

# Soluzioni degli esercizi di riepilogo (Fondamenti di Informatica 1 – Walter Didimo)

## Soluzioni delle prove al calcolatore

### Esercizio 1 (esame del 13/01/2006)

```
class Matrice{
    private String[][] mat;    // variabile che memorizza la matrice rappresentata

    /* costruttore che permette di passare la matrice rappresentata in fase di
       istanziamento di un oggetto Matrice */
    public Matrice (String[][] mat){
        int m = mat.length;
        this.mat = new String[m][m];
        for (int i=0; i<m; i++)
            for (int j=0; j<m; j++)
                this.mat[i][j]=mat[i][j];
    }

    /* metodo che restituisce un array contenente tutte le stringhe della
       diagonale principale che hanno lunghezza minore di k */
    public String[] diagonale (int k) {
        /* conta il numero di stringhe della diagonale di lunghezza minore di k */
        int count = 0;
        for (int i=0; i<this.mat.length; i++)
            if (this.mat[i][i].length() < k)
                count ++;

        /* crea e riempie l'array da restituire */
        String[] risultato = new String[count];
        int j=0;
        for (int i=0; i<this.mat.length; i++)
            if (this.mat[i][i].length() < k){
                risultato[j]=this.mat[i][i];
                j++;
            }
        return risultato;
    }
}

class ProvaMatrice{

    public static void main (String[] args){
        InputWindow in = new InputWindow ();
        OutputWindow out = new OutputWindow ("Prova Matrica");

        int dim = in.readInt("Dimensione della matrice?");
        String[][] mat = new String[dim][dim];
        for (int i=0; i<dim; i++)
            for (int j=0; j<dim; j++)
                mat[i][j] = in.readString ("Stringa (" + i + ", " + j + ")?");

        int k = in.readInt("Inserire un intero");
        Matrice matrice = new Matrice (mat);
        String[] ris = matrice.diagonale (k);
        out.writeln ("Stringhe della diagonale più corte di " + k);
        for (int i=0; i<ris.length; i++)
            out.writeln (ris[i]);
    }
}
```

---

## Esercizio 2 (esame del 30/06/2006)

```
class Esercizio{

    public static int[] eliminaDoppi (int[] v){
        /* esiste[k] varrà true se il numero k è già stato incontrato
           nota: inizialmente tutti gli elementi di esiste varranno false */
        boolean[] esiste = new boolean[10];

        /* temp memorizza temporaneamente i numeri di v senza ripetizioni
           nota: temp sarà riempito solo in parte */
        int[] temp = new int[v.length];

        /* count memorizza il numero di elementi copiati in temp, cioè
           quanti sono gli elementi di v senza ripetizioni */
        int count = 0;
        for (int i=0; i<v.length; i++){
            if (!esiste[v[i]]){
                temp[count]=v[i];
                count++;
                esiste[v[i]]=true;
            }
        }
        /* risultato è l'array da restituire, cioè quello che memorizza gli
           elementi di v senza ripetizioni. Tale array ha esattamente la
           lunghezza che serve, cioè pari a count */
        int[] risultato = new int[count];
        for (int i=0; i<count; i++)
            risultato[i] = temp[i];
        return risultato;
    }
}

class ProvaEsercizio{
    public static void main (String[] args){
        InputWindow in = new InputWindow ();
        OutputWindow out = new OutputWindow ();

        int dim = in.readInt("Numero di elementi della sequenza?");
        int[] sequenza = new int[dim];
        for (int i=0; i<dim; i++)
            sequenza[i]=in.readInt("Inserire elemento " + i);

        int[] seqSenzaDoppi = Esercizio.eliminaDoppi(sequenza);
        out.writeln ("Sequenza senza doppi");
        for (int i=0; i<seqSenzaDoppi.length; i++)
            out.write (seqSenzaDoppi[i] + " ");
    }
}
```

