

# Un Sistema per Prestiti di Libri

- ▶ Descrizione sintetica dei requisiti
  - ▶ L'obiettivo è memorizzare i prestiti di libri, in modo da permettere lo scambio di libri solo fra persone che sono iscritte ad un'associazione
  - ▶ Vincoli: ogni persona (riconosciuta) può prendere in prestito un solo libro alla volta; per ogni libro è presente un'unica copia
- ▶ Descrizione della progettazione
  - ▶ Si individuano le classi: **Socio**, **Libro** e **Prestiti**
  - ▶ Le classi Socio e Libro servono a tenere i dati di persone e libri
  - ▶ La classe Prestiti contiene la tabella in cui sono memorizzati i prestiti, come coppie di istanze di Socio e Libro. La classe Prestiti implementa le operazioni: registrazione prestito, e restituzione libro

Prof. Tramontana - Giugno 2022

```
import java.util.HashMap;
import java.util.Map;

/** Prestiti tiene la tabella dei libri prestati */
public class Prestiti {
    private final Map<Socio, Libro> prestiti = new HashMap<>();

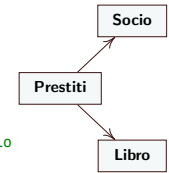
    /**
     * Verifica e registra il prestito di un libro. Un socio può avere un solo
     * prestito alla volta. Ogni libro è disponibile in un'unica copia
     */
    public boolean registra(Socio socio, Libro libro) {
        if (richiestaNonValida(socio, libro))
            return false;
        prestiti.put(socio, libro);
        return true;
    }

    private boolean richiestaNonValida(Socio socio, Libro libro) {
        return socio == null || libro == null ||
            prestiti.containsKey(socio) || prestiti.containsValue(libro);
    }

    /** Rimuove la registrazione, se la trova in tabella */
    public boolean restituisci(Socio socio, Libro libro) {
        return prestiti.remove(socio, libro);
    }
}

public record Libro(String autore, String titolo) { }

public record Socio(String idCode, String name, String telNumber) { }
```



Prof. Tramontana - Giugno 2022

## Progettazione di Test

- ▶ Provare a registrare un prestito a un socio. Risultato: la registrazione deve andare a buon fine
- ▶ Provare a registrare due prestiti allo stesso socio. Risultato: la registrazione del secondo libro deve essere bloccata
- ▶ Provare a registrare due prestiti sullo stesso libro a due soci. Risultato: la registrazione del secondo libro deve essere bloccata
- ▶ Provare a registrare un prestito a un socio null. Risultato: la registrazione deve essere bloccata
- ▶ Provare a registrare un prestito di un libro null. Risultato: la registrazione deve essere bloccata

Prof. Tramontana - Giugno 2022

```
public class TestPrestiti {
    private Libro libro1, libro2;
    private Socio socio1, socio2;
    private Prestiti prest;

    public void beforeEach() {
        libro1 = new Libro("I. Sommerville", "Ingegneria del Software");
        libro2 = new Libro("E. Gamma", "Design Pattern");
        socio1 = new Socio("ABC", "Alan", "33445566");
        socio2 = new Socio("DEF", "Daniel", "556688");
        prest = new Prestiti();
    }

    public void testRegistraUno() {
        assertTrue(prest.registra(socio1, libro1), "registra un prestito");
    }

    public void testRegistraDueLibriStessoSocio() {
        prest.registra(socio1, libro1);
        assertFalse(prest.registra(socio1, libro2), "blocca due prestiti");
    }

    public void testRegistraDueSociStessoLibro() {
        prest.registra(socio1, libro1);
        assertFalse(prest.registra(socio2, libro1), "blocca libro già prestato");
    }

    public void testRegistraNull() {
        assertFalse(prest.registra(null, libro1), "blocca prestito a null");
    }
}

// continua ...
```

Prof. Tramontana - Giugno 2022

```
// continua classe TestPrestiti

public void assertTrue(boolean result, String msg) {
    if (result) System.out.print("OK ");
    else System.out.print("FAIL ");
    System.out.println(msg);
}

public void assertFalse(boolean result, String msg) {
    if (!result) System.out.print("OK ");
    else System.out.print("FAIL ");
    System.out.println(msg);
}

public static void main(String[] args) {
    TestPrestiti test = new TestPrestiti();
    test.beforeEach();
    test.testRegistraUno();
    test.beforeEach();
    test.testRegistraDueLibriStessoSocio();
    test.beforeEach();
    test.testRegistraDueSociStessoLibro();
    test.beforeEach();
    test.testRegistraNull();
}
}
```