

## Polinomio di secondo grado:

```
import GraphicIO.*;

public class Polinomio2
{
    public static void main (String args[])
    {
        GraphicConsole.showWindow();

        GraphicConsole.println ("Inserisci i coefficienti interi del polinomio di
secondo grado");

        int a = GraphicConsole.readInt("a =");
        int b = GraphicConsole.readInt("b =");
        int c = GraphicConsole.readInt("c =");

        GraphicConsole.println ();
        GraphicConsole.println ("p(x) = "+a+"x^2 + "+b+"x + "+c);

        GraphicConsole.println ();
        double x = GraphicConsole.readDouble("x =");

        double valore = a*x*x + b*x + c;

        GraphicConsole.println ("p("+x+") = "+valore);
    }
}
```

## Calcolo delle soluzioni di una equazione di secondo grado:

```
import GraphicIO.*;

class Equazione{

    public static void main (String args[]){

        GraphicConsole.showWindow();

        int a=GraphicConsole.readInt("Inserire il valore di a");
        int b=GraphicConsole.readInt("Inserire il valore di b");
        int c=GraphicConsole.readInt("Inserire il valore di c");

        int delta=b*b-a*a*c;

        GraphicConsole.println("Soluzioni dell'equazione "+a+"X^2"+b+"X"+c+"=0");

        if (delta<0)
            GraphicConsole.println("Non esistono soluzioni reali.");
        else if (delta==0){
            double x=-b/(2.0*a);
            GraphicConsole.println("Soluzioni coincidenti x1=x2="+x);
        }
        else{
            double x1=(-b+Math.sqrt(delta))/(2*a);
            double x2=(-b-Math.sqrt(delta))/(2*a);
            GraphicConsole.println("Soluzioni reali e distinte x1="+x1+"      "x2="+x2);
        }
    }
}
```

## Classificazione di un triangolo date le misure dei suoi lati (1):

```
/**
 * Legge tre numeri interi
 * che rappresentano i lati di un triangolo
 * e stampa un messaggio che dice se il triangolo è
 * equilatero, isoscele oppure scaleno.
 *
 * Versione con if innestati.
 */

import GraphicIO.*;

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

        GraphicConsole.showWindow(); //finestra principale dell'I/O grafico

        // legge il valore dei tre lati

        System.out.println("Scrivi la lunghezza dei lati di un triangolo ");
        int a = GraphicConsole.readInt("a=");
        int b = GraphicConsole.readInt("b=");
        int c = GraphicConsole.readInt("c=");

        GraphicConsole.println();
        GraphicConsole.print("Il triangolo è ");

        // confronta i lati e stampa il messaggio
        if (a==b) // equilatero oppure isoscele
            if (b==c) // equilatero
                GraphicConsole.println("equilatero");
            else // isoscele
                GraphicConsole.println("isoscele");
        else // isoscele oppure scaleno
            if (b==c) // isoscele
                GraphicConsole.println("isoscele");
            else // isoscele oppure scaleno
                if (a==c) // isoscele
                    GraphicConsole.println("isoscele");
                else // scaleno
                    GraphicConsole.println("scaleno");
    }
}
```

## Classificazione di un triangolo date le misure dei suoi lati (2):

```
/**
 * Legge tre numeri interi
 * che rappresentano i lati di un triangolo
 * e stampa un messaggio che dice se il triangolo è
 * equilatero, isoscele oppure scaleno.
 *
 * Versione con contatore dei lati uguali.
 */

import GraphicIO.*;

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

```

int uguali;          // numero di coppie di lati uguali

// legge il valore dei tre lati
GraphicConsole.println("Scrivi la lunghezza dei lati di un triangolo ");
int a = GraphicConsole.readInt("a=");
int b = GraphicConsole.readInt("b=");
int c = GraphicConsole.readInt("b=");

GraphicConsole.print("Il triangolo è ");

// calcola il numero di coppie di lati uguali
uguali = 0;
// confronta i lati
if (a==b)
    uguali++;
if (a==c)
    uguali++;
if (b==c)
    uguali++;