### Esercizio 3 (esame del 7/04/2006)

```
class Esercizio{

    public static int[] calcolaMassimo (int[] a, int[] b, int[] c){
        int[] max = new int[a.length];
        for (int i=0; i<a.length; i++){
            int maxTraAeB;
            if (a[i] < b[i])
                maxTraAeB = b[i];
            else
                maxTraAeB = a[i];
            if (maxTraAeB < c[i])
                max[i] = c[i];
            else
                max[i] = maxTraAeB;
        }
        return max;
    }
}
```

```
class ProvaEsercizio{
    public static void main (String[] args){
        InputWindow in = new InputWindow ();
        OutputWindow out = new OutputWindow ();

        int dim = in.readInt("Dimensione delle sequenze?");
        int[] a = new int[dim];
        int[] b = new int[dim];
        int[] c = new int[dim];

        for (int i=0; i<dim; i++){
            a[i] = in.readInt("Elemento a[" + i + "]?");
            b[i] = in.readInt("Elemento b[" + i + "]?");
            c[i] = in.readInt("Elemento c[" + i + "]?");
        }

        int[] max = Esercizio.calcolaMassimo (a,b,c);
        out.writeln ("Stampa dei massimi");
        for (int i=0; i<max.length; i++)
            out.writeln (max[i]);
    }
}
```

---

### Esercizio 4 (esame del 30/09/2005)

```
class StatSuMatrice{

    public static void main (String[] args){
        InputWindow in = new InputWindow ();
        OutputWindow out = new OutputWindow ();

        int r = in.readInt("Numero di righe?");
        int c = in.readInt("Numero di colonne?");

        /* inserimento matrice */
        int[][] mat = new int[r][c];
        for (int i=0; i<mat.length; i++)
            for (int j=0; j<mat[i].length; j++)
```

```

        mat[i][j] = in.readInt("Elemento (" + i + "," + j + ")?");

/* calcolo e stampa delle somme per riga */
for (int i=0; i<mat.length; i++){
    int somma = 0;
    for (int j=0; j<mat[i].length; j++)
        somma += mat[i][j];
    out.writeln ("Somma elementi su riga " + i + "=" + somma);
}

/* calcolo e stampa del massimo e del minimo */
int max, rMax, cMax;
int min, rMin, cMin;
min = max = mat[0][0];
rMax = rMin = cMax = cMin = 0;
for (int i=0; i<mat.length; i++)
    for (int j=0; j<mat[i].length; j++){
        if (max < mat[i][j]){
            max = mat[i][j];
            rMax = i;
            cMax = j;
        }
        if (min > mat[i][j]){
            min = mat[i][j];
            rMin = i;
            cMin = j;
        }
    }
    out.writeln ("massimo = " + max);
    out.writeln ("posizione = (" + rMax + "," + cMax + ")");
    out.writeln ("minimo = " + min);
    out.writeln ("posizione = (" + rMin + "," + cMin + ")");
}
}

```

---

### Esercizio 5 (esame del 6/04/2005)

```

class Sequenza{
    private int[] seq;
    private int count;        // numero di elementi aggiunti

    /* costruttore: crea una sequenza vuota di dimensione pari a dim */
    public Sequenza(int dim){
        this.seq = new int[dim];
        this.count = 0;
    }

    /* aggiunge il numero k alla sequenza
       PRE: la sequenza non deve essere piena */
    public void aggiungi (int k){
        if (count < this.seq.length){
            this.seq[this.count]=k;
            this.count++;
        }
    }

    /* visualizza tutti i numeri della sequenza */
    public void visualizza (){
        for (int i=0; i<this.count; i++)
            System.out.print (this.seq[i] + " ");
        System.out.println ();
    }
}

```

```

    }

    /* ritorna il numero minimo nella sequenza
       PRE: la sequenza non deve essere vuota */
    public int minimo (){
        int min=this.seq[0];
        for (int i=0; i<this.count; i++)
            if (min > this.seq[i])
                min = this.seq[i];
        return min;
    }
}

class ProvaSequenza{

    public static void main (String[] args){
        InputWindow in = new InputWindow ();

        /* crea sequenza */
        int dim = in.readInt("Dimensione della sequenza?");
        Sequenza s = new Sequenza(dim);

        /* inserisce numeri nella sequenza */
        for (int i=0; i<dim; i++){
            int k = in.readInt ("Elemento " + i + "?");
            s.aggiungi(k);
        }

        /* visualizza la sequenza */
        System.out.println ("La sequenza inserita e'");
        s.visualizza();

        /* visualizza il numero minimo */
        System.out.println ("Il numero minimo e' " + s.minimo());
    }
}

