

Gestione Spiaggia

Gianluca Grossi

29 maggio 2015

Sommario

Gestione Spiaggia è un software open source scritto in Java con licenza
GPL with linking exception.

L'applicazione è stata principalmente realizzata per il corso di
Programmazione ad Oggetti a.a. 2014/2015 della facoltà di Ingegneria e
Scienza Informatiche di Cesena.

Indice

1 Analisi

1.1 Requisiti

1.2 Problema

2 Design

2.1 Architettura

2.2 Design dettagliato

3 Sviluppo

3.1 Testing automatizzato

3.2 Divisione dei compiti e metodologia di lavoro

4 Commenti finali

4.1 Conclusioni e lavori futuri

Capitolo 1

Analisi

1.1 Requisiti

Il software che si intende sviluppare mira a gestire il lavoro di una spiaggia privata tramite l'utilizzo di interfaccia grafica. Con essa sarà possibile gestire gli ombrelloni presenti nella spiaggia e le loro relative prenotazioni da parte dei clienti.

Inizialmente sarà possibile impostare le caratteristiche principali della spiaggia, ovvero il numero e la disposizione degli ombrelloni, la presenza di una o più passerelle e i vari prezzi della spiaggia (singolo lettino, ombrelloni stagionali e non, fino a un massimo di tre lettini a ombrellone, e prezzo da aggiungere per un ombrellone vinico alla riva o alla passerella).

Il programma principalmente permette di prenotare gli ombrelloni da parte dei clienti per una o più giornate: vengono utilizzati diversi colori per evidenziare ombrelloni momentaneamente liberi, occupati e stagionali.

Il programma, inoltre, permette di vedere, oltre ai dati di tutti gli ombrelloni e dei vari loro clienti, i guadagni della spiaggia, nel corso del tempo, ricevuti tramite ombrelloni e/o lettini.

Infine è possibile effettuare altri comandi come ricercare un ombrellone tramite i dati del cliente (nome e cognome), cambiare i vari prezzi della spiaggia (utile per il cambio tra bassa e alta stagione) e rendere possibile il cambio di ombrellone, con uno libero, da parte di un cliente.

1.2 Problema

La difficoltà principale sarà effettuare l'implementazione dello scambio di ombrellone da parte di un cliente. In questo caso occorre tenere d'occhio diversi problemi. Il pagamento che deve effettuare il cliente può essere diverso tra i due ombrelloni, per via di privilegi che il secondo può avere o meno rispetto al primo (sono presenti prezzi maggiori per ombrelloni vicino alla riva e/o passerella). Inoltre occorre gestire in modo corretto le liste di ombrelloni trasferendo l'ombrellone senza creare errori.

Il programma in questione inoltre contiene diversi tipi di problemi non risolvibili nel monte ore, i quali non lo rendono ancora del tutto utilizzabile a pieno, senza una corretta soluzione. I problemi principali del programma sono i seguenti:

- Non permette il salvataggio dello stato della spiaggia, quindi il programma per mantenere i dati non dovrebbe essere mai chiuso;
- Non è possibile prenotare gli ombrelloni in giornate future, ma solo dal giorno stesso della prenotazione in poi;
- I prezzi che vengono utilizzati sono semplicemente il prodotto del prezzo giornaliero con il numero di giorni di permanenza, invece spesso vengono utilizzati listini con prezzi che scalano in base al numero di giorni;
- Viene utilizzato uno schema di ombrelloni rettangolare, mentre spesso le spiagge hanno una disposizione di ombrelloni variabile.

Capitolo 2

Design

2.1 Architettura

L'applicazione è basata sul pattern architetturale Model-View-Controller. Nella Figura 2.1 viene mostrato lo schema MVC, nel quale si è cercato di separare al meglio i tre componenti di questa architettura.

