



# Array Bidimensionali in Java

## Esempi di Elaborazioni su Matrici



## Oggetti e array bidimensionali

- Sviluppiamo un programma Java che effettua elaborazioni su matrici.
- Il programma è composto da due classi:
  - **Matrice**
  - **UsaMatrice**.
- La classe **Matrice** definisce oggetti che sono matrici bidimensionali di valori interi con i metodi **leggi**, **stampa**, **copia**, **estrai**, **sommariga** e **sommacolonna**.
- La classe definisce un costruttore **Matrice** per creare matrici con dimensioni di volta in volta diverse.

- La classe **UsaMatrice** definisce il **main** che usa la classe **Matrice**, creando oggetti **Matrice** e invocando i metodi pubblici della classe.
- Dopo aver creato un oggetto **Matrice**, il **main**:
  - Assegna i valori ai vari elementi leggendoli dalla tastiera,
  - Stampa la matrice,
  - Stampa il valore minimo e massimo delle somme delle righe e delle colonne della matrice stessa.
  - Quindi fa una **copia** della matrice ed estrae da essa una sottomatrice arbitraria tramite il metodo **estrai**.

```
/* Classe Matrice che definisce oggetti matrici
bidimensionali di valori interi con i metodi costruttore,
leggi, stampa, copia, estrai, sommariga e sommacolonna.*/

class Matrice
{
    private int[][] mat;    /* la variabile di istanza è una
                             matrice a due dimensioni */

    Matrice(int nrig, int ncol)    // il metodo costruttore
    {
        mat = new int[nrig][ncol];
    }

    public void leggi(int nr, int nc)    // metodo di lettura
    {
        int i, j;

        for(i=0; i<nr; i++)
            for (j=0; j<nc; j++)
                mat[i][j]= Console.readInt("Inserire elemento " + i
                                             + "," + j + " : ");
    }
}
```

//continua

```
/* metodo per visualizzare la matrice */
public void stampa(int nr, int nc)
{
    int i, j;
    for(i=0; i<nr; i++)
    {
        for (j=0; j<nc; j++)
            System.out.print(mat[i][j] + " ");
        System.out.println();
    }
}

/* metodo per copiare la matrice in un'altra con uguale
numero di righe e colonne */
public void copia(Matrice m)
{
    for(int i=0; i<mat.length; i++)
        for (int j=0; j<mat[0].length; j++)
            mat[i][j] = m.mat[i][j];
} //continua
```

```
/* metodo per estrarre una matrice da un'altra */
public void estrai(Matrice m, int r1, int r2, int c1, int c2)
{
    for(int i=r1; i<=r2; i++)
        for (int j=c1; j<=c2; j++)
            mat[i-r1][j-c1] = m.mat[i][j];
}
/* metodo per sommare gli elementi di una riga */
public int sommariga(int ir)
{
    int j, sommar=0;
    for (j=0; j < mat[ir].length; j++)
        sommar = sommar + mat[ir][j];
    return sommar;
}
/* metodo per sommare gli elementi di una colonna */
public int sommacolonna(int ic)
{
    int i, sommac=0;
    for (i=0; i < mat.length; i++)
        sommac = sommac + mat[i][ic];
    return sommac;
} } // fine classe Matrice
```

```

/* La classe seguente definisce il main che usa la classe
   Matrice, creando oggetti Matrice e invocando i metodi
   pubblici della classe.
class UsaMatrice
{
    public static void main (String args[])
    {
        int numr;
        int numc = 5;
        Matrice mat1, mat2, mat3;
        int sc, sr;
        int minsommarig, minsommacol;
        int maxsommarig, maxsommacol;
        int ri, rf, ci, cf;
        int rig, col;

        numr = Console.readInt("Inserire numero righe: ");
        numc = Console.readInt("Inserire numero colonne: ");
        mat1 = new Matrice(numr,numc);

        mat1.leggi(numr,numc);
        mat1.stampa(numr,numc);
        . . . . .
    }
}
//continua

```

```

        minsommarig=mat1.sommariga(0);
        for (int i=1; i<numr ; i++)
        {
            sr = mat1.sommariga(i);
            if(minsommarig >= sr)
                minsommarig = sr;
        }

        System.out.println("minimo tra le somme delle righe= " +
            minsommarig);

        minsommacol=mat1.sommacolonna(0);
        for (int j=1; j<numc ; j++)
        {
            sc = mat1.sommacolonna(j);
            if(minsommacol >= sc)
                minsommacol = sc;
        }

        System.out.println("minimo tra le somme delle colonne= " +
            minsommacol);
        . . . . .
}
//continua

```

```

maxsommari = mat1.sommariga(0);
for (int i=1; i<numr ; i++)
{
    sr = mat1.sommariga(i);
    if(maxsommari <= sr)
        maxsommari = sr;
}
System.out.println("massimo tra le somme delle righe
                    = " + maxsommari);

maxsommacol = mat1.sommacolonna(0);
for (int j=1; j<numc ; j++)
{
    sc = mat1.sommacolonna(j);
    if(maxsommacol <= sc)
        maxsommacol = sc;
}

System.out.println("massimo tra le somme delle colonne
                    = " + maxsommacol);

. . . . . //continua

```

```

mat2 = new Matrice(numr, numc);

mat2.copia(mat1);
mat2.stampa(numr,numc);

ri = Console.readInt("Inserire riga iniziale: ");
rf = Console.readInt("Inserire riga finale: ");
ci = Console.readInt("Inserire colonna iniziale: ");
cf = Console.readInt("Inserire colonna finale: ");

rig = (rf-ri)+1;
col = (cf-ci)+1;

mat3 = new Matrice(rig, col);

mat3.estrai(mat2, ri, rf, ci, cf);

mat3.stampa(rig,col);
}

} //fine classe UsaMatrice

```



## Array bidimensionali

- Lo stesso programma Java che effettua elaborazioni su matrici senza definire oggetti si può sviluppare come composto da una sola classe **Matrice**.
- La classe **Matrice** definisce array bidimensionali di valori interi che **non sono oggetti** ed i metodi (**statici**) **leggi**, **stampa**, **copia**, **estrai**, **sommariga** e **sommacolonna**.
- La classe definisce al suo interno anche il **main**.
- Il programma è simile al precedente ma non usa oggetti e metodi di istanza, ma variabili e metodi di classe (statici).