// stampa il messaggio
if (uguali==0)
    GraphicConsole.println("scaleno");
else if (uguali==1)
    GraphicConsole.println("isoscele");
else // uguali vale sicuramente 3
    GraphicConsole.println("equilatero");
}
}

```

### **Classificazione di un triangolo date le misure dei suoi lati (3):**

```

/**
 * Legge tre numeri interi
 * che rappresentano i lati di un triangolo
 * e stampa un messaggio che dice se il triangolo è
 * equilatero, isoscele oppure scaleno.
 *
 * Versione con condizioni composte.
 */

import GraphicIO.*;

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

        /* legge il valore dei tre lati */
        GraphicConsole.println("Scrivi la lunghezza dei lati di un triangolo ");
        int a = GraphicConsole.readInt("a=");
        int b = GraphicConsole.readInt("b=");
        int c = GraphicConsole.readInt("c=");

        GraphicConsole.print("Il triangolo è ");

        /* calcola il tipo di triangolo,
         * confrontando i lati, e stampa il messaggio */
        if (a==b && a==c)
            GraphicConsole.println("equilatero");
        else if (a==b || a==c || b==c)
            GraphicConsole.println("isoscele");
        else

```

```

        GraphicConsole.println("scaleno");
    }
}

```

### **Calcolo del fattoriale (con verifica dell'overflow):**

```

import GraphicIO.*;

public class Fattoriale
{
    public static void main (String args[])
    {
        GraphicConsole.showWindow();
        int n = GraphicConsole.readInt("n=");
        if (n < 0)
            GraphicConsole.println ("Impossibile calcolare il fattoriale di un numero
negativo");
        else
        {
            int fattoriale = 1;

            boolean ok = true;

            for (int i = 2; i <= n && ok; i++)
            {
                if (Integer.MAX_VALUE/fattoriale < i)
                {
                    GraphicConsole.println ("Il fattoriale è troppo grande");
                    ok = false;
                }
                fattoriale *= i; // fattoriale = fattoriale * i
            }
            if (ok)
                GraphicConsole.println (n+"! = "+fattoriale);
        }
    }
}

```

### **Giorni trascorsi dall'inizio dell'anno:**

```

import GraphicIO.*;

public class GiorniTrascorsi
{
    public static void main (String args[])
    {
        GraphicConsole.showWindow();

        int giorno = GraphicConsole.readInt ("giorno: ");
        int mese = GraphicConsole.readInt ("mese: ");
        int anno = GraphicConsole.readInt ("anno: ");

        int giorniTrascorsi = 0;

        for (int m = 1; m < mese; m++)
        {
            switch (m)
            {
                case 4:
                case 6:
                case 9:

```

```

        case 11:
            giorniTrascorsi += 30;
            break;
        case 2:
            if ( (anno % 4) == 0)
                giorniTrascorsi += 29;
            else
                giorniTrascorsi += 28;
            break;
        default:
            giorniTrascorsi += 31;
    }
}
giorniTrascorsi += giorno;

GraphicConsole.println("Il "+giorno+"/"+mese+"/"+anno+" è il "+
                        giorniTrascorsi+"° giorno dell'anno "+anno);
}
}

```

### Calcolo della media dei voti (1):

```

import GraphicIO.*;

public class MediaEsami
{
    public static void main (String args[])
    {
        GraphicConsole.showWindow();

        int numeroEsami = GraphicConsole.readInt ("Numero esami: ");

        int totale = 0;

        for (int i = 1; i <= numeroEsami; i++)
            totale += GraphicConsole.readInt ("Voto esame n. "+i+": ");

        float media = (float)totale/numeroEsami;

        GraphicConsole.println ("media = "+media);
    }
}

```

### Calcolo della media dei voti (2):

