Relazione Camping Assistant

Mattia Pini

Maichol Dadi

Michele Capicchioni

28 Febbraio 2016

Indice

Capitolo 1

Analisi

1.1 Requisiti

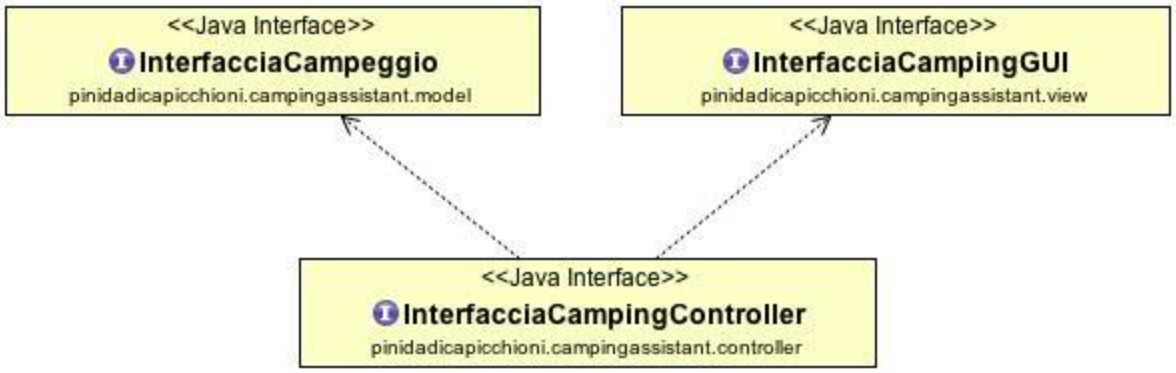
Il software “Camping Assistant” permette di gestire un campeggio in maniera semplificata.
Il programma offre le seguenti feature:

- Creazione di un nuovo campeggio
- Creazione della mappa del campeggio per visualizzare le piazzole attraverso il Drag n Drop, ovvero la possibilità di spostare un oggetto all’interno di uno spazio prestabilito
- Modifica della mappa del campeggio
- Aggiunta e rimozione di clienti
- Aggiunta,rimozione e modifica di dipendenti
- Aggiunta,rimozione,modifica di attività e associazione di una o piu’ attivita’ ai clienti
- Visualizzazione di grafici e statistiche relative alla gestione del campeggio
- Salvataggio e caricamento del campeggio
- Esportazione dei dati in formato csv ed xls

1.2 Analisi e modello del dominio

All’ avvio del programma, l’utente dovrà creare un nuovo campeggio ,specificando inizialmente,
il numero e la posizione delle piazzole che si andranno a creare.
L’utente avrà la possibilità di scegliere il tipo di alloggio che si vorrà andare a creare (Bungalov, Tenda, Roulotte) e potrà inserire tutte le informazioni relative ai clienti che ci alloggeranno, se presenti; ovviamente si dovrà poter modificare la posizione ed il numero delle piazzole create.
L’utente inserirà, se necessario, i dipendenti che lavorano all’interno del camping e se previste, delle attività da svolgere.
Il software potrà esportare i dati relativi al campeggio in formato csv ed excel, ulteriormente si dovrà poter salvare e caricare il campeggio creato in precedenza.

In fine, si dovrà mostrare vari grafici riguardanti le informazioni utili al campeggio (Clienti_entrati, Bilancio, età_media ed le attività praticate).
Il programma si proponeva anche di gestire le prenotazioni dei clienti in modo da occupare le piazzole anche se non fisicamente presente, ma a causa della mancanza di ore, non è stato possibile realizzarlo, e verrà implementato in futuro.
Per quanto riguarda la compatibilità con piattaforme specifiche, è stato riscontrato che le istruzioni per il cambio di colore delle piazzole non sono funzionanti su Mac, mentre funzionano correttamente sia su Windows che su Linux.



Capitolo 2

Design

2.1 Architettura

Design

Il software è stato organizzato seguendo il pattern MVC.
Le richieste della view al model passano attraverso il controller, e gli eventuali risultati vengono riportati da quest'ultimo di nuovo alla view.
Tuttavia l'MVC non è stato implementato in maniera corretta perchè sostituire la GUI comporta molte modifiche al controller.

Model (Maichol Dadi)

Nel model vengono create tutte le funzionalità del programma e delle varie entità che lo compongono, gestendo tutte le varie interazioni che dovranno avere riuscendo a creare del codice estensibile per versioni future.

