

Scrivere una classe `Farmaco` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo per modificare il nome del produttore, un metodo che, dati due farmaci, controlla se appartengono alla stessa categoria e hanno lo stesso principio attivo, ed un metodo che restituisce una stringa che descrive un farmaco.

Esercizio 5. Un *farmaco da banco* è un farmaco per il quale non è necessaria la ricetta medica ed è caratterizzato anche dal prezzo (interamente a carico del paziente) e dalla percentuale massima di sconto che può essere applicata. Scrivere una classe `FarmacoDaBanco` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo per modificare il prezzo ed un metodo che restituisce una stringa che descrive un farmaco da banco.

Esercizio 6. Una *farmacia* può essere caratterizzata tramite nome, nome del proprietario, indirizzo e l'elenco dei farmaci in vendita. Scrivere una classe `Farmacia`, il cui costruttore imposta il numero massimo di farmaci trattati da una farmacia. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire i seguenti metodi:

- un metodo che, dato un farmaco f , aggiunge f nell'elenco dei farmaci di una farmacia;
- un metodo che, dato un farmaco f , controlla se f è in vendita presso una farmacia;
- un metodo che, dato un intero n , restituisce l'elenco dei farmaci di una farmacia che scadono nell'anno n ;
- un metodo che, data una stringa s , restituisce l'elenco di tutti i farmaci da banco in vendita presso una farmacia che sono prodotti dal produttore s .

Esercizio 7. Siano date le seguenti classi:

```
class A {
    private String lab;

    public A (String s) {
        this.lab = s;
    }
    public String getLab() {
        return this.lab;
    }
    public String m (A a) {
        if (this.lab.charAt(0) ==
            a.lab.charAt(0))
            return this.lab + a.lab;
        else
            return a.lab + this.lab;
    }
}

class B extends A {
    private char c;

    public B (String s, char c) {
        super(s);
        this.c = c;
    }
    public int getC() {
        return this.c;
    }
    public String m (A a) {
        return this.c + super.m(a);
    }
}
```

Dato il seguente frammento di codice:

```
A x = new B ("parziale", '2');
A y = new A ("prova scritta");
B z = new B ("totale", '1');
String s1 = x.m(z), s2 = y.m(x);
y = z;
String s3 = z.m(y);
```

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