Progetto di Metodologie di Programmazione

A cura di:

- Bucciero Samuele, matricola: 5618430
  E-mail: samuele.bucciero@gmail.com
  Prove in itinere: Rifiutate
- Mercatanti Elia - matricola: 5619856
  E-mail: eliamercatanti@yahoo.it
  Prove in itinere: Superate

Data consegna: 16/01/2015

Progetto dell'esercizio: Agenzia di Viaggi

Traccia per l'esercizio:

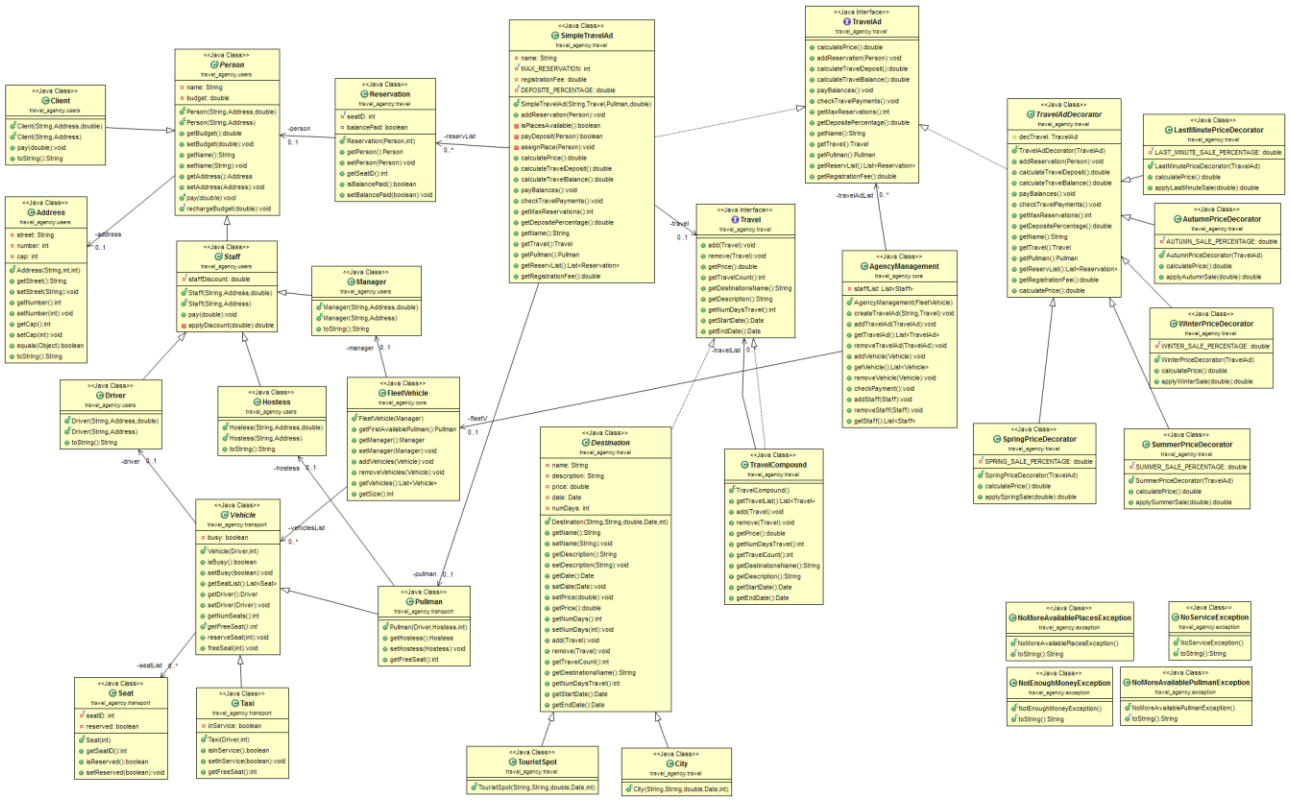
L'agenzia e*vi@ggi svolge la sua attività nel settore dei viaggi di breve/media durata a bordo di Pullman Gran Turismo, verso alcune delle principali mete turistiche italiane, con partenza da Firenze. L'agenzia formula diverse offerte di viaggio, si occupa della gestione delle prenotazioni e della vendita dei viaggi; ha un proprio parco automezzi e ha alle proprie dipendenze un Responsabile e un certo numero di hostess e di autisti (si può prevedere, eventualmente, anche una lista di hotel convenzionati, per ognuna delle mete turistiche trattate).
Ogni viaggio è relativo a un solo programma di viaggio, che è contraddistinto da un nome e ha associata una quota di iscrizione. Un programma di viaggio può essere attuato in più date: i viaggi relativi differiscono tra loro, a parte per le date di partenza/ritorno, per il costo (soggetto a variazioni stagionali e per il numero di posti sul/sui pullman.
Un viaggio può variare dinamicamente la sua quota di iscrizione, per sconti last minute applicati dall'agenzia sui posti rimasti.
A bordo di ogni pullman ci sono un autista e una hostess.
I clienti possono prenotare via web il viaggio che desiderano effettuare. Per ogni posto prenotato, sono inseriti il nome e i dati anagrafici del passeggero cui il posto va intestato.
Al momento della prenotazione è richiesto un acconto del 20% del costo del viaggio, il saldo avviene un giorno prima della partenza del viaggio. In caso contrario, l'acconto è perso: se si vuole, si può gestire una lista di attesa, così i posti non pagati all'atto del saldo sono assegnati a eventuali passeggeri in lista d'attesa.
Il sistema può supportare meccanismi di riprotezione dei viaggi: se il pullman si guasta, ne viene reperito un altro, se necessario affittandolo da una ditta esterna convenzionata.
Si può prevedere la possibilità che il cliente si organizzi un viaggio come pacchetto di viaggi (oppure che anche l'agenzia possa offrire viaggi che siano pacchetti di viaggi).

