

### Esercizio 2

Implementare la classe `dati.Libro`, con le variabili di istanza protette:

- `titolo` e `isbn` di tipo `String`;
- `prezzo` di tipo `float`

e i metodi:

- Costruttore/i
- Di accesso alle variabili, es:
  - `getPrezzo()`, deve ritornare il prezzo del libro
- `equals(Object o)` e `toString()` (sovrascrivere i metodi di `Object`)

### Esercizio 2

Implementare la classe `dati.Carrello` con una variabile di istanza, `libri`, di tipo collezione di oggetti `Libro`, e i metodi `equals` e `toString`.

Nella classe `Carrello`, implementare i metodi:

- `aggiungiLibro(Libro b)`
- `togliLibro(Libro b)`
- un metodo (`getConto`) che calcoli il prezzo totale dei libri nel carrello

### Esercizio 2

Implementare la classe `dati.LibroScontato`, sottoclasse di `Libro`, che definisca una variabile d'istanza `scontoPercentuale` di tipo intero.

Sovrascrivere il metodo `getPrezzo` di `Libro` in `LibroScontato` in modo che ritorni il prezzo scontato del libro.

Scrivere un programma che utilizzi le tre classi sopra definite (classe `shop.Test`).

### Esercizio 2

Implementare la classe `dati.LibroScontato`, sottoclasse di `Libro`, che definisca una variabile d'istanza `scontoPercentuale` di tipo intero.

Sovrascrivere il metodo `getPrezzo` di `Libro` in `LibroScontato` in modo che ritorni il prezzo scontato del libro.

Scrivere un programma che utilizzi le tre classi sopra definite (classe `shop.Test`).

### Esercizio 3 (x casa)

Modificare l'esercizio in questo modo:

- definire l'interfaccia `Prodotto`
- modificare la classe `Libro` in modo che implementi l'interfaccia `Prodotto` (identificare quali metodi definire in `Libro` e quali in `Prodotto`)
- definire una seconda classe (`Dvd`) che implementi l'interfaccia `Prodotto`
- modificare la classe `Carrello` in modo che vi possano essere aggiunti oggetti `Prodotto` di tipo diverso