

### Esercizio 1.

Si progettino le classi `Libro`, `LibroPerBambini` e `Biblioteca` per la gestione di una biblioteca.

La classe `Libro` ha 3 variabili di istanza `titolo`, `autore`, `inPrestito`. La variabile booleana `inPrestito` indica se il libro è attualmente in prestito. Si inseriscano nella classe `Libro` i seguenti metodi:

- `String getDescrizione()`: restituisce una descrizione contenente il titolo e l'autore del libro;
- `boolean getInPrestito()`: restituisce `true` se il libro è in prestito, `false` altrimenti;
- `void setInPrestito(boolean stato)`: cambia lo stato del libro (in prestito oppure disponibile).

La classe `LibroPerBambini`, sottoclasse di `Libro`, deve contenere una variabile d'istanza `etaConsigliata` ed il metodo `getDescrizione()` deve restituire nella descrizione, oltre al titolo ed all'autore, anche l'età consigliata.

La classe `Biblioteca` gestisce la collezione di libri con una struttura dati di tipo `ArrayList`, che può contenere qualsiasi oggetto di tipo `Libro` e `LibroPerBambini`. La biblioteca implementa i seguenti metodi:

- `void addLibro(Libro l)`: aggiunge un nuovo libro alla biblioteca;
- `boolean inBiblioteca(Libro t)`: restituisce `true` se il libro è presente in biblioteca;
- `void prestato(Libro l)`: consente di dare in prestito un libro;
- `void restituito(Libro l)`: consente di restituire un libro dato in prestito;
- `int totInPrestito()`: conta il numero di libri in prestito;
- `boolean nessunPrestito()`: restituisce `true` se nella biblioteca non vi sono libri in prestito;
- `int perBambini(int n)`: conta quanti libri per bambini di età minore o uguale a `n` sono presenti in biblioteca.

Si scriva una classe di test che crea un libro, un libro per bambini, una biblioteca e chiama tutti i metodi.

### Esercizio 2.

Si riscriva il metodo `void prestato(Libro l)` affinché sollevi l'eccezione `EccezioneLibro` nel caso in cui il prestito non sia possibile (il libro non è presente in biblioteca oppure è già in prestito).

### Esercizio 3.

Si dica cosa stampa il seguente programma, motivando la risposta.

```

public class A {
    private int n;

    public A(int n) {
        this.n=n;
    }

    public int getN() {
        return n;
    }

    public double calcola(int m) {
        return n+m;
    }

    public double calcola(double m) {;
        return n+m+1;
    }
}

public class B extends A {

    public B(int n) {
        super(n);
    }

    public double calcola(int m) {
        return getN() + m + 10;
    }
}

public class C extends B{

    public C(int n) {
        super(n);
    }

    public double calcola(int m) {
        return getN() + m + 100;
    }
}

public class Test {

    public static void main(String[] args) {
        C c = new C(2);
        B b = new B(5);
        A a = b;
        System.out.println(c.calcola(a.calcola(0)));
    }
}

```