Specifiche e Funzionalità della soluzione proposta:

Come richiesto dall'esercizio la soluzione proposta in questo progetto andrà a gestire la prenotazione, la pubblicazione e la vendita dei viaggi dell'agenzia e*vi@ggi.
Nella nostra soluzione abbiamo pensato che l'agenzia possa offrire delle tipologie di viaggio tramite vari annunci caricati dall'azienda. Ogni viaggio offerto può essere anche un insieme di pacchetti di viaggi ed ognuno di essi verrà associato ad un annuncio che l'azienda pubblicherà. Ad ogni annuncio di viaggio sarà associato un nome identificativo, la lista delle registrazioni totali per quell'annuncio, un pullman che verrà utilizzato per il trasporto dei clienti registrati per il viaggio e una quota di iscrizione. Ogni pullman avrà un numero di posti fisso a cui è associato un identificativo e in fase di registrazione ad ogni cliente che richiederà di iscriversi verrà associato un posto del pullman. In fase di registrazione verrà chiesto subito al cliente di pagare la quota di iscrizione al viaggio più un acconto del 20% sul totale del costo del viaggio (pacchetto viaggi). Il giorno prima della partenza del viaggio verrà fatto pagare il costo intero del viaggio, se il cliente non lo paga perderà l'acconto versato e la sua registrazione verrà rimossa liberando anche il posto precedentemente assegnato. L'azienda potrà anche applicare vari sconti stagionali o last minute al costo totale del viaggio (pacchetto di viaggi). L'azienda avrà a disposizione, come richiesto, una parco auto, una lista del personale e un manager. Le varie destinazioni che potranno essere incluse nei viaggi proposti differiranno per vari aspetti, come la data, il numero di giorni che il cliente passerà in quella precisa destinazione, la descrizione e il prezzo relativo a quella tappa. La soluzione proposta inoltre non presenterà meccanismi di protezione viaggi e per semplicità non verrà trattata l'interfaccia grafica, che in questo contesto apparirebbe secondo noi troppo complessa e anche irrilevante dal punto di vista della progettazione.