```

/* fa uso di un vettore in cui memorizza i voti */

import GraphicIO.*;

public class MediaEsami2
{
    public static void main (String args[])
    {
        GraphicConsole.showWindow();

        int numeroEsami = GraphicConsole.readInt ("Numero esami: ");

        int esami[] = new int[numeroEsami];

        for (int i = 0; i < esami.length; i++)

```

```

        esami[i] = GraphicConsole.readInt ("Voto esame n. "+(i+1)+" ": );

int totale = 0;

for (int i = 0; i < esami.length; i++)
    totale += esami[i];

float media = (float)totale/numeroEsami;

GraphicConsole.println ("media = "+media);
}
}

```

### Calcolo del massimo, del sub-massimo e del minimo di un vettore:

```

import GraphicIO.*;

class MassimiMinimi{

    public static void main(String args[]){

        GraphicConsole.showWindow();

        //dichiarazione del vettore
        int v[];

        int n=GraphicConsole.readInt("Numero di elementi del vettore:");

        //allocazione del vettore di n elementi
        v=new int[n];

        //ciclo di lettura da input del vettore
        for(int i=0;i<n;i++){
            v[i]=GraphicConsole.readInt("Inserire elemento "+i);
        }

        //calcolo di massimi e minimi
        int massimo=v[0];
        int s_massimo=Integer.MIN_VALUE;
        int minimo=v[0];
        for(int i=1;i<n;i++){
            if (v[i]>massimo){//occorre aggiornare massimo e sub-massimo
                s_massimo=massimo;
                massimo=v[i];
            }
            else if (v[i]>s_massimo && v[i]<massimo)//occorre aggiornare solo il sub-
massimo
                s_massimo=v[i];
            if (v[i]<minimo)
                minimo=v[i];
        }//for

        GraphicConsole.println("Il massimo è: "+massimo);

        if(s_massimo==Integer.MIN_VALUE)
            GraphicConsole.println("Il sub-massimo non è definito per questo
vettore");
        else
            GraphicConsole.println("Il sub-massimo è: "+s_massimo);
            GraphicConsole.println("Il minimo è: "+minimo);
        }//main
    }//class

```



```

public static void main(String args[]){

    GraphicConsole.showWindow();

    //dichiarazione dei vettori
    int v1[],v2[],somma[];

    int n=GraphicConsole.readInt("Numero di elementi di entrambi i vettori:");

    //allocazione dei vettori di n elementi ciascuno
    v1=new int[n];
    v2=new int[n];
    somma=new int[n];

    //ciclo di lettura da input del 1° vettore
    for(int i=0;i<n;i++)
        v1[i]=GraphicConsole.readInt("Inserire elemento "+i+" di V1");

    //ciclo di lettura da input del 2° vettore
    for(int i=0;i<n;i++)
        v2[i]=GraphicConsole.readInt("Inserire elemento "+i+" di V2");

    //calcolo della somma
    GraphicConsole.print("Il vettore somma V1+V2 è:");
    for(int i=0;i<n;i++){
        somma[i]=v1[i]+v2[i];
        GraphicConsole.print(somma[i]+" ");
    }//for

    GraphicConsole.println("");

    //calcolo del prodotto scalare
    int ps=0;
    for(int i=0;i<n;i++)
        ps+=v1[i]*v2[i];
    GraphicConsole.println("Il prodotto scalare di V1 e V2 è: "+ps);
} //main

} //class

```

### **Operazioni di base su matrici:**

```

import GraphicIO.*;

class Matrici {

    /* lettura di una matrice r x c*/
    public static int[][] letturaMatrice(int r, int c){
        int [][] matrice=new int [r][c];
        for (int i=0;i<r;i++)
            for(int j=0;j<c;j++)
                matrice[i][j]=GraphicConsole.readInt("Elemento (" +i+" ,"+j+" ) ");
        return matrice;
    }

    /* stampa di una matrice di quaqlsiasi dimensioni*/
    public static void stampaMatrice(int [][] matrice){
        for (int i=0;i<matrice.length;i++){
            for(int j=0;j<matrice[0].length;j++)
                GraphicConsole.print(matrice[i][j]+" ");
            GraphicConsole.println("");
        }
    }
}

```

