

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

## Soluzioni delle prove al calcolatore

### Esercizio 7 (esame del 08/01/2009)

```
class TabellaDiNumeri{

    private double[][] tabella;

    /* Costruttore: crea un oggetto TabellaDiNumeri che rappresenta la tabella
       tab passata come parametro */
    public TabellaDiNumeri (double[][] tab){
        int righe = tab.length;
        int colonne = tab[0].length;
        this.tabella = new double[righe][colonne];
        for (int i=0; i<righe; i++)
            for (int j=0; j<colonne; j++)
                this.tabella[i][j] = tab[i][j];
    }

    /* Restituisce il numero della tabella il cui valore
       è più vicino a zero */
    public double vicinoAZero (){
        double numPiccolo = this.tabella[0][0];
        for (int i=0; i<this.tabella.length; i++){
            for (int j=0; j<this.tabella[i].length; j++){
                if (Math.abs(this.tabella[i][j]) < Math.abs(numPiccolo))
                    numPiccolo = this.tabella[i][j];
            }
        }
        return numPiccolo;
    }

    /* Restituisce una descrizione completa della tabella sotto forma di
       oggetto String */
    public String toString (){
        String s="";
        for (int i=0; i<this.tabella.length; i++){
            for (int j=0; j<this.tabella[i].length; j++){
                s += this.tabella[i][j];
                s += "\t";
            }
            s += "\n";
        }
        return s;
    }
}

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

        /* fa inserire le dimensioni della tabella */
        int righe = in.readInt ("Numero di righe?");
        int colonne = in.readInt ("Numero di colonne?");
    }
}
```

```

    /* crea la tabella sotto forma di array di array
    e la fa riempire all'utente */
    double[][] tab = new double[righe][colonne];
    for (int i=0; i<tab.length; i++)
        for (int j=0; j<tab[0].length; j++)
            tab[i][j] = in.readDouble ("Elemento (" + i + ", " + j + ")?");

    /* crea un oggetto TabelleDiNumeri che rappresenti
    la tabella appena inserita*/
    TabellaDiNumeri t = new TabellaDiNumeri (tab);

    /* visualizza una descrizione completa della tabella */
    out.writeln ("Descrizione della tabella inserita");
    out.writeln (t.toString());

    /* visualizza il numero più vicino allo zero nella tabella */
    out.write ("Numero piu' vicino a zero: ");
    out.writeln (t.vicinoAZero());
}
}

```

---

### Esercizio 8 (esame del 24/07/2009)

```

class Famiglia{
    String cognome;
    String[] nomeComponente;
    int[] età;

    public Famiglia (String c, String[] nome, int[] e){
        this.cognome = c;
        this.nomeComponente = new String[nome.length];
        this.età = new int[e.length];
        for (int i=0; i<nome.length; i++){
            this.nomeComponente[i] = nome[i];
            this.età[i] = e[i];
        }
    }

    public String piùGrande (){
        String s = this.cognome;
        String nMax = this.nomeComponente[0];
        int indMax = 0;
        for (int i=1; i<this.nomeComponente.length; i++)
            if (this.età[i] > this.età[indMax]){
                indMax = i;
                nMax = this.nomeComponente[indMax];
            }
        s = s + " " + nMax;
        return s;
    }

    public void minori(){
        for (int i=0; i<this.nomeComponente.length; i++)
            if (this.età[i] < 18)
                System.out.println (this.nomeComponente[i]);
    }
}

```



```

        s += "\nNumero di numeri pari = " + pari;
        s += "\nNumero di numeri dispari = " + dispari;
        s += "\nNumero di divisibili per tre = " + divPerTre;
        s += "\n";

        return s;
    }
}

class ProvaMatriceDiInteri{

    public static void main (String[] args){
        /* Fa inserire la matrice */
        InputWindow in = new InputWindow();
        int m = in.readInt ("Inserire numero di righe");
        int n = in.readInt ("Inserire numero di colonne");
        int[][] matrice = new int[m][n];
        for (int i=0; i<matrice.length; i++)
            for (int j=0; j<matrice[i].length; j++)
                matrice[i][j] = in.readInt("Elemento " + i + "," + j);

        /* Crea oggetto MatriceDiInteri */
        MatriceDiInteri mat = new MatriceDiInteri (matrice);

        /* Visualizza le statistiche */
        System.out.println (mat.analizza());
    }
}

```

---

### **Esercizio 10** (esame del 29/06/2007)

```

class Frase{
    private String[] frase; //memorizza le parole che formano la frase

    /* Costruttore: crea un oggetto Frase composto dalla sequenza di parole
    specificate nell'array f */
    public Frase (String[] f){
        this.frase = new String[f.length];
        for (int i=0; i<f.length; i++)
            this.frase[i]=f[i];
    }

    /* Restituisce un array contenente le sole parole della frase che NON
    includono la sottostringa ss */
    public String[] eliminaContententi (String ss){
        String[] risultato; // l'array da restituire

        /* conta il numero di parole che non contengono ss */
        int count=0;
        for (int i=0; i<this.frase.length; i++)
            if (this.frase[i].indexOf(ss) < 0)
                count ++;

        /* crea l'array delle parole che non contengono ss e lo riempie */
        risultato = new String[count];
        int j = 0; // indice per scandire l'array risultato
        for (int i=0; i<this.frase.length; i++)
            if (this.frase[i].indexOf(ss) < 0){
                risultato[j] = this.frase[i];
                j++;
            }
    }
}

