DOCUMENTAZIONE

Traccia 3 – Progetto OOP

1. Descrizione Generale del Problema

   Questa prima parte del progetto ci richiede di implementare il Class Diagram, e la struttura del relativo codice in Java, di un sistema informativo esclusivo per la gestione dell'aeroporto di Napoli. Qui gli utenti, suddivisi in Amministratori ed Utenti Generici, possono interagire con il sistema per effettuare, cercare e modificare prenotazioni, inserire e modificare voli, e altre funzionalità che analizzeremo nel dettaglio.

2. Struttura delle Classi
   - *Utente*: Rappresenta un generico utente del sistema, il quale può accedere tramite login e password. Gli attributi della classe sono "email", "username", "password". La scelta di inserire "username" nella classe Utente, e non in una sottoclasse, è dovuta dall'intenzione di dare un'identità comune a tutti gli utenti, vista la sola presenza di email e password (necessari per effettuare il login). La classe è suddivisa in due sottoclassi con privilegi distinti. La classe Utente presenta un solo metodo "visualizzaVoli()", il quale permette agli utenti (qualsiasi siano i loro privilegi) di visualizzare i voli disponibili.
   - *Amministratore*: Sottoclasse di Utente. Gli amministratori hanno accesso alle operazioni di gestione e aggiornamento dei voli. Non presenta altri attributi in aggiunta a quelli già ereditati dalla classe Utente. Per quanto riguarda i metodi ne troviamo ben 3. "inserimentoVolo()" aggiunge un nuovo volo al sistema, "aggiornaVolo()" permette di aggiornare i dettagli di un volo esistente, "modificaGateImbarco()" invece permette di gestire il gate di imbarco per i voli in partenza.

- *Utente Generico*: Sottoclasse di Utente. Gli utenti generici possono effettuare prenotazioni e gestirle attraverso il sistema. Anche questa classe non ha attributi ulteriori a quelli ereditati dalla superclasse. Qui troviamo 4 ulteriori metodi. “prenotaVolo()” registra una nuova prenotazione nel sistema, “cercaPrenotazioneVolo(numeroVolo) permette di trovare la prenotazione partendo dal numero identificativo del volo interessato, “cercaPrenotazioneVolo(username)” permette invece di trovare la prenotazione fornendo però la sola username dell’utente, “modificaPrenotazione()” consente di aggiornare i dati della prenotazione qualora fosse permesso.
- *Volo*: Rappresenta un volo schedulato all’interno del sistema. I suoi attributi sono “numeroVolo” (ovvero identificativo univoco del volo), “compagniaAerea” (nome della compagnia aerea), “orarioPrevisto” (orario di partenza/arrivo previsto), “data”, “ritardo” (eventuale!), “statoVolo” (indica la fase attuale del volo). Qui troviamo l’enumerazione “Stato Volo”, che permette di descriverne lo stato selezionando uno dei 5 valori preimpostati, ovvero “programmato”, “decollato”, “in ritardo”, “atterrato”, “cancellato”.
- *Volo in Partenza*: Analizzando la traccia assegnata, viene specificato di voler assegnare l’aeroporto di destinazione solamente per i voli in partenza da Napoli! Di conseguenza, troviamo questa sottoclasse contenente come attributi “gateImbarco” e “aeroportoDestinazione”.
- *Volo in Arrivo*: Anche questa eredita gli attributi della superclasse “Volo”, ai quali viene aggiunto “aeroportoOrigine”. Questo perché la traccia richiede che l’aeroporto di origine venga specificato solo per i voli in arrivo da Napoli! Sia ‘Volo in Arrivo’ che ‘Volo in Partenza’ possono essere considerate specializzazioni di ‘Volo’, in quanto aggiungono dettagli specifici per ciascun caso (arrivo e partenza).
- *Gate*: Infine abbiamo la classe Gate, che serve banalmente per identificare i punti di imbarco per i voli in partenza. Qui abbiamo un attributo “numeroGate” di tipo short.
- *Prenotazione*: Rappresenta la prenotazione di un volo effettuato da un utente. Gli attributi sono “codicePrenotazione” (identificativo della prenotazione) e “stato” (stato attuale della prenotazione). Anche qui troviamo l’enumerazione “Stato Prenotazione”, che ci permette di

scegliere uno tra i 3 valori predefiniti, ovvero “confermata”, “in attesa”, “cancellata”.

