

Modulo di Laboratorio di Programmazione I (CdL Informatica)

a.a. 2009-2010

Prova Scritta - 24 Giugno 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;
int i = 0, j = 0, k = 0;
boolean p = false;
for (i=0; i<a.length; i++) {
    j = 0; p = false;
    while (j<a[i].length() && !p) {
        if (a[i].charAt(j) == c)
            p = true;
        j++;
    }
    if (p)
        k++;
}
```

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

- 1) `b = {"abddd", "fd", "dgh", "gag"}` e `c = 'd'`;
- 2) `b = {"kfg", "m", "bbrrr"}` e `c = 'r'`.

Esercizio 2. Scrivere un metodo statico iterativo che, dato un array monodimensionale `a` di caratteri, restituisce un array bidimensionale `b` di caratteri tale che la prima riga di `b` contiene un'occorrenza del primo elemento di `a`, la seconda riga di `b` contiene due occorrenze del secondo elemento di `a`, ..., l'ultima riga di `b` contiene `a.length` occorrenze dell'ultimo elemento di `a`. Esempio: dato `a = {'g', 'w', 'p', 'z'}`, il metodo restituisce l'array `b = {{'g'}, {'w', 'w'}, {'p', 'p', 'p'}, {'z', 'z', 'z', 'z'}}`.

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

Esercizio 4. Una *via cittadina* può essere caratterizzata tramite il nome, la città in cui si trova, la lunghezza (espressa in km) e l'essere pedonale o meno. Scrivere una classe `ViaCittadina` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo per modificare il nome della via, un metodo per modificare lo status di via pedonale o meno, ed un metodo che restituisce una stringa che descrive una via cittadina.

Esercizio 5. Una *via storica* è una via cittadina caratterizzata anche dall'elenco dei nomi dei monumenti storici che si trovano in tale via. Scrivere una classe `ViaStorica`, il cui costruttore imposta il numero massimo di monumenti storici di una via. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire un metodo che aggiunge il nome di un monumento nell'elenco di una via storica ed un metodo che restituisce una stringa che descrive una via storica.

Esercizio 6. Una *corsa podistica cittadina* può essere caratterizzata tramite il nome della corsa, il nome dello sponsor, la lunghezza (espressa in km), l'essere competitiva o meno, e l'elenco delle vie cittadine che ne costituiscono il percorso. Scrivere una classe `Corsa`, il cui costruttore imposta il numero massimo delle vie cittadine che possono costituire il percorso. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire:

- un metodo che aggiunge una via cittadina nell'elenco delle vie di una corsa;
- un metodo che, data una via cittadina, controlla se essa compare o meno nel percorso di una corsa;
- un metodo che restituisce l'elenco delle vie storiche che compaiono nel percorso di una corsa.

Esercizio 7. Siano date le seguenti classi:

```
class A1 {
    private char car;

    public A1 (char c) {
        this.car = c;
    }
    public char getCar() {
        return this.car;
    }
    public String m (A1 a) {
        if (this.car != a.car)
            return this.car+"!="+a.car;
        else
            return this.car+"=="a.car;
    }
}

class A2 extends A1 {
    private String s;

    public A2 (char c, String t) {
        super(c);
        this.s = t;
    }
    public String getS() {
        return this.s;
    }
    public String m (A1 a) {
        if (this.s.charAt(0)==a.getCar())
            return super.m(a) + this.s;
        else
            return this.s + super.m(a);
    }
}
```

Dato il seguente frammento di codice:

```
A1 obj1 = new A2 ('t', "titti");
A1 obj2 = new A1 ('s');
String s1 = obj2.m(obj1);
A2 obj3 = new A2 ('t', "tatties");
String s2 = obj3.m(obj2);
obj2 = obj1;
String s3 = obj2.m(obj3);
```

determinare il valore delle variabili `s1`, `s2` ed `s3`.