## Oggetti e array bidimensionali

- Il metodo **main** crea array bidimensionali e invoca i metodi della classe per effettuare elaborazioni sulle matrici.
- Dopo aver creato un array **mat1**,
  - legge i valori dei vari elementi,
  - stampa la matrice,
  - stampa il valore minimo e massimo delle somme delle righe e delle colonne della matrice,
  - quindi fa una **copia** della matrice **mat1** in **mat2** ed estrae da essa una sottomatrice arbitraria **mat3** tramite il metodo statico **estrai**.

```
public class Matrice
{
    public static void leggi(int[][] mat, int nr, int nc)
    {
        int i, j;

        for(i=0; i<nr; i++)
            for (j=0; j<nc; j++)
                mat[i][j]= Console.readInt("Inserire elemento " + i
                    + "," + j + " : ");
    }

    public static void stampa(int[][] mat, int nr, int nc)
    {
        int i, j;
        for(i=0; i<nr; i++)
        {
            for (j=0; j<nc; j++)
                System.out.print(mat[i][j] + " ");
            System.out.println();
        }
    }
} //continua
```

```
public static void copia(int[][] mat1, int[][] mat2)
{
    for(int i=0; i<mat1.length; i++)
        for (int j=0; j<mat1[0].length; j++)
            mat2[i][j] = mat1[i][j];
}

public static int[][] estrai(int[][]mat, int r1, int r2,
                             int c1, int c2)
{int sm[][] = new int[(r2-r1)+1][(c2-c1)+1];

    for(int i=r1; i<=r2; i++)
        for (int j=c1; j<=c2; j++)
            sm[i-r1][j-c1] = mat[i][j];
    return sm;
}

public static int sommariga(int[][] mat, int ir)
{
    int j, sommar=0;
    for (j=0; j < mat[ir].length; j++)
        sommar = sommar + mat[ir][j];
    return sommar;
} //continua
```

```

public static int  sommacolonna(int[][] mat, int ic)
{
    int i, sommac=0;

    for (i=0; i < mat.length; i++)
        sommac = sommac + mat[i][ic];
    return sommac;
}

/*  Inizio del metodo main  */

public static void main (String args[])
{
    int numr;
    int numc = 5;
    int [][] mat1, mat2, mat3; /*  matrici a due dimensioni */
    int sc, sr;
    int minsommarig, minsommacol;
    int maxsommarig, maxsommacol;
    int ri, rf, ci, cf;
    int rig, col;

                                                                    //continua

```

```

numr = Console.readInt("Inserire numero righe: ");
numc = Console.readInt("Inserire numero colonne: ");

mat1 = new int[numr][numc];

leggi(mat1, numr,numc);

stampa(mat1, numr,numc);

minsommarig=sommariga(mat1, 0);
for (int i=1; i<numr ; i++)
{
    sr = sommariga(mat1, i);
    if(minsommarig >= sr)
        minsommarig = sr;
}
System.out.println("minimo tra le somme delle righe      = "
                    + minsommarig);

                                                                    //continua

```



```

minsommacol=sommacolonna(mat1, 0);
for (int j=1; j<numc ; j++)
{
    sc = sommacolonna(mat1, j);
    if(minsommacol >= sc)
        minsommacol = sc;
}

System.out.println("minimo tra le somme delle colonne = "
    + minsommacol);

maxsommari =sommari (mat1, 0);
for (int i=1; i<numr ; i++)
{
    sr = sommariga (mat1, i);
    if(maxsommari <= sr)
        maxsommari = sr;
}
System.out.println("massimo tra le somme delle righe = " +
    maxsommari);

maxsommacol=sommacolonna(mat1, 0);
for (int j=1; j<numc ; j++)
{
    sc = sommacolonna(mat1, j);
    if(maxsommacol <= sc)
        maxsommacol = sc;
}
System.out.println("massimo tra le somme delle colonne = " +
    maxsommacol); //continua

```

```

mat2 = new int[numr][numc];

copia(mat1, mat2);
stampa(mat2, numr,numc);

ri = Console.readInt("Inserire riga iniziale: ");
rf = Console.readInt("Inserire riga finale: ");
ci = Console.readInt("Inserire colonna iniziale: ");
cf = Console.readInt("Inserire colonna finale: ");

rig = (rf-ri)+1;
col = (cf-ci)+1;

mat3 = new int[rig][col];

mat3 = estrai(mat2, ri, rf, ci, cf);

stampa(mat3, rig,col);

} // fine del main
} // fine della classe

```



## Matrici in Java

- Per estendere i metodi della classe `Matrice` bisognerebbe:
  - Inserire un metodo per il calcolo dei valori medi di ogni riga e ogni colonna della matrice.
  - Inserire un metodo per estrarre una matrice quadrata.
  - Inserire un metodo per il calcolo della diagonale.
  - Modificare il metodo `estrai` per controllare gli indici forniti.
  - Definire i metodi di `somma` e `prodotto` tra matrici
  - .....
  
- Per estendere il programma `main` si potrebbe
  - Usare i metodi per effettuare operazioni più complesse sulle matrici.
  - .....