



POLITECNICO
MILANO 1863

Ingegneria del Software

Esercitazione 04

Gianpaolo Cugola

Esercizio 1

- Si vuole scrivere un programma per gestire i piani di studio di uno o più studenti
- Ogni studente è caratterizzato da nome, cognome e id persona
- Ogni piano di studio contiene un insieme di “esami”
 - Ognuno caratterizzato da:
 - I dati del corso:
 - Codice corso, nome corso e crediti
 - I dati del docente che eroga quel corso per quel particolare studente
 - Nome, cognome, email e ruolo del docente (ordinario, associato, ricercatore)
 - L’informazione se l’esame sia stato superato o meno e in caso affermativo data e voto di superamento (con indicazione della lode, se necessario)
 - Nel piano di studio non possono esserci esami duplicati (ovvero relativi al medesimo corso)
 - Deve essere possibile scorrere l’insieme dei corsi nel piano di studio
 - Deve essere possibile stampare il piano di studio con elenco degli esami superati o meno, numero di crediti acquisiti e media pesata

Esercizio 2

```
interface Product {
    public double getPrice();
}

class Drink implements Product {
    private double price;
    public Drink(double price) {
        this.price = price;
    }
    public double getPrice() { return price; }
    public boolean moreExpensive(Drink d) {
        return this.getPrice() > d.getPrice();
    }
}

class Coca extends Drink {
    public Coca(double price) { super(price); }
    public int getCalories() { return 200; }
}

class CocaZero extends Coca {
    public CocaZero(double price) { super(price); }
    public int getCalories() { return 0; }
    public boolean moreExpensive(CocaZero c) {
        return false;
    }
}
```

```
public class Polimorfismo {
    public static void main(String[] args) {
        Product p;
        Drink d1, d2;
        Coca c1, c2;

        p = new Coca(2.0);
        d1 = p;
        d2 = new Drink(1.5);
        c1 = new Coca(2.1);
        c2 = new CocaZero(2.5);

        System.out.println(p.getPrice());
        System.out.println(d1.getPrice());
        System.out.println(d2.getPrice());
        System.out.println(c1.getPrice());
        System.out.println(c2.getPrice());
        System.out.println(p.getCalories());
        System.out.println(c1.getCalories());
        System.out.println(c2.getCalories());
        System.out.println(c1.moreExpensive(d2));
        System.out.println(c1.moreExpensive(c2));
        CocaZero cz = new CocaZero(2.0);
        System.out.println(c2.moreExpensive(cz));
    }
}
```