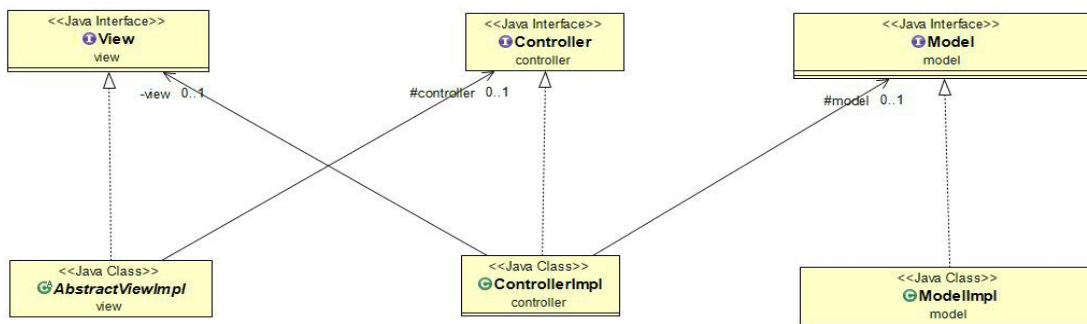


Figura 2.1: Schema UML delle relazioni tra Model-View-Controller

2.2 Design Dettagliato

Il problema del cambio di ombrellone da parte di un cliente viene gestito nel seguente modo. Il cliente, che desidera cambiare postazione, viene fatto partire e si aggiunge alla lista dei vecchi ombrelloni, contando il suo pagamento fino a quel punto. Poi vengono salvati i suoi dati e quelli dell'ombrellone vecchio in quello nuovo e viene calcolato il prezzo da pagare come se il cliente fosse arrivato lo stesso giorno. Diversamente per gli ombrelloni stagionali, il cliente se effettua un cambio di postazione, dovrà pagare la somma più alta tra le due posizioni di ombrellone.

Il software è costituito da tre package:

- model:

La figura 2.2 mostra un UML che rappresenta come è strutturato il model dell'applicazione, mentre la figura 2.3 mostra in modo più dettagliato l'interfaccia Umbrella e le sue relazioni con le altre entità.

