

L6 - Ereditarieta' e polimorfismo - Avanzati

Java · classi astratte, polimorfismo, instanceof, casting

Materiale didattico di Moreno Sardella · morenosardella.com

Prima di iniziare. Questi esercizi sono piu' impegnativi: combinano piu' concetti e si avvicinano al livello della verifica. Progetta prima su carta (classi, metodi, dati), poi scrivi il codice. Le soluzioni in fondo mostrano *una* via corretta: la tua puo' differire, conta che la logica funzioni e che il codice compili.

Esercizi

Esercizio 1 — Classe astratta Figura

ABSTRACT

Crea **abstract class** **Figura** con metodo astratto **double area()**. Crea **Cerchio** (raggio) e **Quadrato** (lato) che lo implementano. Non si deve poter fare **new Figura()**.

```
abstract class Figura {  
    public abstract double area();  
}
```

Esercizio 2 — Array polimorfico

POLIMORFISMO

Crea un array **Figura[]** con un **Cerchio** e un **Quadrato**. Scorilo con un for-each e stampa l'area di ciascuno: ogni oggetto usa il **suo** metodo **area()** (polimorfismo).

```
Figura[] figure = { new Cerchio(2), new Quadrato(3) };  
for (Figura f : figure) System.out.println(f.area());
```

Esercizio 3 — instanceof e casting

RICONOSCERE IL TIPO

Scorri un array **Figura[]**: se l'elemento e' un **Cerchio** (usa **instanceof**), fai il cast e stampa il raggio; altrimenti stampa solo l'area.

```
if (f instanceof Cerchio) {  
    Cerchio c = (Cerchio) f;  
    // c.getRaggio()  
}
```

Esercizio 4 — equals tra oggetti

CONFRONTO LOGICO

In **Punto** (x, y) fai l'override di **equals(Object o)** in modo che due punti siano uguali se hanno le stesse coordinate. Verifica con due **Punto(1,2)**.

```
public boolean equals(Object o) {  
    if (!(o instanceof Punto)) return false;  
    Punto p = (Punto) o;  
    // confronta x e y  
}
```

Soluzioni

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

Soluzione 1

```
abstract class Figura { public abstract double area(); }
class Cerchio extends Figura {
    private double r; Cerchio(double r){this.r=r;}
    public double getRaggio(){return r;}
    public double area() { return Math.PI * r * r; }
}
class Quadrato extends Figura {
    private double l; Quadrato(double l){this.l=l;}
    public double area() { return l * l; }
}
```

Soluzione 2

Il tipo della variabile e' Figura, ma a runtime viene eseguito il metodo dell'oggetto reale: questo e' il polimorfismo.

```
Figura[] figure = { new Cerchio(2), new Quadrato(3) };
for (Figura f : figure)
    System.out.printf("Area: %.2f%n", f.area());
```

Soluzione 3

```
for (Figura f : figure) {
    if (f instanceof Cerchio) {
        Cerchio c = (Cerchio) f;
        System.out.println("Cerchio r=" + c.getRaggio());
    } else {
        System.out.println("Area=" + f.area());
    }
}
```

Soluzione 4

```
@Override
public boolean equals(Object o) {
    if (!(o instanceof Punto)) return false;
    Punto p = (Punto) o;
    return this.x == p.x && this.y == p.y;
}
```