```

    }
}

/* Calcola la somma degli elementi dell'array di array a. */
public static int sommaDiretta(int[][] a) {
    int s=0; // somma degli elementi di a
    for (int i=0; i<a.length; i++)
        for (int j=0; j<a[0].length; j++)
            s += a[i][j];
    return s;
}

/* Calcola la somma degli elementi dell'array a. */
public static int somma(int[] a) {
    int s=0; // somma degli elementi di a
    for (int i=0; i<a.length; i++)
        s += a[i];
    return s;
}

/* Calcola la somma degli elementi dell'array di array a. */
public static int somma(int[][] a) {
    int s=0; // somma degli elementi di a
    for (int i=0; i<a.length; i++)
        s += somma(a[i]);
    return s;
}

/* Calcola la somma di due matrici di dimensioni identiche*/
public static int[][] sommaMatrici(int[][] matrice1,int[][] matrice2){
    int n=matrice1.length;
    int m=matrice1[0].length;
    int[][] risultato=new int[n][m];
    for (int i=0;i<n;i++)
        for (int j=0;j<m;j++)
            risultato[i][j]=matrice1[i][j]+matrice2[i][j];
    return risultato;
}

public static void main(String[] args) {
    GraphicConsole.showWindow();
    int [][] m1=letturaMatrice(2,2);
    GraphicConsole.println("Somma diretta degli elementi della prima matrice
"+sommaDiretta(m1));
    GraphicConsole.println("Somma per righe degli elementi della prima matrice
"+somma(m1));
    GraphicConsole.println("");

    int [][] m2=letturaMatrice(2,2);

    GraphicConsole.println("Matrice 1");
    stampaMatrice(m1);

    GraphicConsole.println("Matrice 2");
    stampaMatrice(m2);

    int [][] somma=sommaMatrici(m1,m2);

    GraphicConsole.println("Matrice Somma");
    stampaMatrice(somma);
} //main

} //class

```

## Esempio di risoluzione di una traccia d'esame :

### Esercizio 1

Si consideri il seguente metodo:

