

# Prenotazione Voli



Thread  
synchronized

# Descrizione

Realizzare una classe **Assegnatore** che dispone di 20 posti su un volo

La classe fornisce 2 metodi di business:

- ▶ `void assegnaPosti(String cliente,  
int numPosti)`
  - ▶ Verifica se sono disponibili i posti specificati, in caso affermativo li scala dal totale, esegue una stampa dove riporta il nome cliente e il totale posti assegnati
  - ▶ Viceversa segnala la mancata disponibilità sollevando un'eccezione `PostiNonDispException`
  - ▶ NOTA: i posti si assegnano tutti o nessuno!
- ▶ `int getTotalePosti()`
  - ▶ Calcola il totale dei posti rimanenti

La classe Assegnatore sarà condivisa dai Thread clienti descritti di seguito

# Clienti

Realizzare una classe **Cliente** che rappresenta l'acquirente dei voli.

Sarà definita da un nome, un numero di posti che richiede e un'istanza di Assegnatore.

- ▶ Il Cliente prova a prenotare i voli richiedendoli al suo assegnatore
- ▶ Se i posti sono disponibili, stampa un **messaggio di successo** che indica il numero di posti acquistati
- ▶ Se non c'è più disponibilità di posti, gestisce l'eccezione ricevuta e verrà mostra un **messaggio di fallimento** per l'acquisto del numero di posti specificato

# Simulazione

Realizzare infine una classe **Simulazione** col main nel quale verranno istanziati un Assegnatore e 4 Clienti che condividono l'assegnatore.

Le richieste totali dei clienti dovranno superare il numero massimo di posti disponibili.

Utilizzare la keyword **synchronized** per assicurarsi che non ci siano operazioni incongruenti sul totale dei voli venduti.