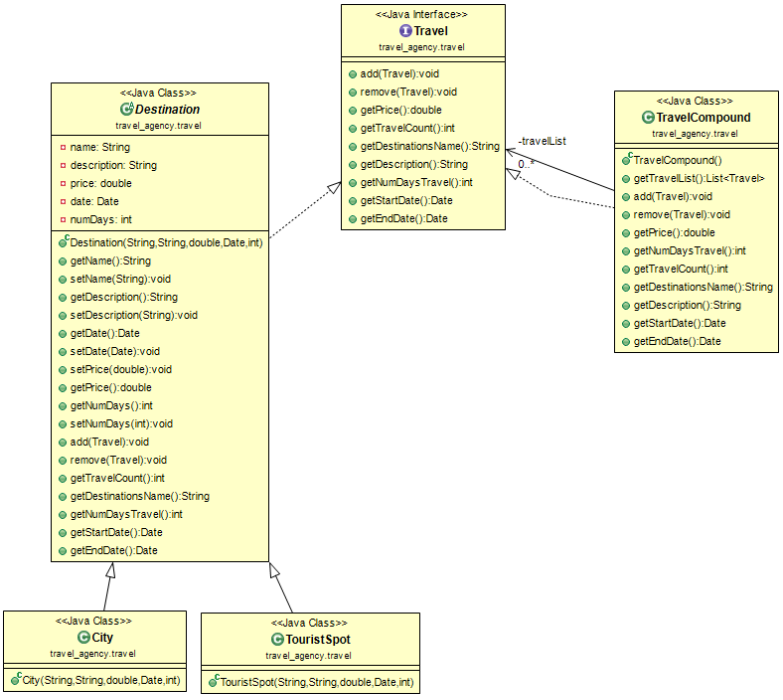
Il diagramma UML completo dell'intero progetto essendo abbastanza grande verrà allegato alla relazione in un file .png per visualizzarlo al meglio.

Diagramma UML generale del progetto proposto:



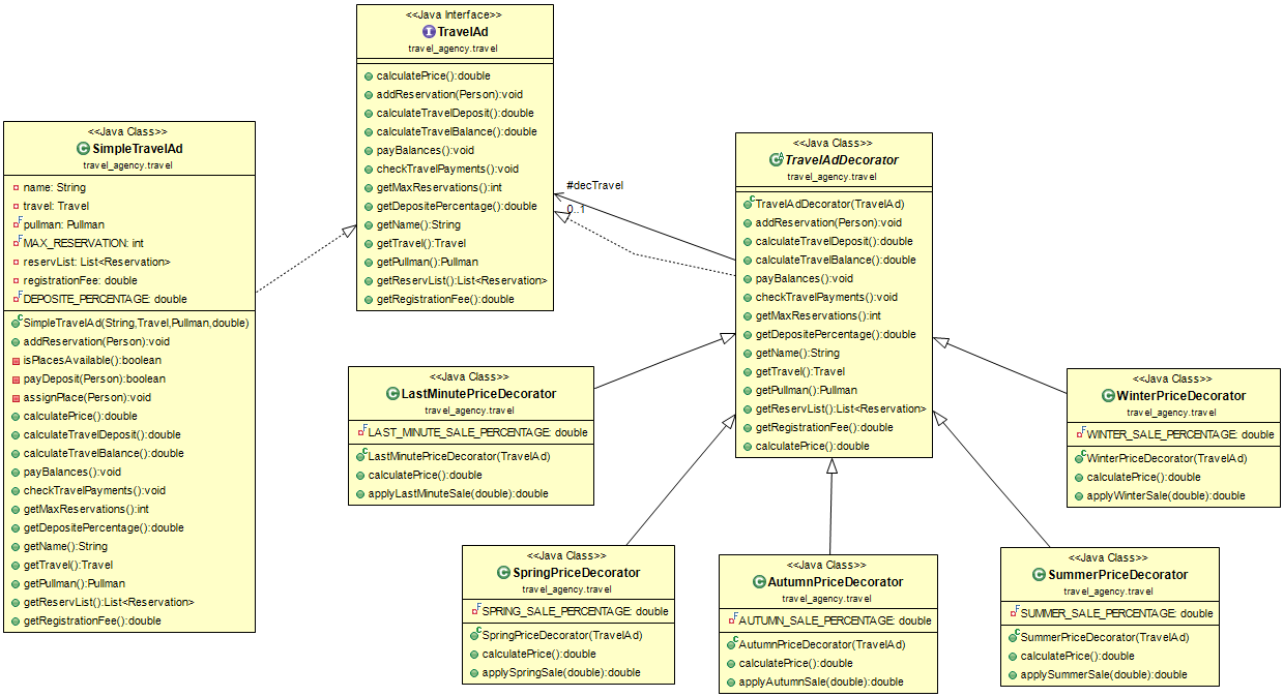
Scelte di Design: Pattern utilizzati

Composite Pattern:



