

Modulo di Laboratorio di Programmazione I (CdL Informatica) &
Corso di Laboratorio di Programmazione (disattivato - CdL Matematica)
a.a. 2011-2012

Prova Totale - 15 Febbraio 2012

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:

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

Determinare il valore finale delle variabili **i**, **j**, **p** ed **s** nei seguenti casi:

- 1) $b = \{\{9,3\}, \{3,1,-1,-2,-3\}, \{3,12,4\}, \{8,-2\}\}$;
- 2) $b = \{\{7\}, \{10,6,0\}, \{14,8,3,2,2,-1\}\}$.

Esercizio 2. Scrivere un metodo *statico* iterativo che, dati un array monodimensionale di stringhe *a* ed un intero $k > 0$, restituisce *true* se *ogni* stringa di *a* ha *almeno* *k* coppie di caratteri adiacenti uguali, altrimenti il metodo restituisce *false*.

Esempio: se $a = \{\text{"accesso"}, \text{"bbtcrryy"}, \text{"ggg"}\}$ e $k = 2$, il metodo restituisce *true*.
Modificare la definizione del metodo in modo da sollevare un'eccezione se $k \leq 0$.

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

Esercizio 4. Un *comune* può essere caratterizzato da nome, nome della provincia in cui si trova, nome del sindaco, numero di abitanti, superficie e altitudine. Scrivere una classe **Comune** con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo che modifica il numero di abitanti, un metodo che modifica il nome del sindaco ed un metodo che, dato un intero $k > 0$, restituisce *true* se un comune ha più di *k* abitanti, *false* altrimenti. Infine, definire un metodo che restituisce una stringa che descrive un comune.

Esercizio 5. Un *comune commissariato* è un comune caratterizzato anche dal nome del commissario, il motivo del commissariamento ed il numero del decreto di scioglimento del consiglio comunale. Scrivere una classe `ComuneCommiss` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Definire inoltre un metodo che restituisce una stringa che descrive un oggetto della classe `ComuneCommiss`.

Esercizio 6. Una *provincia* può essere caratterizzata tramite nome, nome del presidente, numero di abitanti, superficie e l'elenco dei comuni appartenenti alla provincia. Scrivere una classe `Provincia`, il cui costruttore imposta il numero massimo dei comuni che possono appartenere ad una provincia. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire i seguenti metodi:

- un metodo che, dato un comune c , aggiunge c nell'elenco dei comuni, a patto che il nome della provincia coincida con quello che compare in c ;
- un metodo che, dati gli interi $k, m > 0$, restituisce *true* se in una provincia esistono almeno m comuni con più di k abitanti, *false* altrimenti (N.B. usare il metodo definito nell'Es.4);
- un metodo che, data una stringa s , restituisce l'elenco di tutti i comuni di una provincia commissariati con motivo s .

Esercizio 7. Siano date le seguenti classi:

```

class A {
    private String s;

    public A (String t) {
        this.s = t;
    }
    public String getS() {
        return this.s;
    }
    public String m (A a) {
        if (this.s.equals(a.s))
            return this.s + "*2";
        else
            return this.s + a.s;
    }
}

class B extends A {
    private int m;

    public B (String t, int r) {
        super(t);
        this.m = r;
    }
    public int getM() {
        return this.m;
    }
    public String m (A a) {
        if (this.m > a.getS().length())
            return this.m + super.m(a);
        else
            return super.m(a);
    }
}

```

Dato il seguente frammento di codice:

```

A a1 = new A ("neve");
A a2 = new B ("snow",5);
B a3 = new B ("neige",5);
String s = a2.m(a1), t = a1.m(a2);
a1 = a2;
String u = a1.m(a3);

```

determinare il valore delle variabili s , t ed u .