```

---

### **Esercizio 6** (*esame del 9/12/2005*)

```

class Quadrato{

    public static int sommaIntorno (int[][] m, int i, int j){
        int dim = m.length;
        int somma = 0;
        for (int k=i-1; k<=i+1; k++)
            for (int h=j-1; h<=j+1; h++)
                if ( !(k==i && h==j) &&
                    (k>=0 && k<dim) &&
                    (h>=0 && h<dim)
                )
                    somma += m[k][h];
        return somma;
    }
}

class ProvaQuadrato{
    public static void main (String[] args){
        InputWindow in = new InputWindow ();

        int dim = in.readInt ("Dimensione matrice?");
    }
}

```

```

int[][] mat = new int[dim][dim];
for (int i=0; i<dim; i++)
    for (int j=0; j<dim; j++)
        mat[i][j]=in.readInt ("Elemento (" + i + ", " + j + ")?");

for (int i=0; i<dim; i++)
    for (int j=0; j<dim; j++){
        int somma = Quadrato.sommaIntorno (mat,i,j);
        System.out.println ("Somma dell'intorno di (" + i + ", " + j +
            ") = " + somma);
    }
}
}

```

## Soluzioni delle prove scritte

*Prova del 9/12/2005*

### Esercizio 1.

```

class Esame{
    private String nomeEsame;
    private int numStudenti;
    private double votoMinimo;
    .....
}

```

### Esercizio 2.

Errore 1: z non è inizializzata quando si valuta  $z = z + y$  (errore di compilazione)  
 Errore 2: il metodo deve restituire uno short, e non un int (errore di compilazione)  
 Errore 3: nel for, la condizione  $i \leq x.length$  causerebbe un `ArrayIndexOutOfBoundsException` (errore di esecuzione)

---

*Prova del 6 aprile 2005*

### Esercizio 1.

Un array in Java è un oggetto che rappresenta una sequenza finita di variabili tutte dello stesso tipo.

Esempio di istanziazione di un array di array di oggetti String.

```
new String[10][10];
```

### Esercizio 2.

```
for (int i=6; i<100; i+=6)
    System.out.println (i);
```

*Oppure (ma è una soluzione meno elegante)*

```
for (int i=1; i<100; i++)
    if (i%6==0)
        System.out.println (i);
```

---

*Prova del 24 giugno 2005*

### **Esercizio 1**

```
public static void ipotenusa (double cateto1, double cateto2){
    double ip = Math.sqrt (cateto1*cateto1 + cateto2*cateto2);
    System.out.println (ip);
}
```

### **Esercizio 2**

```
3b
b12
b3
b1
```

---

*Prova del 13 gennaio 2006*

### **Esercizio 1.**

```
public static int[] arrayDiNumeri(){
    int[] numero = new int[11];
    for (int i=0; i<numero.length; i++)
        numero[i]=i;
    return numero;
}
```

### **Esercizio 2.**

```
Variabili di classe:    telefonoCentralino, numeroUtenzeTotali
Variabili di istanza:  nome, cognome, telefono
Variabili locali:     n, c, t
```

---

*Prova del 30 giugno 2006*

**Esercizio 1** Scrivi un metodo di classe (completo di prototipo) che riceve in input un array di interi e stampa a video tutti i numeri dell'array che hanno un valore compreso nell'intervallo [0,10]

```
public static void stampaArray (int[] a){
    for (int i=0; i<a.length; i++)
        if (a[i]<=10 && a[i]>=0)
            System.out.println (a[i]);
}
```

### **Esercizio 2**

|         |        |
|---------|--------|
| byte    | 8 bit  |
| short   | 16 bit |
| int     | 32 bit |
| long    | 64 bit |
| float   | 32 bit |
| double  | 64 bit |
| char    | 16 bit |
| boolean | 1 bit  |