```
public static int esercizio(int[] v){
    for( int i=1; i<v.length; i++ )
        if (v[i] < 0)
            return -1;
    for( int i=1; i<v.length; i++ )
    {
        int x=1;
        for(int j=i-1; j>=0; j-- )
            x *= v[j];
        if (x == v[i])
            return 1;
    }
    return 0;
}
```

Si descriva sinteticamente l'algoritmo che esso implementa e, in particolare, si specifichi il valore restituito dal metodo nel caso il parametro  $v$  valga: [1, 2, 1, 7, 14, 26, 48].

### Esercizio 2

Si progetti una classe *Esercizio2* per elaborare matrici di interi. In particolare bisogna estrarre da una matrice rettangolare  $M$  di dimensione  $4 \times 12$  tre sottomatrici quadrate  $S, T, V$  di dimensione  $4 \times 4$ :

$$M = \begin{array}{|c|c|c|} \hline S & T & V \\ \hline \end{array}$$

e calcolare la matrice  $R = V * S^t + T$ , in cui  $S^t$  è la matrice trasposta di  $S$ .

Si implementino quindi (almeno) i seguenti metodi:

1. *leggi* che effettua la lettura da input degli elementi di una matrice  $m$  ricevuta parametricamente;
2. *scrivi* che scrive su output il contenuto di una matrice  $m$  passata come parametro
3. *moltiplica* che riceve due matrici quadrate  $m1$  ed  $m2$ , supposte della stessa dimensione, e restituisce la matrice prodotto;
4. *addiziona* che riceve due matrici quadrate  $m1$  ed  $m2$ , supposte della stessa dimensione, e restituisce la matrice somma;
5. *trasponi* che riceve una matrice  $m$  e restituisce la sua trasposta;
6. *estrai* che riceve la matrice rettangolare  $m$  ed un intero  $i$  e restituisce una delle matrici quadrate  $S, T, V$  a seconda del valore assunto da  $i$  (per esempio per  $i = 0$  la matrice estratta sarà la  $S$ , per  $i = 1$  la matrice estratta sarà la  $T$  e per  $i = 2$  la matrice estratta sarà la  $V$ ).
7. *main* che, utilizzando i metodi precedenti, provvede a effettuare l'estrazione di  $S, T$  e  $V$ , il calcolo di  $R$  e, infine, la visualizzazione del suo contenuto.

### Esercizio 3

Scrivere un metodo *calcolasommasucc* che riceve in input due array  $V1$  e  $V2$  di  $n$  e  $k$  interi positivi rispettivamente e ritorna un vettore  $V3$  di dimensione  $n$  i cui elementi sono calcolati nel seguente modo: ogni elemento  $V3[i]$  di  $V3$ , per  $i=0, n-1$ , è uguale alla somma degli elementi di  $V2$  dalla posizione  $V1[i]$  alla posizione  $k-1$ . Ad esempio, se  $V1 = [5,3,2,6]$  e  $V2 = [4,5,2,6,7,1,10]$  allora  $V3[0] = V2[5] + V2[6]$  e così via, quindi  $V3 = [11,24,26,10]$ .

## Soluzione Esercizio 1:

All'invocazione del metodo `public static int esercizio (int [] v):`

- Viene effettuata una prima scansione del vettore (dal secondo all'ultimo elemento) alla ricerca di eventuali valori negativi. Il primo elemento negativo incontrato causa la terminazione del metodo con valore di uscita pari a -1.
- Qualora la ricerca precedente non abbia successo, viene eseguita una seconda scansione del vettore (sempre dal secondo all'ultimo elemento); durante questa scansione, per ogni  $i$  compreso tra 1 e  $v.length-1$ , viene calcolato il prodotto di tutti gli elementi compresi tra la posizione 0 e la posizione  $i-1$ . Se trova un elemento il cui valore è uguale al prodotto degli elementi alla sua sinistra, restituisce 1.
- Se neppure questa circostanza è verificata, il metodo restituisce 0.

Nel caso in esempio, il metodo restituisce 1, in quanto  $v[4] = 14$  è pari a  $v[0]*v[1]*v[2]*v[3]$ .

## Soluzione Esercizio 2:

```
import corejava.*;

public class Esercizio2
{
    public static void main (String args[])
    {
        int M[][] = new int[4][12];
        leggi (M);
        int S[][] = estrai (M,0);
        int T[][] = estrai (M,1);
        int V[][] = estrai (M,2);
        int St[][] = trasponi (S);
        int P[][] = moltiplica (V,St);
        int R[][] = addiziona (P,T);
        scrivi (R);
    }

    public static void leggi (int m[][])
    {
        for (int i = 0; i < m.length; i++)
            for (int j = 0; j < m[0].length; j++)
                m[i][j] = Console.readInt ("m["+i+"]["+j+"]=");
    }

    public static void scrivi (int m[][])
    {
        for (int i = 0; i < m.length; i++)
        {
            for (int j = 0; j < m[0].length; j++)
                System.out.print (m[i][j]+" ");
            System.out.println ();
        }
    }

    public static int[][] moltiplica (int m1[][], int m2[][])
    {
```

```

    int dim = m1.length;

    int p[][] = new int[dim][dim];

    for (int i = 0; i < dim; i++)
        for (int j = 0; j < dim; j++)
            {
                int somma = 0;
                for (int k = 0; k < dim; k++)
                    somma += m1[i][k] * m2[k][j];
                p[i][j] = somma;
            }

    return p;
}

public static int[][] addiziona (int m1[][], int m2[][])
{
    int dim = m1.length;

    int s[][] = new int[dim][dim];

    for (int i = 0; i < dim; i++)
        for (int j = 0; j < dim; j++)
            s[i][j] = m1[i][j] + m2[i][j];

    return s;
}

public static int[][] trasponi(int m[][])
{
    int t[][] = new int[m[0].length][m.length];

    for (int i = 0; i < t.length; i++)
        for (int j = 0; j < t[0].length; j++)
            t[i][j] = m[j][i];

    return t;
}

public static int[][] estrai(int m[][], int i)
{
    int nRighe = m.length;
    int nColonne = m[0].length/3;

    int e[][] = new int[nRighe][nColonne];

    for (int k = 0; k < nRighe; k++)
        for (int h = 0; h < nColonne; h++)
            e[k][h] = m[k][h+nColonne*i];
    return e;
}
}

```

### **Soluzione Esercizio 3:**

