

## Modulo di Laboratorio di Programmazione I (CdL Informatica)

a.a. 2009-2010

Prova Totale - 1 Marzo 2010

NOME

COGNOME

MATRICOLA

CORSO DI LAUREA

Scrivere in stampatello **Nome**, **Cognome** e **Matricola** su ogni foglio consegnato.

**Nota Bene:** Il testo del compito deve essere riconsegnato insieme al proprio elaborato scritto. Le soluzioni degli esercizi **non** devono contenere alcun metodo `main` né alcuna istruzione di stampa né istruzioni `switch`, `break`, `continue`, etc. La soluzione per l'Esercizio 3 **non** deve contenere alcuna istruzione di ciclo (`for`, `while`, etc.). La non osservanza di tali requisiti comporterà l'annullamento della prova scritta dello studente.

**Esercizio 1.** Si consideri il seguente frammento di codice in Java:

```
String[] [] a = b;
boolean p = true;
int i = 0, j = 0, k = 0;
while (i < a.length && p) {
    for (j=0; j < a[i].length-1; j++) {
        if (a[i][j].length() >= a[i][j+1].length())
            p = false;
    }
    if (!p)
        k = i;
    i++;
}
```

Determinare il valore finale delle variabili `i`, `j`, `k` e `p` per i seguenti valori di `b`:

- 1) `b = {"isa", "lisa", "elisa"}, {"luc", "luke"}, {"ada", "aida", "ida", "iva"};`
- 2) `b = {"lea", "luca", "lucas"}, {"paolo", "paul", "pal", "per"}.`

**Esercizio 2.** Scrivere un metodo statico iterativo che, dati un array monodimensionale `a` di stringhe ed un array monodimensionale `c` di caratteri, restituisce un array monodimensionale `b` di stringhe tale che `b[i]` è la stringa ottenuta concatenando tutti i caratteri di `a[i]` uguali a `c[i]`. Nel caso in cui `c[i]` non compaia in `a[i]`, allora `b[i]` è la stringa vuota.

Esempio: dati `a = {"minnie", "lillo", "pluto", "pippo"}` e `c = {'i', 'l', 'r', 'p'}`, il metodo restituisce l'array `b = {"ii", "lll", "", "ppp"}`.

Si assuma che gli array `a` e `c` abbiano la stessa lunghezza.

**Esercizio 3.** Scrivere una versione ricorsiva del metodo definito per risolvere l'Es.2.

**Esercizio 4.** Un *corso universitario* può essere caratterizzato da nome, nome del docente titolare, numero di crediti, area scientifica e anno di insegnamento. Scrivere una classe `Corso` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza.

Inoltre, definire un metodo per modificare il nome del docente titolare di un corso, un metodo `equals` che restituisce `true` se due corsi sono uguali (altrimenti restituisce `false`) ed un metodo che restituisce una stringa che descrive un corso.

**Esercizio 5.** Un *laboratorio* è un corso universitario caratterizzato anche dal numero di ore di lezione svolte in laboratorio e dalla sua frequenza che può essere obbligatoria o meno. Scrivere una classe `Laboratorio` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo per modificare il numero di ore di un laboratorio ed un metodo che restituisce una stringa che descrive un laboratorio.

**Esercizio 6.** L'*offerta formativa* di un corso di laurea può essere caratterizzata tramite il titolo della laurea, la sede universitaria e l'elenco dei corsi universitari erogati. Scrivere una classe `OffertaFormativa`, il cui costruttore imposta il numero massimo dei corsi erogati. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire i seguenti metodi:

- un metodo che, dato un corso universitario `c`, aggiunge `c` nell'elenco dei corsi dell'offerta formativa;
- un metodo che, data una stringa `s`, restituisce l'elenco dei corsi in una data offerta formativa la cui area scientifica è `s`;
- un metodo che restituisce l'elenco dei laboratori in una data offerta formativa la cui frequenza è obbligatoria.

**Esercizio 7.** Siano date le seguenti classi:

```
class Prima {
    private String s;

    public Prima (String s) {
        this.s = s;
    }
    public String getS() {
        return this.s;
    }
    public String m (Prima a) {
        if (this.s.length()==a.s.length())
            return this.s + a.s;
        else
            return a.s + this.s;
    }
}

class Sec extends Prima {
    private String t;

    public Sec (String s, String t) {
        super(s);
        this.t = t;
    }
    public String getT() {
        return this.t;
    }
    public String m (Prima a) {
        if (this.t.equals(a.getS()))
            return super.m(a) + this.t;
        else
            return this.t + super.m(a);
    }
}
```

Dato il seguente frammento di codice:

```
Prima a = new Prima ("hello");
Sec b = new Sec ("ciao","byebye");
String x = b.m(a);
String y = a.m(b);
a = b;
String z = a.m(b);
```

determinare qual è il valore delle variabili `x`, `y` e `z`.