View (Mattia Pini)

Si occupa della visualizzazione degli elementi e di passare al controller i dati inseriti dall'utente. La GUI visualizzerà tutto ciò che concerne con la gestione di un campeggio: clienti,dipendenti,statistiche ecc ecc

Controller (Michele Capicchioni)

Il controller si occupa di coordinare le azioni della view con le interfacce del model in modo da trasferire i dati tra uno e l'altro e coordinare le operazioni.

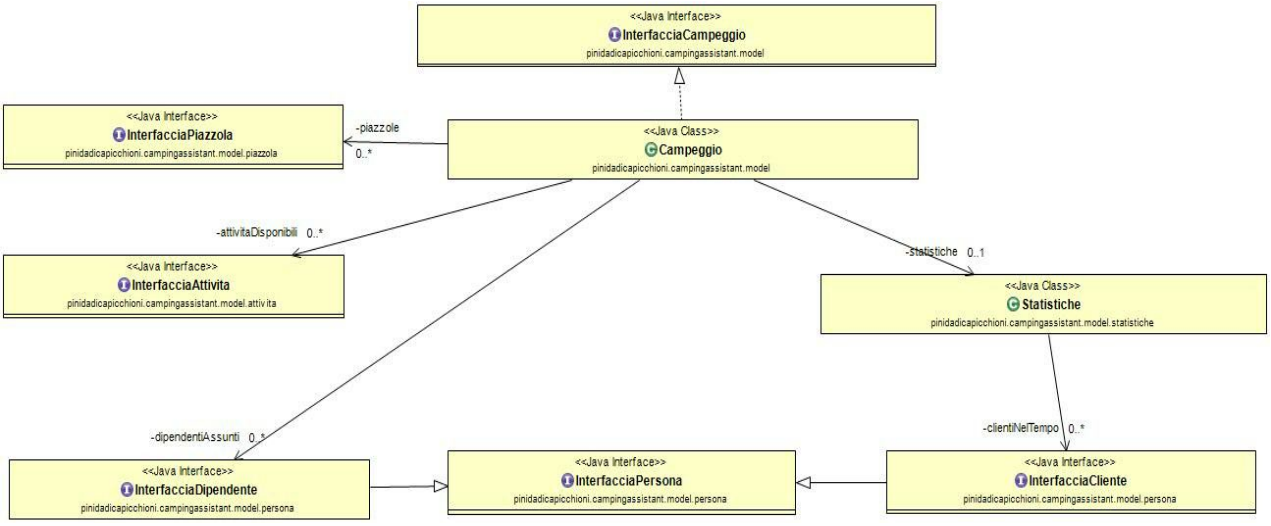
2.2 Design dettagliato

Model (Maichol Dadi)

Nel model si è cercato di creare ogni entita' del campeggio in maniera più chiara possibile per l'utilizzo da parte del controller mediante l'uso di interfacce, infatti ogni classe principale ne

possiede una , grazie alla quale è possibile far corrispondere ogni oggetto creato al proprio scenario di utilizzo.

Si è voluto utilizzare un approccio “Top-Down” per la risoluzione dei problemi incontrati, ovvero, nel model si è voluto dividere ogni macro problema in molti piu piccoli, rendendo cosi il codice piùfacile da controllare e debuggare in cerca di eventuali errori.