```

public static int[] calcolasommasucc (int[] v1, int[] v2)
{
    int n=v1.length, k=v2.length;
    int[] v3=new int[n];
    for (int i=0;i<n;i++){
        v3[i]=0;
    }
}

```

```

    for (int j=v1[i]; j<k; j++)
        v3[i]+=v2[j];
    }
    return v3;
}

```

## Altri esercizi d'esame :

### Esercizio 1

Si progetti una classe Matrice che implementi i seguenti metodi:

8. un metodo che effettui la lettura da input degli elementi della matrice;
9. un metodo che verifichi se la matrice **A** è a **strisce orizzontali** considerando la definizione sotto riportata;
10. il metodo **main** che utilizzi i metodi precedenti.

Definizione: Sia **A** una matrice **quadrata** di interi di dimensione **d**; **A** è detta **a strisce orizzontali** se le righe di indice **dispari** presentano valori **tutti nulli**, mentre le **altre** righe presentano valori **tutti non nulli** (per maggiore chiarezza si veda l'esempio seguente).

Esempio: la seguente matrice (**d = 5**) è a strisce orizzontali

1	2	3	4	5
0	0	0	0	0
1	2	3	4	5
0	0	0	0	0
1	2	3	4	5

### Esercizio 2

Dato un vettore **V** di **interi** di lunghezza **n**, scrivere una funzione che restituisca **true** se **ogni** elemento di **V** è **maggiore** del precedente (cioè **V[i] > V[i-1]** con **i > 0**), **false** altrimenti.

Ad esempio, se **V=[5, 6, 8, 11, 15, 20, 26]**, la funzione restituirà true.

### Soluzione Esercizio 1:

```

import corejava.*;

class Esercizio1
{
    private static void leggi (int mat[][])
    {
        for (int i = 0; i < mat.length; i++)
            for (int j = 0; j < mat[0].length; j++)
                mat[i][j] = Console.readInt ("M["+i+"]["+j+"]=");
    }

    private static boolean aStrisce (int mat[][])
    {
        boolean verificato = true;

        // righe di indice pari
        for (int i = 0; i < mat.length && verificato; i+=2)

```

```

        for (int j = 0; j < mat[0].length && verificato; j++)
            if (mat[i][j] == 0)
                verificato = false;

// righe di indice dispari
for (int i = 1; i < mat.length && verificato; i+=2)
    for (int j = 0; j < mat[0].length && verificato; j++)
        if (mat[i][j] != 0)
            verificato = false;

return verificato;
}

public static void main (String args[])
{
    int numeroRighe = 3;
    int numeroColonne = 3;
    int m[][] = new int [numeroRighe][numeroColonne];
    Matrice.leggi (m);
    if (Matrice.aStrisce(m))
        System.out.println ("La matrice e' a strisce orizzontali");
    else
        System.out.println ("La matrice non e' a strisce orizzontali");
}
}

```

## Soluzione Esercizio 2:

```

import corejava.*;

public class Esercizio2
{
    public static void main (String args[])
    {
        int V[] = {5, 6, 8, 11, 15, 20, 26};

        if (ordinato(V))
            System.out.print ("ordinato");
        else
            System.out.print ("non ordinato");
    }

    static boolean ordinato (int V[])
    {
        boolean ordinato = true;
        for (int i = 1; i < V.length && ordinato; i++)
            if (V[i] <= V[i-1])
                ordinato = false;
        return ordinato;
    }
}

```