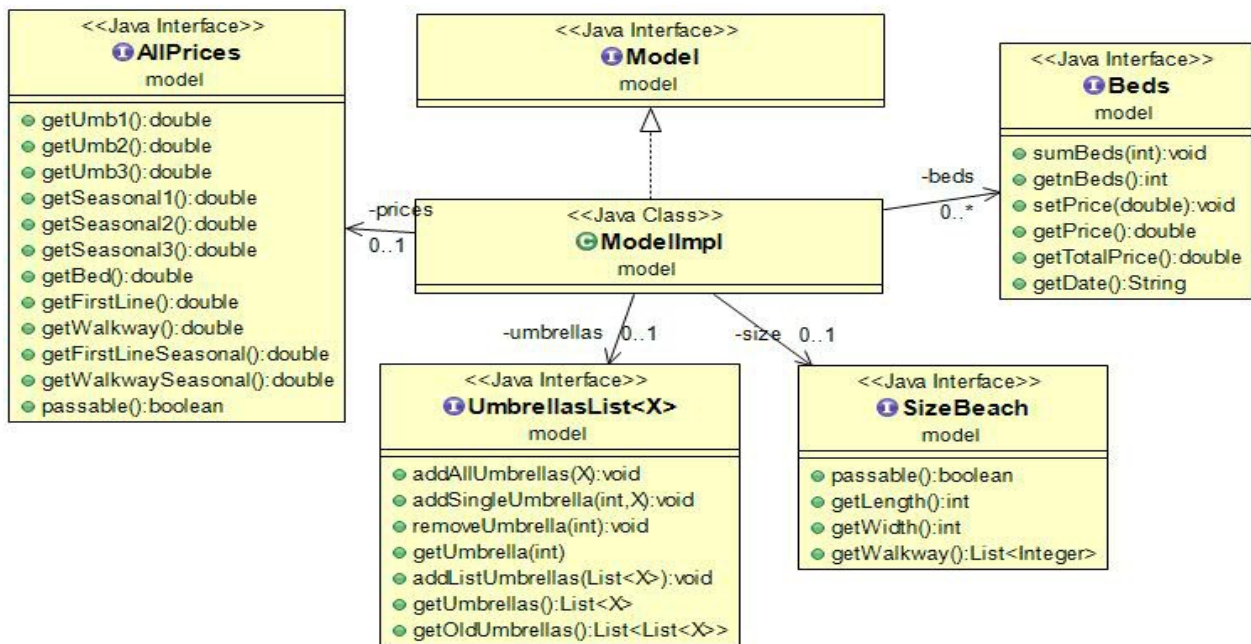


Figura 2.2: Schema UML delle entità principali del package model

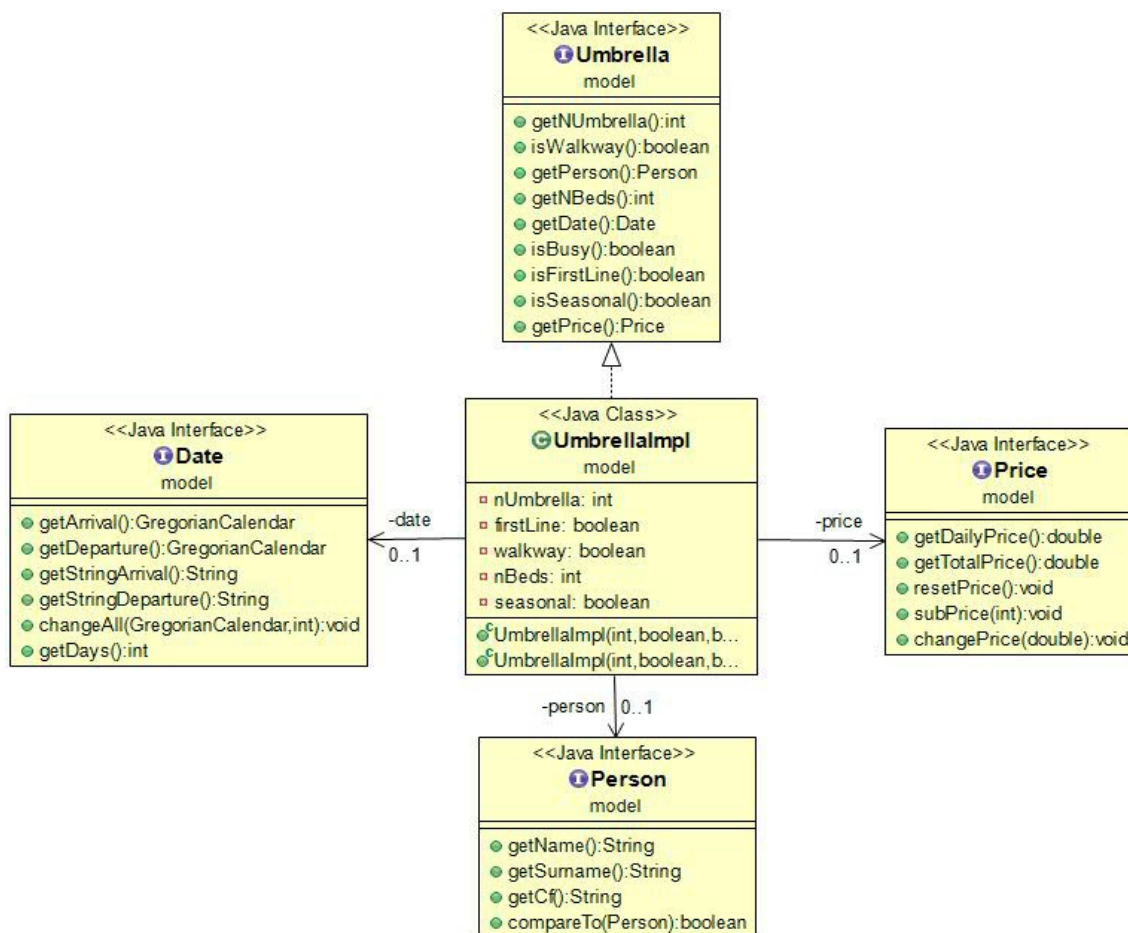


Figura 2.2: Schema UML dell'interfaccia Umbrella

- view:

Nella figura 2.4 vengono mostrate le entità principali del package view. E' stata creata l'interfaccia principale View implementata dalla classe AbstractViewImpl. Quest'ultima contiene un metodo astratto (template method) implementato poi dalle sottoclassi.