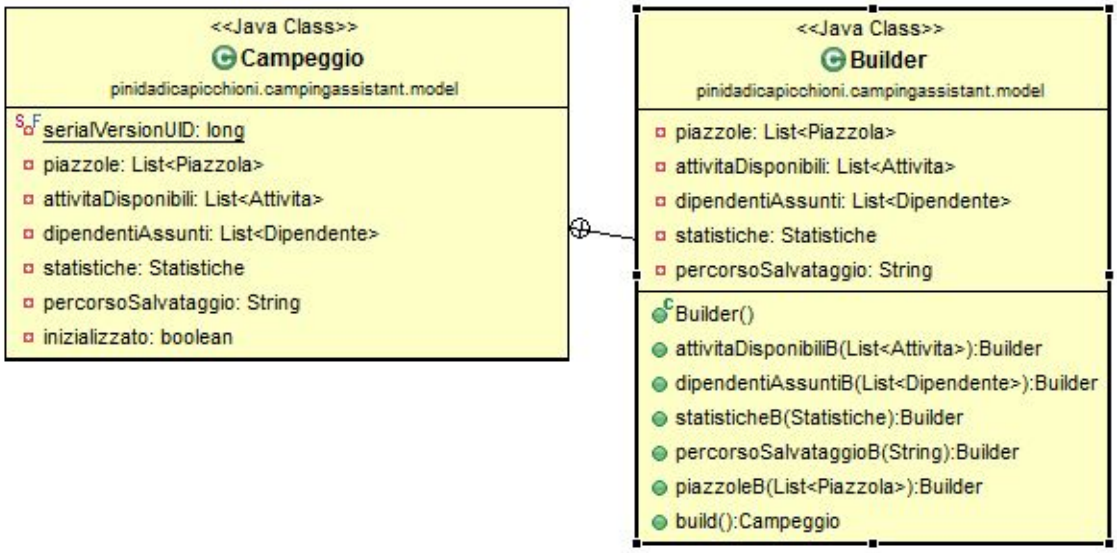


Schema UML relativo alla gestione del campeggio

Pattern utilizzati :

Builder

Si è voluto utilizzare il pattern builder in Campeggio,per un‘ eventuale estensione futura,nel caso in cui si volesse creare un oggetto campeggio, passando dei parametri per la sua costruzione



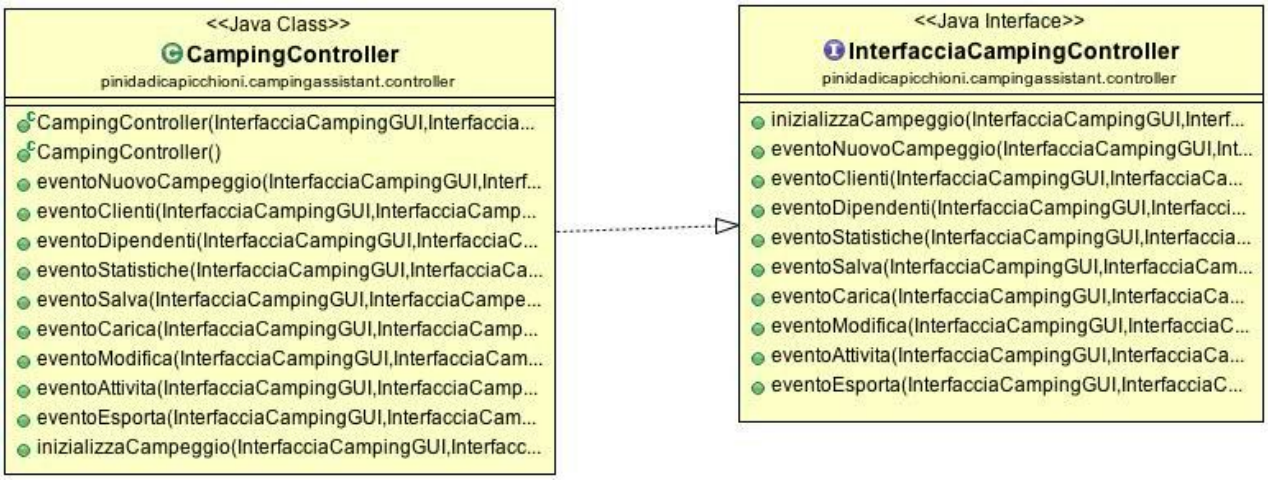
UML relativo all'implementazione del pattern **Builder**

Decorator

Pattern utilizzato in esporta csv(clienti e dipendenti) per scrivere un buffer alla volta, migliorando cosi l'accesso ai file e aumentando le prestazioni.
utilizzo:BufferedOutputStream.

Controller: Il controller si occupa di gestire qualsiasi comunicazione tra view e model, che non vengono mai in contatto fra di loro.
Il controller si occupa di creare istanze degli oggetti principali del model e della view, per poi passarle ai suoi metodi che si occupano di associare i vari bottoni della GUI gli eventi che corretti, prendere dalla stessa i vari dati attraverso le sue interfacce, passare questi dati ai metodi contenuti nelle interfacce del model e infine restituire questi risultati di nuovo alla view per mostrarli all'utente.

Il controller è stato suddiviso su più classi, ognuna con la sua interfaccia, ognuna delle quali si occupa di una determinata situazione.



View: La view si occupa, attraverso diverse interfacce grafice, di gestire un campeggio. Nel frame principale che viene aperto al lancio dell'applicazione, che è CampingGUI, vi è una serie di menu' a tendina, ai quale corrisponde un frame per ogni voce. In questi menu' ci sono tutti gli strumenti per gestire un campeggio. La libreria usata per costruire le interfaccie grafiche è Swing.