Uno dei principali pattern che abbiamo utilizzato è il Composite Pattern. Esso permette di organizzare gli oggetti in una struttura ad albero, nella quale i nodi sono degli oggetti composti e le foglie sono oggetti semplici ed è quindi utilizzato per dare la possibilità ai client di manipolare oggetti singoli e composizioni in modo uniforme senza particolari discriminazioni. Proprio per questo abbiamo scelto questo pattern per sviluppare la struttura dei pacchetti di viaggi.
Nel nostro caso abbiamo utilizzato la Classe TravelCompund per gestire e rappresentare i viaggi composti, mentre per rappresentare le "foglie" di questa struttura abbiamo utilizzato la Classe Destination, che rappresenta appunto le caratteristiche che deve avere un viaggio semplice, ovvero con una destinazione. Ogni destinazioni dovrà avere infatti un nome e una descrizione che la identifica, una data che rappresenta il giorno in cui verrà raggiunta tale destinazione, il numero di giorni di permanenza e infine il costo del viaggio verso quella destinazione. Le destinazioni possono essere poi specializzate in citta oppure luoghi turistici.
Utilizzando questo pattern abbiamo quindi reso possibile all'azienda la possibilità di creare pacchetti di viaggi che potranno poi essere pubblicati e proposti ai vari clienti garantendo una gestione omogenea sia delle destinazioni singole sia dei viaggi composti. Al pacchetto di viaggio creato con questo pattern, grazie all'interfaccia comune Travel, potremo così in seguito richiedere alcune informazioni utili come il costo totale del pacchetto, una descrizione generale di esso, le date di inizio e fine viaggio ecc.

Decorator Pattern:



Il secondo pattern che abbiamo utilizzato è stato il Decorator Pattern, che viene utilizzato per aggiungere durante il run-time nuove funzionalità ad oggetti già esistenti, costruendo una nuova classe decoratore che "avvolge" l'oggetto originale.
Nel nostro caso abbiamo quindi utilizzato questo specifico pattern per poter applicare dinamicamente (in fase di run-time) agli annunci di viaggio dei particolari sconti stagionali o sconti last minute. In particolare lo sconto verrà applicato sul costo totale di viaggio ottenuto dalla struttura del pacchetto di viaggi. Abbiamo utilizzato la classe TravelAdDecorator come classe di base per applicare i vari sconti al calcolo del prezzo totale del viaggio, e in seguito utilizzato le classi AutumnPriceDecorator, LastMinutePriceDecorator ecc. per specializzare la classe TravelAdDecorator in modo tale da applicare lo sconto desiderato. La classe SimpleTravelAd, che rappresenta l'annuncio relativo al

viaggio/pacchetto di viaggi pubblicato dall'azienda, sarà utilizzata come classe concreta a cui aggiungere tali sconti.

Scelte di Design: Principi di progettazione

I principali principi di progettazione utilizzati sono stati:

- Dependency Inversion Principle: ovvero che le classi dovrebbero dipendere da astrazioni e non da classi concrete. Questo principio è stato utilizzato un po' ovunque nel progetto per cercare di non legare troppo le classi client a classi concrete facilmente modificabili. In particolare è stato utilizzato per la gestione dei Vehicle dell'azienda ma anche per la gestione degli users, ovvero per la classe Person e Staff.
- Single Responsibility Principle: che afferma che ogni elemento di un programma (classe, metodo, variabile) deve avere una sola responsabilità, e che tale responsabilità debba essere interamente incapsulata dall'elemento stesso. Anche questo principio è stato utilizzato in gran parte del progetto ma soprattutto per la gestione delle Destination e per la gestione del Personale

