

L3 - Incapsulamento - Base

Java · [private](#), [getter/setter](#), [validazione](#), [classe completa](#)

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 — Rendi privato

PRIVATE

Parti dalla classe **Studiante** con attributi pubblici e rendili **private**. Aggiungi i **getter** per leggere nome e media dall'esterno.

```
private String nome;  
public String getNome() { return nome; }
```

Esercizio 2 — Setter con controllo

SETTER & MIDDLE DOT; VALIDAZIONE

Aggiungi **setMedia(double m)** che accetta il valore solo se compreso tra 0 e 10; altrimenti stampa un messaggio di errore e lascia la media invariata.

```
public void setMedia(double m) {  
    if (m >= 0 & & m <= 10) media = m;  
    else // errore  
}
```

Esercizio 3 — Conto incapsulato

STATO PROTETTO

Crea **Conto** con **saldo** privato, metodi **versa** e **preleva** (preleva solo se c'e' saldo) e **getSaldo()**. Il saldo non deve poter diventare negativo dall'esterno.

```
public class Conto {  
    private double saldo;  
    // versa, preleva, getSaldo  
}
```

Esercizio 4 — Validazione nel costruttore

OGGETTO SEMPRE VALIDO

Crea **Prodotto** (nome, prezzo, quantita). Nel costruttore, se la **quantita** ≤ 0 stampa un errore e segna l'oggetto come non valido (boolean valido = false). Aggiungi **valoreTotale()** = prezzo * quantita.

```
public Prodotto(String n, double p, int q) {  
    if (q <= 0) { /* errore, valido=false */ }  
    // ...  
}
```

Soluzioni

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

Soluzione 1

```
public class Studente {
    private String nome; private double media;
    public Studente(String n, double m) { nome=n; media=m; }
    public String getNome() { return nome; }
    public double getMedia() { return media; }
}
```

Soluzione 2

```
public void setMedia(double m) {
    if (m >= 0 && m <= 10) media = m;
    else System.out.println("Media non valida: " + m);
}
```

Soluzione 3

```
public class Conto {
    private double saldo;
    public Conto(double iniziale) { saldo = iniziale; }
    public void versa(double x) { if (x > 0) saldo += x; }
    public void preleva(double x) {
        if (x > 0 && x <= saldo) saldo -= x;
        else System.out.println("Operazione non valida");
    }
    public double getSaldo() { return saldo; }
}
```

Soluzione 4

```
public class Prodotto {
    private String nome; private double prezzo; private int quantita;
    private boolean valido = true;
    public Prodotto(String n, double p, int q) {
        if (q <= 0) { System.out.println("Errore: quantita non valida"); valido = false; return; }
        nome=n; prezzo=p; quantita=q;
    }
    public boolean isValido() { return valido; }
    public double valoreTotale() { return prezzo * quantita; }
}
```