Ogni classe della view ha una sua interfaccia con i metodi che il controller utilizzerà per ricevere i dati inseriti in input e aggiornare i vari componenti.

Tutti i componenti dei vari frame sono stati posizionati usando il layout GridBagLayout (il quale è stato studiato approfonditamente prima di essere usato) che permette di posizionare gli elementi in modo ordinato.

In fase di progettazione si è deciso di iniziare dalla GUI principale per poi aggiungere mano a mano i frame successivi.

All'avvio la GUI mostrerà un messaggio al centro del frame; quando creato un nuovo campeggio o caricato uno esistente, la GUI si modificherà mostrando a destra la mappa del campeggio e a sinistra le relative informazioni sulla piazzola selezionata.

Particolare attenzione sulla creazione del Nuovo campeggio, che ci permetterà di creare una mappa "personalizzata" del campeggio, permettendoci di spostare le piazzole a nostro piacimento all'interno della mappa(stessa cosa nel Modifica campeggio).

Capitolo 3

Sviluppo

3.1 Testing automatizzato

Il testing del model è situato in “model.TestCampeggio” eseguito mediante JUnit.
Qui si sono testate tutte le funzioni principali e più “problematiche” per il software, ogni test è separato da una riga di commento che indica quali funzioni vengono testate.

3.1 Sviluppo

Inizialmente ci siamo incontrati per concordarci sulla divisione dei ruoli e sulle funzionalità principali da gestire per il nostro gestionale.
La suddivisione dei ruoli è avvenuta come concordato precedentemente:

- M.Pini : View
- M.Capicchioni: Controller
- M.Dadi: Model

Durante il progetto siamo sempre rimasti in contatto sia attraverso il DVCS BitBucket che attraverso i messaggi di commit riuscivamo a capire quali modifiche erano state effettuate volta per volta.
Oltre a rimanere in contatto attraverso BitBucket, ci sentivamo anche attraverso altri canali di comunicazione come Skype e Whatsapp. Il lavoro è stato svolto quasi del tutto autonomamente, poi solo successivamente ci siamo coordinati per risolvere i vari bug e le varie mancanze al progetto.

3.3 Note di Sviluppo

Il nostro programma utilizza le seguenti librerie esterne :
JFreechart: utilizzato per la stampa dei grafici
JCalendar: utilizzato per la scelta della data di arrivo/partenza del cliente
ApachePoi: utilizzato per l’esportazione dei dati in Excel
Inoltre non sono mancati vari spunti presi da Internet per l’implementazione della View, come il consiglio di usare un GridBagLayout(Mattia Pini)

Un altro importante aiuto è venuto da questo link:
http://www.codeproject.com/Articles/116088/Draggable-Components-in-Java-Swing
che mi ha permesso di implementare il Drag n Drop con estrema semplicità.

Capitolo 4

Commenti finali

4.1 Autovalutazione e lavori futuri

Non siamo soddisfatti per niente del nostro progetto, soprattutto per quanto riguarda la parte del controller e della view, perchè inizialmente non avevamo ben capito come implementare l'architettura MVC e seguendo una guida molto votata su internet, ci ha fatto creare dipendenze tra il model e il controller.
Purtroppo ci siamo accorti troppo tardi di questo errore perchè abbiamo dato maggiore importanza a far funzionare il programma piuttosto di pensare a un pattern adeguato per collegare view e model e il monte ore era già stato superato.

Maichol Dadi
Si è cercato di rispettare l'information hiding al meglio, facendo vedere al controller solo i metodi necessari grazie all'utilizzi di interfacce.
A causa del poco tempo assegnatoci non sono riuscito a evitare il problema della "Sindrome dell intoccabilità" rendendo cosi il codice con delle dipendenze le cui modifiche potrebbero risultare troppo "costose".
Non sono soddisfatto del mio lavoro, causa lavoro e esami, credo che avremmo dovuto cambiare deadline, non rendendoci effettivamente conto del carico di lavoro aspettato.
Mi dispiace di non aver potuto approfondire tutti i pattern di progettazione ma abbiamo deciso di utilizzare il poco tempo rimasto per aggiungere funzionalità, correggere bug e rendere il programma il più completo possibile.

Mattia Pini
Non sono per niente soddisfatto del lavoro fatto, perché ci siamo trovati alle ultime 3 settimane dalla scadenza a fare praticamente tutto il progetto causa esami arretrati e non.
Sono solo un po soddisfatto del fatto di aver imparato il layout GridBagLayuot e del Drag n Drop.