- *Ticket Passeggero*: Questo contiene i dettagli del passeggero associato ad una prenotazione. Bisogna specificare che, nei sistemi moderni, una prenotazione piò esser associata a più passeggeri, generando quindi più ticket sotto un unico codice di prenotazione. Questa classe ci permetterà di evitare ridondanze sgradevoli. Gli attributi sono “numeroTicket” (numero del biglietto. È univoco per una specifica prenotazione! Una prenotazione non può avere due “numeroTicket” identici!!), “nomeCompleto” (si intende il nome con il quale si è registrati all’anagrafe nazionale), “cognome”, “dataNascita”, “numeroDocumento”, “postoAssegnato” (anche questo attributo, volendo, può esser inteso come primario. In uno specifico volo non potremo avere due viaggiatori con lo stesso posto assegnato).

3. Relazioni tra le Classi

- Un ‘Utente Generico’ può effettuare svariate ‘Prenotazioni’, così come può non effettuarne nessuna. Una prenotazione viene effettuata da un solo Utente. Non possiamo avere due utenti che effettuano la stessa prenotazione!
- Una ‘Prenotazione’ può portare alla creazione di uno o più ‘Biglietti’, m a un ticket può appartenere ad una sola ed unica Prenotazione! Non potremo avere un biglietto associato a due prenotazioni distinte.
- Una ‘Prenotazione’ può far riferimento ad un solo specifico ‘Volo’, ma al contrario uno specifico volo può esser soggetto a svariate prenotazioni.
- Un utente ‘Amministratore’ può gestire anche svariati ‘Voli’. Siccome nella traccia non viene fatto alcun riferimento a questa relazione, ipotizziamo che un volo possa esser gestito da un solo utente Amministratore, il quale riuscirà ad apportare modifiche grazie a privilegi adeguati.
- Un ‘Gate’ può esser associato a più ‘Voli in Partenza’, anche se l’imbarco non avviene nello stesso preciso istante. Un volo in partenza, al contrario, avrà un solo gate tramite il quale le persone potranno imbarcarsi.

4. Funzionamento Essenziale del Sistema

- Gli amministratori possono aggiungere e aggiornare voli, modificare gate e gestire il sistema.
- Gli utenti generici possono prenotare voli e gestire (limitatamente) le proprie prenotazioni.
- Il sistema permette di visualizzare lo stato dei voli e delle prenotazioni, oltre alla gestione dei gate di imbarco.

5. Princìpi Utilizzati nel Class Diagram

L'utilizzo dell'ereditarietà ci consente di organizzare le informazioni in maniera più efficace e chiara, anche a livello visivo. Ad esempio, la classe utente funge da superclasse, da cui derivano le sottoclassi Amministratore e Utente Generico. Ciò consente di condividere attributi comuni (come email, username e password), evitandone la ridondanza. Non solo vengono 'ereditati' gli attributi, ma nel nostro caso sono presenti anche metodi ereditati dalla superclasse! È stato accennato anche il concetto di specializzazione, che vediamo con le classi "Volo in Arrivo" e "Volo in Partenza". Anche questa volta abbiamo il vantaggio di evitare la ridondanza di attributi, ed un eventuale loro utilizzo improprio. Ad esempio, sappiamo che i voli in partenza necessitano di un gate disposto all'imbarco dei passeggeri, mentre per quanto riguarda i voli in arrivo ciò non accade!!!

Antonio Andrea Montella, N86005652
Adriano Montella, N86005823