



Programmazione Object Oriented

Esercitazione Finale

Dr. Dario Di Nucci - 12/12/2022

Cosa impariamo oggi?

	Lunedì (ore: 9-13 Aula: Lab Hopper)	Lunedì (ore: 9-11 Aula: Lab Hopper)	Martedì (ore: 9-11 Aula: F/1)	Giovedì (ore: 9-11 Aula: F/1)
Settimana 5 (17-20 Ottobre)	Lab: Packages e variabili statiche		Progettazione di Classi	Interfacce e Polimorfismo
Settimana 6 (24-27 Ottobre)	Lab: Interfacce		Interfacce e Polimorfismo	Ereditarietà
Settimana 7 (31 Ottobre - 3 Novembre)	Lab: Ereditarietà			Ereditarietà
Settimana 8 (7-10 Novembre)		Lab: GregorianCalendar	Testing e Debugging	Collezioni Programmazione Generica
Settimana 9 (14-17 Novembre)		Lab: Eccezioni	Gestione delle Eccezioni	IO e Flussi
Settimana 10 (21-24 Novembre)		Lab: File	Programmazione Grafica	Gestione degli Eventi
Settimana 11 (28 Nov - 1 Dic)		Lab: Eventi e interfacce grafiche	Interfacce Grafiche Utente	Lambda Expressions
Settimana 12 (5-7 Dicembre)			Reflections	
Settimana 13 (12-15 Dicembre)	Lab: Esercitazione finale		Reflections	Simulazione prova

Esercizio – Gestione Albergo

Realizzare una piccola applicazione con GUI per la gestione delle prenotazioni di un albergo.

Il sistema deve tenere traccia

- dei clienti (cognome, nome, telefono, numero di carta di credito),
- delle stanze (numero, piano, tipologia, servizi, telefono),
- delle prenotazioni (check-in, check-out, cliente, stanza).

All'inizio il sistema chiede il nome del file dal quale prelevare i dati, permettendo poi di selezionare tra *Richiesta Prenotazione* e *Gestione Prenotazione*.

Con *Richiesta Prenotazione* è possibile inserire le date di check-in/check-out, la tipologia (i.e., singola, doppia, matrimoniale) della stanza e chiedere la disponibilità.

A questo punto viene è possibile (i) visualizzare i *Dettagli* dell'eventuale stanza disponibile, e (ii) *Prenotare* la stanza

Nel primo caso vengono evidenziate le caratteristiche della stanza in formato testo.

Nel secondo caso si richiedono i dati del cliente e viene mostrato un messaggio di conferma della prenotazione oppure un messaggio d'errore appropriato (e.g., carta di credito non valida, stanza già prenotata).

Con *Gestione Prenotazione* si richiedono le informazioni del cliente e la richiesta (se cancellazione o modifica di caratteristiche).

N.B. Quando una prenotazione è passata, cioè la data di check-out è antecedente la data odierna, questa deve essere cancellata dall'archivio.

Esercizio - Aeroporto

Un volo contiene le informazioni sulla tratta (aeroporto di partenza e di destinazione), codice volo, compagnia aerea, tipologie di biglietto acquistabili (i.e., economy, business, first class) e la lista di posti con lo stato (i.e., disponibile/occupato).

Implementare il metodo `List<Seat> filterSeats(Filter f)` che restituisca la lista di posti che soddisfino il filtro `f`. Sovrascrivere i metodi `toString`, `equals` e `clone` di `Object` in maniera da massimizzare il riutilizzo del codice.

Un posto ha un numero di fila, una lettera indicante la posizione nella fila e uno stato (i.e., libero/prenotato).

Un filtro permette di filtrare i posti per numero di fila, posizione nella fila e stato.

Una prenotazione tiene traccia delle informazioni anagrafiche del cliente e delle informazioni sul volo prenotato (codice volo, tipo biglietto, posto).

Scrivere la classe `FlightList` che modelli una collezione di voli e fornisca i seguenti metodi:

- `void addFlight(Flight v)` che inserisca un volo nell'archivio. Se l'oggetto `v` ha un codice volo uguale ad uno già presente nell'archivio viene lanciata l'eccezione controllata `BadCodeException`.
- `FlightList search(String o, String d)` che restituisca un oggetto `FlightList` contenente tutti i voli che partono da `o` ed arrivano a `d`.

Esercizio - Aeroporto 2

Considerando le classi ai punti precedenti, scrivere una classe Java che realizzi un'interfaccia grafica per la prenotazione di posti su un volo.

Nell'interfaccia grafica

- deve essere possibile specificare una tratta e alla pressione di un pulsante visualizzare in un area di testo i voli abilitati su quella tratta.
- i voli visualizzati devono poter essere selezionati con un pulsante a menu (casella combinata).
- una volta selezionato un volo, permette di scegliere il posto (tra quelli liberi). Usare il metodo `filterSeats` creando un opportuno filtro.
- una volta selezionato il posto si apre una finestra che consente di immettere i dati anagrafici del passeggero.
- una volta terminata la prenotazione, i dati del volo devono essere registrati su file.

Implementare un'eccezione non controllata per segnalare che un volo non ha posti liberi. L'eccezione deve essere lanciata quando si seleziona un volo senza posti liberi.