
}
```

/\*Plesa Diana lab\_2

4. Scrieti o aplicatie Java in care cititi de la tastatura un numar intreg si afisati toti divizorii numarului citi.\*/

```
package lab2;
```

```
import java.util.Scanner;
```

```
public class lab2 {
```

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

```
        int nr, i;
```

```
        System.out.print("Introduceti un numar: ");
```

```
        nr = new Scanner(System.in).nextInt();
```

```
        System.out.print("Divizorii sunt: \n");
```

```
        for(i=1;i<=nr;i++)
```

```
        {
```

```
            if(nr%i==0)
```

```
                System.out.println(+ i);
```

```
        }
```

```
    }
```

```
}
```



/\*Plesa Diana lab\_2

2. Scrieti o aplicatie Java in care cititi de la tastatura o valoare intreaga si se afiseaza factorialul numarului citit.\*/

```
package lab2;
```

```
import java.util.Scanner;
```

```
public class lab2 {
```

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

```
        int n, f=1, i=1;
```

```
        System.out.print("Introduceti un numar: ");
```

```
        n=new Scanner(System.in).nextInt();
```

```
        while(i < n)
```

```
        {
```

```
            f=f*(i+1);
```

```
            i++;
```

```
        }
```

```
        System.out.println("Factorialul numarului este = " + f);
```

```
    }
```

```
}
```

/\*Plesa Diana lab\_2

3. Scrieti o aplicatie Java in care se citesc de la tastatura 2 valori intregi si se afiseaza cmmddc-ul valorilor  
.\*/

```
package lab2;
```

```
import java.util.Scanner;
```

```
public class lab2 {
```

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

```
        int a, b;
```

```
        System.out.print("Introduceti primul numar: ");
```

```
        a=new Scanner(System.in).nextInt();
```

```
        System.out.print("Introduceti al doilea numar: ");
```

```
        b=new Scanner(System.in).nextInt();
```

```
        while(a != b)
```

```
        {
```

```
            if(a > b)
```

```
                a = a - b;
```

```
            else
```

```
                b = b - a;
```

```
        }
```

```
        System.out.println("cmmddc = " + a);
```

```
    }
```

```
}
```

/\*Plesa Diana lab\_2

5. Scrieti o aplicatie Java in care cititi de la tastatura mai multe numere intregi si verificati daca sunt numere prime. Numarul de valori citite este preluat de la tastatura sau din linia de comanda.\*/

```
package lab2;
```

```
import java.util.Scanner;
```

```
public class lab2 {
```

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

```
        int nr, j, n, prim, d;
```

```
        System.out.print("Introduceti cate numare se citesc: ");
```

```
        n = new Scanner(System.in).nextInt();
```

```
        for(j=1;j<=n;j++)
```

```
        {
```

```
            System.out.print("Introduceti numarul " + j);
```

```
            nr = new Scanner(System.in).nextInt();
```

```
            prim=1;
```

```
            d=2;
```

```
            while (d<=nr/2)
```

```
            {
```

```
                if(nr%d==0) prim=0;
```

```
                d++;
```

```
            }
```

```
            if(prim==1) System.out.println(nr + " Este numar prim.");
```

```
            else
```

```
                System.out.println(nr + " Nu este numar prim.");
```

```
        }
```

```
    }
```

```
}
```