

L7 - Interfacce - Base

Java · interface, implements, metodi da contratto

Materiale didattico di Moreno Sardella · morenosardella.com

Prima di iniziare. Questi esercizi coprono i concetti minimi della lezione: falli tutti, nell'ordine. Scrivi il codice a mano o in un IDE, compila ed esegui. Per ognuno e' indicato un **prototipo suggerito**: usalo come punto di partenza. Le soluzioni complete sono in fondo: guardale solo dopo aver provato.

Esercizi

Esercizio 1 — La prima interfaccia

INTERFACE & MIDDLE DOT; IMPLEMENTS

Crea l'interfaccia **Suonabile** con un metodo **void suona()**. Crea **Chitarra** e **Pianoforte** che la implementano stampando il proprio suono.

```
interface Suonabile { void suona(); }
class Chitarra implements Suonabile { /* ... */ }
```

Esercizio 2 — Interfaccia con piu' metodi

CONTRATTO

Crea **Figura** (interfaccia) con **double area()** e **double perimetro()**. Implementala in **Rettangolo**. Chi implementa l'interfaccia DEVE definire entrambi i metodi.

```
interface Figura {
    double area();
    double perimetro();
}
```

Esercizio 3 — Pagabile

VERSO LA VERIFICA

Crea l'interfaccia **Pagabile** con **double getImporto()**, **String getTipo()** e **String getId()**. Implementala in una classe **Bolletta** (importo fisso).

```
interface Pagabile {
    double getImporto();
    String getTipo();
    String getId();
}
```

Esercizio 4 — Usare il tipo interfaccia

VARIABILE = INTERFACCIA

Dichiara una variabile di tipo **Suonabile** e assegnale prima una Chitarra, poi un Pianoforte, chiamando suona() ogni volta. Mostra che il tipo interfaccia puo' riferire qualsiasi implementazione.

```
Suonabile s = new Chitarra();
s.suona();
s = new Pianoforte();
s.suona();
```

Soluzioni

Confronta solo dopo aver provato da solo. Esistono soluzioni equivalenti: conta la logica, non la copia esatta.

Soluzione 1

```
interface Suonabile { void suona(); }
class Chitarra implements Suonabile {
    public void suona() { System.out.println("Strimpella"); }
}
class Pianoforte implements Suonabile {
    public void suona() { System.out.println("Plin plin"); }
}
```

Soluzione 2

```
interface Figura { double area(); double perimetro(); }
class Rettangolo implements Figura {
    private double b, h;
    Rettangolo(double b, double h){this.b=b;this.h=h;}
    public double area() { return b*h; }
    public double perimetro() { return 2*(b+h); }
}
```

Soluzione 3

```
interface Pagabile { double getImporto(); String getTipo(); String getId(); }
class Bolletta implements Pagabile {
    private String id; private double importo;
    Bolletta(String id, double importo){this.id=id;this.importo=importo;}
    public double getImporto() { return importo; }
    public String getTipo() { return "Bolletta"; }
    public String getId() { return id; }
}
```

Soluzione 4

```
Suonabile s = new Chitarra(); s.suona();
s = new Pianoforte(); s.suona();
// la stessa variabile usa implementazioni diverse
```