Volevo implementare il pattern Observer, ma il tempo e la scelta iniziale sbagliata, non mi hanno permesso di farlo.
Purtroppo non sono riuscito a implementare i bordi del Drag n Drop causa limite ore superato.
Inoltre, c'è stato una discussione con i miei compagni facendo una scelta illogica, ovvero quella di non distinguere i Bungalow dalle Piazzole.
Nonostante questo screzio, il lavoro con i compagni è andato bene coordinandoci sulle varie modifiche e migliorie, lavorando simultaneamente sulle nostre rispettive parti.

Michele Capicchioni
Anche se il lavoro non è stato fatto nel migliore dei modi, dato che si sono create forti dipendeze tra le varie componenti, questo mi ha permesso di imparare molte cose.
Avrei voluto gestire in maniera diversa la gestione della view, e ho notato che a volte il controller entra troppo nel merito del model.
Purtroppo a causa del monte ore raggiunto e della deadline sempre più vicina questo non è stato possibile, ma ho dato la precedenza al corretto funzionamento di ciò che già era stato implementato.

4.2 Difficolta incontrate e commenti per i docenti

Tra le difficoltà riscontrate, per prima viene il corretto utilizzo di MVC, infatti inizialmente non è stato facile capire come gestire le varie interfacce.
Inoltre c'è stata anche una certa difficoltà nel coordinare le scelte su come portare avanti il progetto.
Infine è stata riscontrata una certa difficoltà nel determinare i pattern utilizzati.

Appendice A

Guida utente

Questo software permette di gestire un campeggio, ed espone tutte le funzionalità per farlo al meglio. All' avvio verrà mostrato un messaggio che vi dirà che dovrete creare un nuovo campeggio o caricarne uno già esistente. Per crearne uno nuovo basta cliccare su

- File -> Nuovo Campeggio

e verrà mostrata una nuova finestra dove le farà inserire le piazzole.
Per fare ciò basta inserire nell'apposito spazio un ID della piazzola e cliccare poi su Aggiungi Piazzola. Verrà creata una piazzola e avrete modo di spostarla all'interno della mappa.
Se create più piazzole alla volta, si posizioneranno una sopra l' altra, ma basta poi spostarle ed il gioco è fatto!
Una volta create tutte le piazzole, cliccare sul bottone SALVA CAMPEGGIO in fondo alla finestra altrimenti dovrete rifare tutto.
Una volta creata la mappa del vostro campeggio, avrete la possibilità di gestirlo appieno ovvero aggiungendo dipendenti, attività e clienti, associare clienti ad attività, vedere il bilancio del proprio campeggio e molto altro attraverso delle finestre molto semplici.
Una volta creato il campeggio, verranno caricati la mappa sulla destra e le informazioni della piazzola sulla sinistra che mostrerà se è occupata e da chi.
Per vedere le informazioni della piazzola basta posizionarsi sopra con il puntatore del mouse, se invece volete aggiungere dei clienti in quella piazzola, cliccateci sopra.
Il software inoltre permette anche di:

- **Modificare il Campeggio :** Per modificare il campeggio(spostare,eliminare e inserire piazzole), click sul menù modifica.
- **Gestione i Dipendenti:** Click sul menù dipendenti e qui si potrà inserire il dipendente, e modificare informazioni dei dipendenti già inseriti in precedenza.
- **Visualizzare tutti i Clienti:** Con il click sul menu "Clienti" è possibile vedere tutti i clienti nel campeggio e relative informazioni

- **Gestiew le Attività :** Dal menu “Attivita” è possibile creare un’attivita , assegnare partecipanti e cancellarla
- **Statistiche:** Dal menu “Statistiche” è possibile visualizzare grafici per le categorie : Bilancio,Clienti entrati,età media e attivita preferite del mese corrente; è possibile controllare anche i clienti entrati in una specifica data o periodo
- **Esporta :** Tramite File -> Esporta è possibile esportare le informazioni relative ai clienti e dipendenti nei formati “csv” e “excel”
- **Salva/Carica:** Tramite File -> Salva è possibile salvare il campeggio creato e tutte le informazioni ad esso relative; tramite File -> Carica vengono caricate le informazioni salvate.
- Inoltre è possibile Esportare anche i dati dai grafici, per fare questo bisogna cliccare col destro sul grafico e selezionare “Save as” selezionando l’estensione desiderata.
- La piazzola inoltre ha diversi colori: se è verde è vuota, se è azzurra ci sono tende, se è grigia c’è un camper, se è blu c’è un bungalow, se è gialla c’è una roulotte.