

10 Dicembre 2001

Alcune Soluzioni

ESERCIZIO 1. Si consideri il seguente frammento di codice in linguaggio Java:

```
int[] [] a = new int[n][n];
for (int i = 0; i < a.length; i++) {
    for (int j = i; j < a.length; j++) {
        System.out.println(i+" "+j);
        a[i][j] = (i+j)%2;
    }
}
```

Determinare:

- 1) cosa esattamente appare sullo schermo nel caso $n=5$;
- 2) i valori di tutti gli elementi dell'array **a** nel caso $n=4$.

Soluzione

- 1) Nel caso $n=5$ sullo schermo appare:

```
0 0
0 1
0 2
0 3
0 4
1 1
1 2
1 3
1 4
2 2
2 3
2 4
3 3
3 4
4 4
```

- 2) Nel caso $n=4$ i valori di tutti gli elementi dell'array **a** sono:

```
0 1 0 1
0 0 1 0
0 0 0 1
0 0 0 0
```

ESERCIZIO 2. Si consideri il seguente frammento di codice in linguaggio Java:

```
int[] b = lista;
boolean condizione = true;
int i = 0;
while (i < b.length-1 && condizione) {
    if (b[i] == b[i+1])
        condizione = false;
    else i++;
}
```

Soluzione

- 1) $i = 4$ e **condizione** = true;
- 2) $i = 2$ e **condizione** = false;
- 3) $i = 0$ e **condizione** = true.

Determinare il valore finale delle variabili **condizione** e **i** nei seguenti casi:

- 1) $\text{lista}=\{4,5,6,4,5\}$;
- 2) $\text{lista}=\{4,3,4,4,4\}$;
- 3) $\text{lista}=\{4\}$.

ESERCIZIO 3. Scrivere un metodo in linguaggio Java che data una stringa *s* e due caratteri *c1* e *c2* determini se il numero di occorrenze di *c1* in *s* sia uguale o meno al numero di occorrenze di *c2*.

Soluzione

```
public static boolean occorrenze (String s, char c1, char c2) {
    int count1 = 0;
    int count2 = 0;
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) == c1)
            count1 += 1;
        if (s.charAt(i) == c2)
            count2 += 1;
    }
    return (count1 == count2);
}
```

ESERCIZIO 4. Si consideri il seguente metodo ricorsivo:

```
public static String metodo(String s) {
    if (s.length() == 0)
        return "";
    else
        return s.charAt(0) + metodo(s.substring(1)) + s.charAt(0);
}
```

Determinare la stringa restituita dalla chiamata `metodo("ciao")`. Scrivere inoltre un metodo non ricorsivo in linguaggio Java che realizzi la stessa funzione.

Soluzione

La stringa restituita dalla chiamata `metodo("ciao")` è "ciaooaic".

Un metodo non ricorsivo che realizza la stessa funzione è il seguente:

```
public static String metodoNonRic(String s) {
    String aux = s;
    for (int i = s.length()-1; i >= 0; i--)
        aux = aux + s.charAt(i);
    return aux;
}
```

ESERCIZIO 5. Scrivere un metodo ricorsivo in linguaggio Java che dato un array di interi ne restituisca il valore minimo.

Soluzione

```
public static int auxmin(int[] a, int i) {
    if (i == a.length-1)
        return a[i];
    return Math.min(a[i], auxmin(a, i+1));
}

public static int minimo(int[] a) {
    return auxmin(a, 0);
}
```

ESERCIZIO 6. Si consideri la seguente classe Java `Studiante`. Le variabili istanza sono il nome e i voti riportati in cinque materie. I metodi previsti sono, oltre a quello che restituisce il nome dello studente (`get_nome`), quelli per inserire (`set_voto`) e ottenere (`get_voto`) i voti nelle diverse materie. E' previsto infine un metodo (`get_media`) che restituisce la media dei voti:

```
public class Studiante {
    public Studiante (String nome) {
        this.nome = nome;
    }

    public String get_nome() {
        return this.nome;
    }

    public void set_voto(int materia, int votazione) {
        this.voto[materia] = votazione;
    }

    public int get_voto(int materia) {
        return this.voto[materia];
    }

    public double get_media() {
        int somma = 0;
        for (int i = 0; i < 5; i++)
            somma += this.voto[i];
        return somma/5.0;
    }

    private String nome;
    private int[] voto = new int[5];
}
```