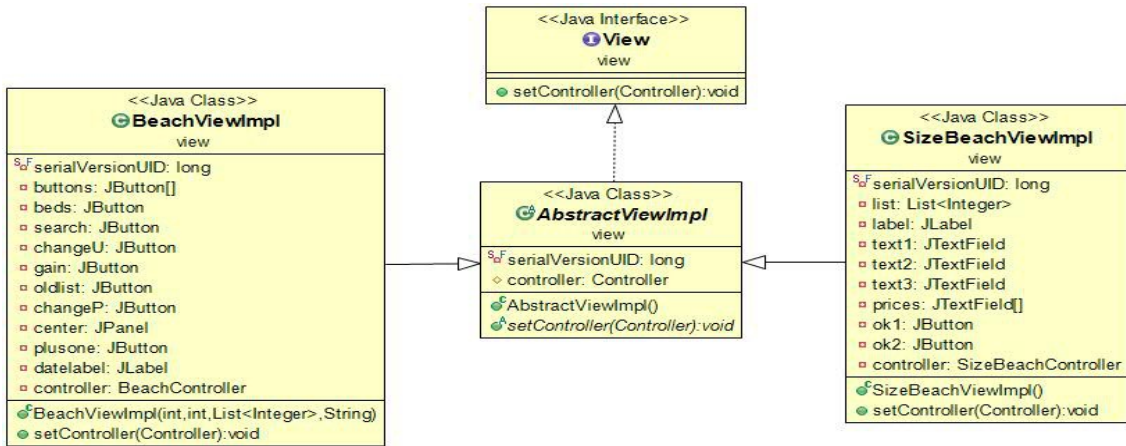


Figura 2.4: Schema UML delle entità principali del package view

- controller

Nella figura 2.5 vengono mostrate le entità del package controller. L'interfaccia Controller viene implementata da ControllerImpl (contenente metodi di setView e setModel) la quale viene estesa dalle sottoclassi.

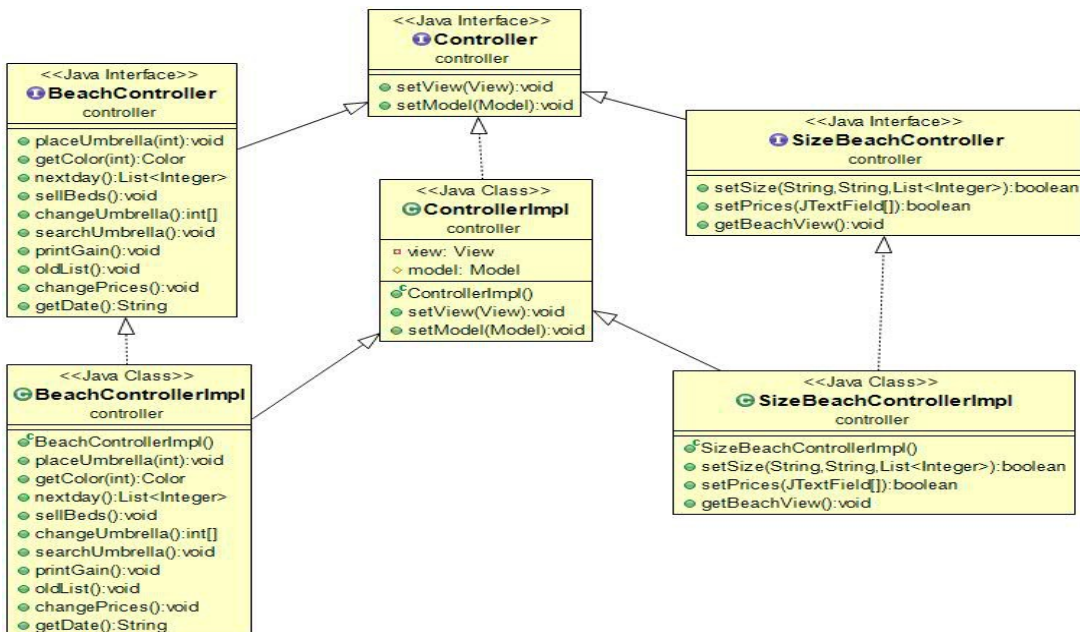


Figura 2.5: Schema UML delle entità del package controller

Capitolo 3

Sviluppo

3.1 Testing automatizzato

Sulla parte riguardante il modello si è provveduto a realizzare una classe di test automatizzata, tramite la suite di JUnit.

In particolare si è controllato che il Model dell'applicazione funzionasse correttamente. Difatti vengono controllati tramite la classe TestModel i suoi diversi metodi, verificando che il programma funzioni in maniera corretta.

3.2 Divisione dei compiti e metodologia di lavoro

L'applicazione è stata implementata esclusivamente dal sottoscritto ed è presente nel seguente repository Bitbucket:

<https://bitbucket.org/gianlucag22/oop14-gestione-spiaggia>

Capitolo 4

Commenti finali

4.1 Conclusioni e lavori futuri

Questo software, secondo il mio punto di vista, non ha una complessità elevata per essere realizzato.

Il programma è stato prodotto principalmente come attività didattica per il progetto del corso di Programmazione ad Oggetti a.a. 2014/2015 della facoltà di Ingegneria e Scienza Informatiche di Cesena. Momentaneamente quindi non si prevedono sviluppi futuri.