

CLASSE SCAFFALE

ESEMPIO

A partire dalle classi *Libro* e *Mensola* possiamo pensare di sviluppare una classe *Scaffale* strutturata come un insieme ordinato di 5 mensole; in sintesi uno scaffale è un oggetto composto da 5 mensole ciascuna delle quali può contenere fino a 15 libri (FIGURA 6).

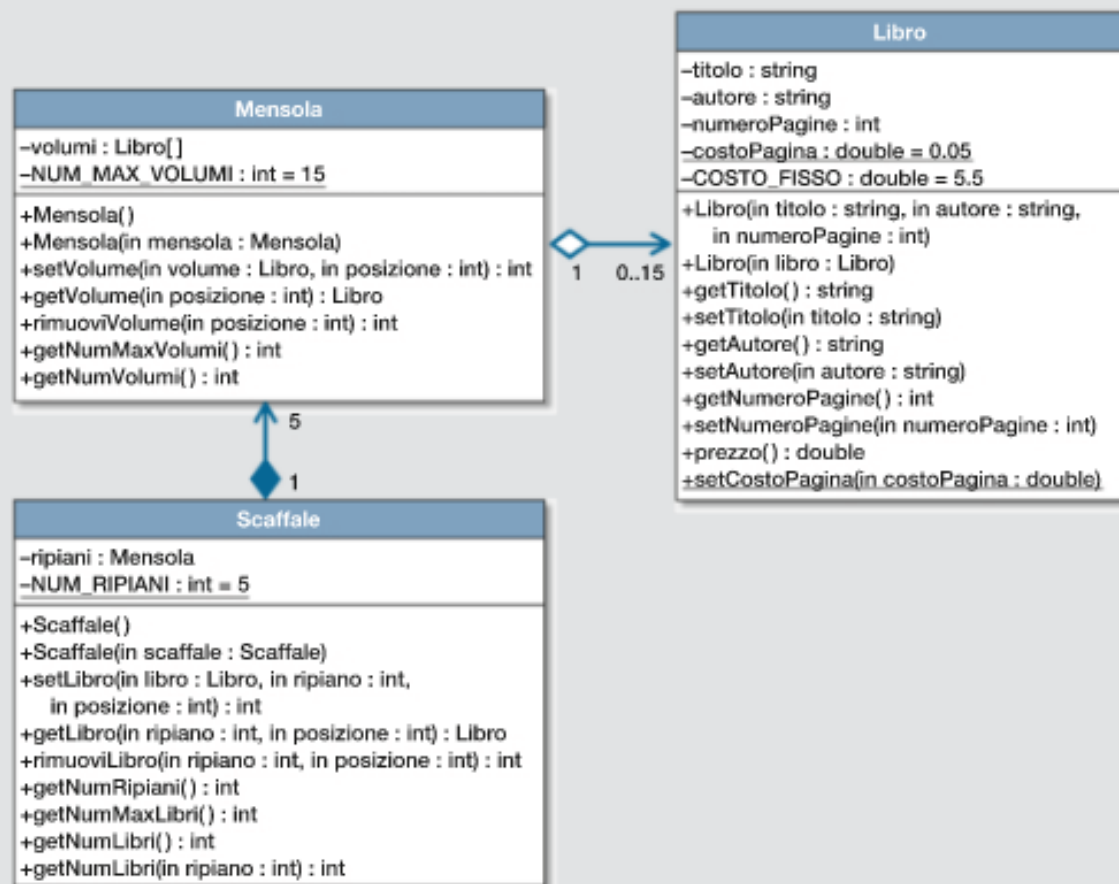


FIGURA 6

Uno scaffale non può esistere senza le mensole che lo costituiscono; per questa ragione l'associazione tra le classi *Scaffale* e *Mensola* è di tipo compositivo. Una possibile implementazione Java della classe *Scaffale* è la seguente:

```
public class Scaffale {
    // attributi
    private static final int NUM_RIPIANI=5;
    private Mensola ripiani[];

    // costruttori
    public Scaffale() {
        int ripiano;

        ripiani=new Mensola[NUM_RIPIANI];
        for (ripiano=0; ripiano<NUM_RIPIANI; ripiano++) {
            ripiani[ripiano]=new Mensola();
        }
    }

    public Scaffale(Scaffale scaffale) {
        int ripiano, posizione;
        Libro libro;

        ripiani=new Mensola[NUM_RIPIANI];
        for (ripiano=0; ripiano<NUM_RIPIANI; ripiano++) {
            ripiani[ripiano]=new Mensola();
            for (posizione=0; posizione<ripiani[ripiano].getNumMaxVolumi(); ripiano++) {
                libro=scaffale.getLibro(ripiano, posizione);
                if (libro!= null)
                    ripiani[ripiano].setVolume(libro, posizione);
            }
        }
    }

    // metodi
    public Libro getLibro(int ripiano, int posizione) {
        if ((ripiano<0)|| (ripiano>=NUM_RIPIANI))
            return null; // ripiano non valido

        return ripiani[ripiano].getVolume(posizione);
    }

    public int setLibro(Libro libro, int ripiano, int posizione) {
        if ((ripiano<0)|| (ripiano>=NUM_RIPIANI))
            return -1; // ripiano non valido

        if (ripiani[ripiano].setVolume(libro, posizione)<0)
            return -2; // posizione nel ripiano non valida o non vuota

        return 1; // inserimento effettuato
    }

    public int rimuoviLibro(int ripiano, int posizione) {
        if ((ripiano<0)|| (ripiano>=NUM_RIPIANI))
            return -1; // ripiano non valido
    }
}
```



```

        if (ripiani[ripiano].rimuoviVolume(posizione)<0)
            return -2; // posizione nel ripiano non valida o vuota

        return 1; // eliminazione effettuata
    }

    public int getNumRipiani() {
        return NUM_RIPIANI;
    }

    public int getNumMaxLibri() {
        int ripiano, n=0;

        for (ripiano=0; ripiano<NUM_RIPIANI; ripiano++) {
            n += ripiani[ripiano].getNumMaxVolumi();
        }
        return n;
    }

    public int getNumLibri() {
        int ripiano, n=0;

        for (ripiano=0; ripiano<NUM_RIPIANI; ripiano++) {
            n += ripiani[ripiano].getNumVolumi();
        }
        return n;
    }

    public int getNumLibri(int ripiano) {
        if ((ripiano<0) || (ripiano>=NUM_RIPIANI))
            return -1; // ripiano non valido

        return ripiani[ripiano].getNumVolumi();
    }
}