Scrivere una nuova classe `Corso`, dove ciascun corso è caratterizzato da un nome e da un elenco di studenti (il cui numero massimo deve essere impostato dal costruttore della classe), che abbia i seguenti metodi:

1. un metodo che restituisce il nome del corso;
2. un metodo che aggiunge uno studente al corso;
3. un metodo che restituisce l'elenco dei nomi di tutti gli studenti del corso;
4. un metodo che restituisce l'elenco dei nomi degli studenti che abbiano una media superiore ad un dato valore x .

Scrivere inoltre un breve programma di prova che illustri l'uso della classe `Corso` e dei suoi metodi.

Soluzione

```
public class Corso {

    private String nome_corso;
    private Studiante[] studenti;
    private int count = 0;

    public Corso (String nome, int n) {
```

```

    nome_corso = nome;
    studenti = new Studente[n];
}

public String get_nome_corso() {
    return nome_corso;
}

public void add_studente(Studente s) {
    if (count < studenti.length) {
        studenti[count] = s;
        count++;
    }
}

public String[] get_nomi_studenti() {
    String[] s = new String[count];
    for (int i = 0; i < count; i++)
        s[i] = studenti[i].get_nome();
    return s;
}

public String[] get_nomi_media(double x) {
    int n = 0;
    for (int j = 0; j < count; j++)
        if (studenti[j].get_media() > x)
            n++;
    String[] s = new String[n];
    int j = 0;
    for (int i = 0; i < count; i++)
        if (studenti[i].get_media() > x) {
            s[j] = studenti[i].get_nome();
            j++;
        }
    return s;
}
}

```

ESERCIZIO 7 Si consideri la seguente classe Nazione:

```

public class Nazione {
    public Nazione(String nome, String capitale) {
        this.nome = nome;
        this.capitale = capitale;
    }
    public String get_nome_stato() {
        return this.nome;
    }
    public String get_nome_capitale() {
        return this.capitale;
    }
    private String nome, capitale;
}

```

Scrivere due sottoclassi, Monarchia e Repubblica, della classe Nazione con le seguenti caratteristiche:

1. le classi Monarchia e Repubblica hanno un capo di stato, un re nel caso di Monarchia, un presidente nel caso di Repubblica. Hanno inoltre un metodo che restituisce il nome del capo di stato.
2. la stringa restituita dal metodo `get_nome_stato` di queste due classi deve contenere, oltre al nome, la dizione "Regno di" nel caso di una monarchia e "Repubblica di" nel caso di una repubblica. Esempio: per un oggetto della classe Monarchia con `nome = "Spagna"` il metodo `get_nome_stato` deve restituire "Regno di Spagna".

Scrivere inoltre un breve programma di prova che illustri l'uso della classe Nazione e delle sottoclassi Monarchia e Repubblica.

Soluzione

```
public class Monarchia extends Nazione {

    private String nome_re;

    public Monarchia (String nome, String capitale, String re) {
        super(nome, capitale);
        nome_re = re;
    }

    public String get_nome_re() {
        return nome_re;
    }

    public String get_nome_stato() {
        return ("Regno di" + super.get_nome_stato());
    }
}

public class Repubblica extends Nazione {

    private String nome_presidente;

    public Repubblica (String nome, String capitale, String presidente) {
        super(nome, capitale);
        nome_presidente = presidente;
    }

    public String get_nome_presidente() {
        return nome_presidente;
    }

    public String get_nome_stato() {
        return ("Repubblica di" + super.get_nome_stato());
    }
}
```