Descrizione sintetica della parte implementata e dei test effettuati:

Il progetto si divide in cinque pacchetti:

- Pacchetto travel_agency.core: racchiude tutte le classi che in qualche modo inglobano la gestione principale dell'azienda di viaggi, come ad esempio la classe AgencyManagment di cui discuteremo in breve di seguito quale siano le sue funzioni.
- Pacchetto travel_agency.exception: racchiude le principali eccezione che l'applicazione andrà a gestire.
- Pacchetto travel_agency.transport: racchiude le classi che gestiscono i veicoli e in generali i mezzi di trasporto a disposizione dell'agenzia.
- Pacchetto travel_agency.travel: racchiude le classi che gestiscono la creazione, la pubblicazione e la gstione dei viaggi proposti dall'agenzia.
- Pacchetto travel_agency.users: racchiude le classi che gestiscono il personale dell'azienda e anche i possibili clienti di essa.

Il pacchetto core contiene la classe FleetVehicle che gestisce il parco auto che possiede l'agenzia, contiene la lista di veicoli posseduti dall'azienda e il manager che lo gestisce. AgencyManagement invece verrà utilizzata per gestire la lista degli annunci che l'agenzia pubblicherà, la lista del personale dell'azienda e un riferimento al parco auto posseduto dall'azienda.

Il pacchetto exception contiene la classe NoMoreAvailablePlacesException, eccezione che verrà lanciata quando saranno finiti i posti sul pullman relativo ad un determinato annuncio, l'eccezione NoServiceException che verrà lanciata per avvisare che un determinato veicolo non è in servizio, l'eccezione NotEnoughMoneyException che verrà lanciata per indicare che la persona corente non ha abbastanza soldi per completare l'oprazione di pagamento, e l'eccezione NoMoreAvailablePullmanException che verrà lanciata quando saranno finiti i pullman disponibili per i viaggi.

La classe principale del pacchetto transport è la classe Vehicle che fa da classe base per ogni tipo di veicolo appartenete all'azienda. Le classi Taxi e Pullman la specializzano. La classe Pullman verrà

poi utilizzata nella classe SimpleTravelAd per assegnare ad ogni annuncio di viaggio un pullman che verrà utilizzato per il trasporto.

Il pacchetto travel oltre a contenere le classi per la gestione degli sconti stagionali e per la creazione e la gestione del pacchetto di viaggi, di cui abbiamo già discusso, contiene la classe SimpleTravelAd che come abbiamo già accenato gestisce l'annuncio di viaggio che l'azienda può pubblicare. In particolare SimpleTravelAd avrà quindi il compito di calcolare il prezzo totale del viaggio e di applicarci gli eventuali sconti, di gestire una lista di prenotazioni di clienti che vogliono intraprendere il viaggio e di far pagare al momente giusto ai clienti sia l'acconto che il prezzo finale del viaggio. La classe Reservation è la classe che gestisce le prenotazioni al viaggio, associa ad ogni cliente che ha conseguito la registrazione il suo relativo posto sul pullman, inoltre tiene conto dello stato di pagamento per il viaggio, in particolare controlla se il cliente a pagato o meno l'intera somma dovuta.

Il pacchetto users in particolare conterrà la classe Person che fa da classe base per ogni possibile utente che l'agenzia dovrà gestire. Ogni Person oltre ad avere un nome ed un indirizzo avrà anche un budget che rappresenta sostanzialmente una sorta di conto che l'utente ha a disposizione per eseguire i pagamenti. La classe Staff invece viene utilizzata come classe base per il personale che lavora per l'azienda in questione. Infine è presente anche una classe Address per gestire correttamente gli indirizzi di tutti i possibili utenti.

Il proggetto contiene anche tutti i vari test dei vari pacchetti e quindi di tutto il test del codice per la soluzione proposta rispettando tutti i vari principi per il testing corretto. Tutti i test sono stati eseguiti tramite il framework JUnit presente in Eclipse.
Il diagramma UML completo dell'intero progetto essendo abbastanza grande verrà allegato alla relazione in un file .png per visualizzarlo al meglio.