```

```

        }
        return risultato;
    }
}

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

        /* Fa inserire una frase composta da parole senza spazio */
        int numParole = in.readInt ("Numero parole nella frase?");
        String[] frase = new String[numParole];
        for (int i=0; i<frase.length; i++)
            frase[i] = in.readString ("Parola numero " + i);

        /* Crea un oggetto Frase */
        Frase f = new Frase (frase);

        /* Fa inserire una ulteriore parola */
        String ss = in.readString ("Ora inserisci un'altra parola");

        /* Visualizza le stringhe della frase che non contengono
        l'ultima parola inserita */
        String[] senzaSS = f.eliminaContenenti (ss);
        out.writeln ("Parole che non contengono " + ss + ":");
        for (int i=0; i<senzaSS.length; i++)
            out.writeln (senzaSS[i]);

    }
}

```

---

## Soluzioni delle prove scritte

*Prova del 08/01/2008*

### Esercizio 1.

```

public static void stringheContenenti (String[] a, String s){
    for (int i=0; i<a.length; i++)
        if (a[i].indexOf(s) == 0)
            System.out.println (a[i]);
}

```

### Esercizio 2.

- Le parti principali che compongono la struttura di una classe Java sono:

- a) l'intestazione, che comprende il nome della classe;
- b) il corpo della classe, che si struttura nella definizione degli attributi (variabili di istanza e di classe) e nella definizione dei metodi.

- Il cast esplicito si rende necessario quando il dominio di valori della variabile in cui si vuol scrivere è più piccolo del dominio di appartenenza del valore di assegnare. Segue un esempio.

```

int a = 10;
short b = (int) a;

```

---

Prova del 24/07/2009

**Esercizio 1.** Scrivi un metodo di classe che prende in ingresso, come parametro formale, un intero k e che restituisce in uscita una matrice quadrata di dimensione k, avente tutti 1 sulla diagonale principale e tutti 0 nelle altre celle.

```
public static int[][] creaMatrice (int k){
    int[][] mat = new int[k][k];
    for (int i=0; i<mat.length; i++)
        mat[i][i] = 1;
}
```

**Esercizio 2.** Svolgi i seguenti punti.

```
String temp = s.toUpperCase ();
int numVoc = 0;
for (int i=0; i<temp.length(); i++){
    char c = temp.charAt(i);
    if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
        numVoc++;
}
System.out.println (numVoc);
```

La differenza tra i tipi byte, short, int e long sta nell'ampiezza dei del dominio di valori che essi rappresentano. Precisamente, i domini di valori sono i seguenti:

- Dominio byte  $[-2^7, 2^7-1]$
  - Dominio short  $[-2^{15}, 2^{15}-1]$
  - Dominio int  $[-2^{31}, 2^{31}-1]$
  - Dominio long  $[-2^{63}, 2^{63}-1]$
- 

Prova del 11/01/2008

**Esercizio 1.**

```
public static boolean[][] verificaPari (int[][] mat){
    int m = mat.length;
    int n = mat[0].length;
    boolean[][] mat1 = new boolean[m][n];
    for (int i=0; i<m; i++)
        for (int j=0; j<n; j++)
            if (mat[i][j]%2 == 0)
                mat1[i][j] = true;
    return mat1;
}
```

**Esercizio 2.**

1. Un oggetto è l'istanza di una classe. La classe definisce gli attributi e i metodi delle sue istanze.
2. Sì, in Java ogni istruzione iterativa può opportunamente essere rimpiazzata da un'altra. E' sufficiente effettuare in modo adeguato piccole modifiche locali al codice.

Prova del 29/06/2007

---

### Esercizio 1.

```
public static void metodo (String[][] mat){
    for (int i=0; i<mat.length; i++)
        for (int j; j<mat[i].length; j++){
            char inizio = mat[i][j].charAt(0);
            char fine = mat[i][j].charAt(mat[i][j].length()-1);
            if ( (inizio=='a' || inizio=='A') &&
                (fine=='a' || fine=='A') )
                System.out.println (mat[i][j]);
        }
}
```

### Esercizio 2.

- 1) **[F]**. Il riferimento `this` non può essere usato all'interno di un metodo di classe, perché si presuppone che il metodo di classe sia invocato sulla classe, e quindi non ha senso parlare di oggetto ricevente
  - 2) **[F]**. Ogni variabile di classe esiste in unica copia per tutte le istanze della classe.
  - 3) **[V]**. Perché il risultato dell'espressione `i%10` è sempre un numero più piccolo di 10, essendo il resto della divisione intera tra `i` e 10.
-