

/*Andrea Guarino 4BI

Creare una classe che vada a gestire un set di numeri interi appartenenti ad un range prefissato.

Ogni gestore di numeri interi istanziato conoscerà il valore minimo, il valore massimo (in riferimento al range da gestire) e se deve accettare numeri pari o numeri dispari.

La classe dovrà presentare i seguenti metodi:

Costruttori:

Default: istanzia un oggetto che accetta numeri pari compresi tra 0 e 200

Con parametri

Di copia

Getters e Setters

Un metodo per aggiungere un numero al set

Un metodo che restituisce una stringa csv di tutti i numeri del set

Un metodo che restituisce la media dei numeri del set

Un metodo che restituisce, in base ad un valore passato come parametro, il numero massimo o il numero minimo, tra i numeri del set

Crea un programma di prova che verifichi il funzionamento della classe.

*/

```
public class Main {
    public static void main(String[] args) {
        GestoreNumeri gestoreNumeri1 = new GestoreNumeri(0,300,true);
        char sc;
        do {
            menu();
            sc = Input.leggiChar('a','e',"Scelta: ");
            switch (sc) {
                case 'a': {
                    aggiungiNumero(gestoreNumeri1);
                    break;
                }
                case 'b': {
                    stampaNumeri(gestoreNumeri1);
                    break;
                }
                case 'c': {
                    mediaNumeri(gestoreNumeri1);
                    break;
                }
                case 'd': {
                    numeroMaxMin(gestoreNumeri1);
                    break;
                }
            }
        }while(sc!='e');
    }

    private static void numeroMaxMin(GestoreNumeri gestoreNumeri1) {
```

```

        int scelta, number;
        System.out.println("Cosa vuoi stampare :");
        System.out.println("1) numero massimo");
        System.out.println("2) numero minimo");
        scelta = Input.leggiInt(1,2,"Scelta :");
        number = gestoreNumeri1.findMAXorMin(scelta);
        if(scelta==1){
            System.out.println("Il numero massimo è: " + number);
        }else
            System.out.println("Il numero minimo è: " + number);
    }

    private static void aggiungiNumero(GestoreNumeri gestoreNumeri1) {
        int num,max;
        max= gestoreNumeri1.getVmax();
        num = Input.leggiInt(0,max,"Inserire numero da posizionare nella
sequenza di numeri");
        if(!gestoreNumeri1.addNumber(num)){
            System.out.println("Numero non divisibile per due");
        }else
            System.out.println("Numero inserito con successo");
    }

    private static void stampaNumeri(GestoreNumeri gestoreNumeri1) {
        String s = gestoreNumeri1.csvStringNumbers();
        System.out.println(s);
    }

    private static void mediaNumeri(GestoreNumeri gestoreNumeri1){
        double media = gestoreNumeri1.mediaNumbers();
        if(media== -1){
            System.out.println("Impossibile ottenere la media");
        }else{
            System.out.println("La media di tutti i numeri inseriti è: " +
media);
        }
    }

    }

    public static void menu(){
        System.out.println("a) Aggiungi numero alla sequenza");
        System.out.println("b) Stampa riga con tutti i numeri");
        System.out.println("c) Stampa media della sequenza");
        System.out.println("d) Valore massimo o minimo della sequenza");
        System.out.println("e) Termina il programma");
    }
}

////////////////////////////////////
public class GestoreNumeri {
    private final int NMAX = 100;
    private int vmin;
    private int vmax;
    private int numeri[];
    private boolean pari;