```

OSSERVAZIONE Nell'esempio precedente il codice dei costruttori, oltre a inizializzare il vettore *ripiani*, provvede a invocare esplicitamente il costruttore di ogni singola mensola che costituisce un ripiano per evitare che il vettore *ripiani* abbia elementi nulli. Questo è uno dei motivi per cui le specifiche *Javabeans* prevedono che ogni classe abbia un costruttore di default privo di parametri.

In Java gli *array* hanno un attributo specifico, denominato *length*, che indica il numero di elementi; di conseguenza uno qualsiasi dei cicli dell'esempio precedente potrebbe essere così riformulato:

```

for (ripiano=0; ripiano<ripiani.length; ripiano++)
{
    ...
    ...
    ...
}

```

L'esistenza dell'attributo *length* per gli oggetti di tipo *array* consente il passaggio di un *array* come parametro di un metodo senza la necessità di specificarne la dimensione mediante un diverso parametro, com'è solito fare in C/C++.

ESEMPIO

Un test parziale della classe *Scaffale* può essere eseguito mediante il seguente metodo *main*:

```
public static void main (String[] args) {
    Scaffale scaffale = new Scaffale();
    Libro libro;

    // creazione di tre oggetti di tipo Libro
    Libro l1=new Libro("Pinocchio", "C. Collodi", 150);
    Libro l2=new Libro("Pollicino", "C. Perrault", 80);
    Libro l3=new Libro("La bella addormentata nel bosco", "C. Perrault", 50);

    // inserimento mensola #0
    scaffale.setLibro(l1, 0, 10);
    scaffale.setLibro(l2, 0, 0);
    // test inserimento mensola #1
    libro=new Libro("Cappuccetto Rosso", "F.lli Grimm", 150);
    scaffale.setLibro(libro, 1, 1);

    // test errori inserimento
    if (scaffale.setLibro(l3, 10, 0) == -1)
        System.out.println("mensola non valida");
    if (scaffale.setLibro(l3, 0, 20) == -2)
        System.out.println("posizione non valida o non libera");
    if (scaffale.setLibro(l3, 0, 10) == -2)
        System.out.println("posizione non valida o non libera");

    // inserimento mensola #1
    scaffale.setLibro(l3, 1, 0);

    // visualizzazione contenuto mensole
    for (int ripiano=0; ripiano<scaffale.getNumRipiani(); ripiano++) {
        for (int posizione=0; posizione<scaffale.ripiani[ripiano].getNumMaxVolumi();
            posizione++) {
            libro = scaffale.getLibro(ripiano, posizione);
            if (libro != null)
                System.out.println("ripiano: "+ripiano+" posizione: "+posizione+" -> "+
                    libro.getTitolo()+" "+libro.prezzo()+"€");
        }
    }

    // modifica titolo e autore libro estratto da scaffale
    libro=scaffale.getLibro(0,0);
    if (libro != null) {
        libro.setTitolo("Sussi e Biribissi");
        libro.setAutore("Collodi nipote");
    }

    // visualizzazione contenuto mensole
    for (int ripiano=0; ripiano<scaffale.getNumRipiani(); ripiano++) {
```

```

    for (int posizione=0; posizione<scaffale.ripiani[ripiano].getNumMaxVolumi();
        posizione++) {
        libro = scaffale.getLibro(ripiano, posizione);
        if (libro != null)
            System.out.println("ripiano: "+ripiano+" posizione: "+posizione+" -> "+
                               libro.getTitolo()+" "+libro.prezzo()+"€");
        }
    }
}

```

il cui output è:

```

mensola non valida
posizione non valida o non libera
posizione non valida o non libera
ripiano: 0 posizione: 0 -> Pollicino 9.5€
ripiano: 0 posizione: 10 -> Pinocchio 13.0€
ripiano: 1 posizione: 0 -> La bella addormentata nel bosco 8.0€
ripiano: 1 posizione: 1 -> Cappuccetto Rosso 13.0€
ripiano: 0 posizione: 0 -> Sussi e Biribissi 9.5€
ripiano: 0 posizione: 10 -> Pinocchio 13.0€
ripiano: 1 posizione: 0 -> La bella addormentata nel bosco 8.0€
ripiano: 1 posizione: 1 -> Cappuccetto Rosso 13.0€

```

Nell'esempio precedente gli attributi dei libri contenuti in un oggetto di tipo *Scaffale* possono essere alterati modificando gli attributi di un oggetto restituito dal metodo *getLibro*. Questo comportamento è conseguenza del fatto che il metodo *getLibro* restituisce un riferimento allo stesso oggetto di classe *Libro* riferito dall'oggetto *Scaffale* stesso e non è il comportamento atteso dal programmatore: rappresenta una situazione che, come vedremo più avanti, dovrebbe essere evitata.