



Design Pattern Facade

// Codice Java che implementa il design pattern Facade

```

import java.util.Arrays;
import java.util.List;

/**
 * Permessi implementa le regole per la convalida dei prestiti e delle letture c
 * Permessi e' una classe del sottosistema
 */

public class Permessi {
    // autorizzati alla lettura
    private final List<String> lettura = Arrays.asList("Alan", "Brian", "Clair");
    // autorizzati ai prestiti
    private final List<String> prestito = Arrays.asList("Alan", "Brian");

    public boolean seLeggibile(String nomePers) {
        System.out.println("Permessi: verifica regola lettura");
        return lettura.contains(nomePers);
    }

    public boolean sePrestabile(String nomePers) {
        System.out.println("Permessi: verifica regole prestito");
    }
  
```

```
        return prestito.contains(nomePers);
    }
}

import java.util.Arrays;
import java.util.List;

/**
 * Socio verifica se un utente e' registrato. Socio e' una classe del sottosistema
 */
public class Socio {
    private final List< String > listaSoci = Arrays.asList("Alan", "Dan");

    public boolean verificaId(String nome) {
        System.out.println("Socio: in controlla id");
        return listaSoci.contains(nome);
    }
}

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Prestiti tiene una lista dei libri prestati, e' una classe del sottosistema
 */
public class Prestiti {
    // tabPrestiti e' la tabella dei prestiti: la chiave e' il nome della
    // persona, il valore e' la lista dei titoli dei libri presi in prestito
    private Map< String, List< String >> tabPrestiti = new HashMap< >();

    public void registra(String nome, String libro) {
        System.out.println("Prestiti: prendi libro");
        if (tabPrestiti.containsKey(nome))
            tabPrestiti.get(nome).add(libro);
        else {
            List< String > lis = new ArrayList< >();
            lis.add(libro);
            tabPrestiti.put(nome, lis);
        }
    }

    public String trovaPersona(String libro) {
        System.out.println("Prestiti: trova persona");
        for (String s : tabPrestiti.keySet())
            if (tabPrestiti.get(s).contains(libro)) return s;
        return null;
    }
}
```

```
/** Libro e' una classe del sistema */

public class Libro {
    private String titolo;

    public Libro(int x) {
        if (x == 1) titolo = "Curious George goes fishing";
        else titolo = "Curious George at the zoo";
    }

    public String getTitolo() {
        System.out.println("Libro: titolo");
        return titolo;
    }

    public String getAutore() {
        return "A. H. Rey";
    }
}

/**
 * Registro implementa il ruolo Facade per il sottosistema formato da Prestiti,
 * Permessi e Socio. Registro verifica se una certa azione su libro (presta o le
 * e' autorizzata e registra i dati rilevanti
 */

public class Registro {
    private final Permessi perm = new Permessi();
    private final Socio so = new Socio();
    private static Prestiti prt = new Prestiti();

    // notare che prestaLibro() chiama i metodi delle classi del sottosistema
    public boolean prestaLibro(String nom, Libro lib) {
        System.out.println("Registro: presta");
        if (so.verificaId(nom) && perm.sePrestabile(nom)) {
            System.out.println("Registro: prestabile");
            prt.registra(nom, lib.getTitolo());
            return true;
        }
        return false;
    }

    public boolean leggiLibro(Libro lib) {
        System.out.println("Registro: leggi");
        final String nomePers = prt.trovaPersona(lib.getTitolo());
        System.out.println("Registro: persona che ha in prestito il libro " + nomePers);
        return (so.verificaId(nomePers) && perm.seLeggibile(nomePers));
    }
}
```

```
/** Client e' una classe che invoca i metodi del Facade Registro */

public class Client {
    private static Libro l1 = new Libro(1);
    private static Libro l2 = new Libro(2);
    private static String nome = "Alan";
    private static Registro r = new Registro();

    public static void main(String[] args) {
        presta(l1);
        presta(l2);
        leggi(l1);
    }

    public static void presta(Libro lib) {
        if (r.prestaLibro(nome, lib))
            System.out.println("Prestito registrato");
    }

    public static void leggi(Libro lib) {
        if (r.leggiLibro(lib))
            System.out.println(nome+" legge "+lib.getTitolo());
    }
}
```

Output dell'esecuzione

```
Registro: presta
Socio: in controlla id
Permessi: verifica regole prestito
Registro: prestabile
Libro: titolo
Prestiti: prendi libro
Prestito registrato
```

```
Registro: presta
Socio: in controlla id
Permessi: verifica regole prestito
Registro: prestabile
Libro: titolo
Prestiti: prendi libro
Prestito registrato
```

```
Registro: leggi
Libro: titolo
Prestiti: trova persona
Registro: persona che ha in prestito il libro Alan
Socio: in controlla id
Permessi: verifica regola lettura
```

Libro: titolo

Alan legge Curious George goes fishing