```

```
private int lastNumero;
```

```
//METODI
```

```
public int findMAXorMin(int num){
    int number= this.numeri[0];
    if(num==1){
        for(int i=0;i<this.lastNumero;i++){
            if(this.numeri[i]>number)
                number=this.numeri[i];
        }
    }
    if(num==2){
        for(int i=0;i<this.lastNumero;i++){
            if(this.numeri[i]<number)
                number=this.numeri[i];
        }
    }
    if(num!=1 && num!=2)
        number=-1;
    return number;
}
```

```
public double mediaNumbers(){
    int somma=0;
    double media=-1;
    for(int i=0;i<this.lastNumero;i++){
        somma+=this.numeri[i];
    }
    media=(double)somma/this.lastNumero;
    return media;
}
```

```
public String csvStringNumbers(){
    String csvString = "";
    for(int i=0;i<this.lastNumero;i++){
        csvString += this.numeri[i] + ";";
    }
    return csvString;
}
```

```
public boolean addNumber(int num){
    boolean check=false;
    if(this.pari==true && num%2==0 && num>=this.vmin && num<=this.vmax){
        this.numeri[this.lastNumero]=num;
        check=true;
        this.lastNumero++;
    }
    return check;
}
```

```

//COSTRUTTORI

public GestoreNumeri(){
    setVmin(0);
    setVmax(200);
    setPari(true);
    this.numeri = new int[NMAX];
}

public GestoreNumeri(int vmin, int vmax,boolean pari){
    setVmin(vmin);
    setVmax(vmax);
    setPari(pari);
    this.numeri = new int[NMAX];
}

public void GestoreNumeri(GestoreNumeri a){
    setVmin(a.vmin);
    setVmax(a.vmax);
    setPari(a.pari);
    this.numeri = new int[NMAX];
}

//SETTER

public void setVmin(int vmin){
    if(vmin>this.vmax)
        this.vmin = 0;
    else
        this.vmin = vmin;
}

public void setVmax(int vmax) {
    if(vmax < this.vmin)
        this.vmax = 200;
    else
        this.vmax = vmax;
}

public void setPari(boolean pari) {
    this.pari=pari;
}

//GETTER

public int getVmax() {
    return vmax;
}

public int getVmin() {
    return vmin;
}

public boolean getPari(){
    return pari;
}

```

```

}
////////////////////////////////////
import java.util.InputMismatchException;
import java.util.Scanner;
public class Input {
    public static int leggiInt (int vmin, int vmax,String msg){
        int n=0;
        boolean err;
        do{
            System.out.println(msg);
            Scanner input = new Scanner(System.in);
            err = false;
            try {
                n = input.nextInt();
                input.nextLine();
            }catch (InputMismatchException e){
                System.out.println("Ciò che hai inserito non è un numero");
                err = true;
            }
            if(n<vmin || n>vmax)
                System.out.println("Errore");
        }while((n<vmin || n>vmax) || err);
        return n;
    }

    public static char leggiChar (char vmin, char vmax,String msg){
        char c='.';
        boolean err;
        do{
            System.out.println(msg);
            Scanner input = new Scanner(System.in);
            err = false;
            try {
                c = input.nextLine().charAt(0);
                input.nextLine();
            }catch (InputMismatchException e){
                System.out.println("Ciò che hai inserito non è un carattere");
                err = true;
            }
            if(c<vmin || c>vmax)
                System.out.println("Errore");
        }while((c<vmin || c>vmax) || err);
        return c;
    }

    public static float leggiFloat (float vmin, float vmax,String msg){
        float n=0;
        boolean err;
        do{
            System.out.println(msg);
            Scanner input = new Scanner(System.in);
            err = false;
            try {
                n = input.nextFloat();
            }

```

```

        input.nextLine();
    }catch (InputMismatchException e){
        System.out.println("Ciò che hai inserito non è un numero
decimale");
        err = true;
    }
    if(n<vmin || n>vmax)
        System.out.println("Errore");
    }while((n<vmin || n>vmax) || err);
    return n;
}

public static Double leggiDouble (double vmin, double vmax,String msg){
    double n=0;
    boolean err;
    do{
        System.out.println(msg);
        Scanner input = new Scanner(System.in);
        err = false;
        try {
            n = input.nextFloat();
            input.nextLine();
        }catch (InputMismatchException e){
            System.out.println("Ciò che hai inserito non è un numero
decimale");
            err = true;
        }
        if(n<vmin || n>vmax)
            System.out.println("Errore");
    }while((n<vmin || n>vmax) || err);
    return n;
}

public static String leggiStr(String msg) {
    String s="";
    boolean err;
    do{
        System.out.println(msg);
        Scanner input = new Scanner(System.in);
        err = false;
        try {
            s = input.nextLine();
            input.nextLine();
        }catch (InputMismatchException e){
            System.out.println("Ciò che hai inserito non è una Stringa");
            err = true;
        }
        if(s.trim().isEmpty())
            System.out.println("Errore");
    }while(s.trim().isEmpty() || err);
    return s;
}

}

```