

Aeroporto

Progettare ed implementare un programma che consenta di gestire le informazioni relative a degli aeroporti ed ai viaggi da essi effettuati. Tutte le classi, ad eccezione della classe `Esempio`, devono appartenere al package `aeroporto`.

R1. Aeroporto

La classe principale con cui interagire è la classe `Aeroporto`, il cui costruttore riceve come parametri la denominazione dell’aeroporto, l’indirizzo ed il numero massimo di aerei che la struttura è in grado di ospitare (ad esempio, l’aeroporto Sandro Pertini Torino-Caselle, in Via Aeroporto 12, con 50 aerei). Tali informazioni sono accessibili attraverso i rispettivi metodi `getDenominazione()`, `getIndirizzo()` e `getNumeroAerei()`.

E’ possibile impostare il numero di aerei che possono accedere contemporaneamente alla pista di decollo attraverso il metodo `setNumeroDecolli()`. Il metodo `descrizioneAeroporto()` della classe `Aeroporto` restituisce una stringa contenente il numero massimo di aerei ospitabili e, tra parentesi, il numero di decolli contemporanei consentiti, separati da uno spazio: ad esempio `50 (2)` per un aeroporto che può ospitare 50 aerei e che permette un massimo di 2 decolli contemporanei.

R2. Aerei

Il programma memorizza le informazioni relative agli aerei situati in un particolare aeroporto. L’aggiunta di un aereo ad un aeroporto avviene attraverso il metodo `aggiungiAereo()` della classe `Aeroporto`, che riceve come parametri il modello, la capienza dell’aereo (numero di passeggeri trasportabili) e il numero massimo di chilometri percorribili per una tratta; il metodo assegna inoltre all’aereo aggiunto un codice numerico identificativo progressivo a partire da 0. Il metodo restituisce il codice numerico assegnato.

Si assuma che lo stesso aereo non possa essere aggiunto più volte: cercando di aggiungere un aereo già presente (stesse informazioni per tutti gli attributi), il metodo restituisce semplicemente l’identificativo numerico relativo all’aereo già presente. Cercando invece di aggiungere un numero di aerei superiore a quello ammissibile per l’aeroporto, il metodo non sortisce alcun effetto (ed il valore di ritorno non è rilevante).

Per ottenere le informazioni relative ad un aereo presente nella flotta di un determinato aeroporto è possibile utilizzare il metodo `aereo()` della classe `Aeroporto`, che riceve come parametro il codice numerico dell’aereo e restituisce una stringa contenente modello, capienza e chilometri percorribili, separati dal carattere `;` (esempio, `Boeing 747;300;5000`).

Il metodo `aerei()` della classe `Aeroporto` restituisce invece un array contenente delle stringhe formate dagli identificativi e dai modelli degli aerei presenti nell’aeroporto, in quel determinato istante, separati dal carattere `;` (esempio, `1;Boeing 747`). La dimensione dell’array è pari al numero di aerei attualmente presenti, i cui modelli sono riportati in ordine di aggiunta.

R3. Viaggi

Per aggiungere un viaggio in partenza dall’aeroporto si utilizza il metodo `aggiungiViaggio()` della classe `Aeroporto`. Il metodo riceve come parametri il nome della tratta, il numero di passeggeri prenotati per la partenza e il numero di chilometri che caratterizza la tratta. Il metodo assegna al viaggio un aereo disponibile che soddisfi i requisiti in termini di capienza di passeggeri e chilometri percorribili massimi. Il metodo cerca fra gli aerei disponibili quello che minimizza lo scarto in termini di posti occupati (ovvero l’aereo con il numero di posti disponibili più vicino a quelli prenotati per il viaggio); in caso di parità tra più aerei, il metodo cerca l’aereo che minimizza lo scarto rispetto al numero di chilometri percorribili (ovvero l’aereo con l’autonomia più vicina a quella necessaria per il viaggio). Nel caso di ulteriore parità, il metodo assegna il primo aereo disponibile (in ordine di aggiunta). Il metodo deve infine assicurarsi che un aereo sia assegnato ad un solo viaggio. Nel caso in cui un aereo che soddisfi i suddetti requisiti sia disponibile, il metodo effettua l’assegnazione e restituisce una stringa composta dal codice identificativo dell’aereo selezionato e il nome della tratta, separati dal carattere `;` (ad esempio, nel caso in cui il viaggio con tratta Torino-Bari venga assegnato all’aereo con identificativo 1, il metodo restituirà la stringa `1;Torino-Bari`). Nel caso in cui nessuno degli aerei soddisfi i suddetti requisiti o nel caso in cui a tutti gli aerei sia già assegnato un viaggio, il metodo non sortisce alcun effetto (ed il valore di ritorno non è rilevante).

R4. Elenco viaggi

Per ottenere le informazioni relative ad un viaggio è possibile utilizzare il metodo `viaggio()` della classe `Aeroporto`, che riceve come parametri l’identificativo di un aereo ed il nome di una tratta. In caso di identificativo esistente e di aereo assegnato ad un viaggio il cui nome della tratta è passato come parametro, il metodo restituisce una stringa composta dal codice numerico dell’aereo, nome della tratta, numero di passeggeri prenotati e numero di chilometri della tratta separati da `;` (esempio, `5;Torino-Bari;150;1050`). In tutti gli altri, casi il metodo non sortisce alcun effetto (ed il valore di ritorno non è rilevante).

Il metodo `viaggi()` della classe `Aeroporto` restituisce una stringa contenente tutti i viaggi finora organizzati, uno per riga, nell’ordine in cui sono stati aggiunti. Ogni riga ha il formato sopra riportato, ed è terminata da un carattere di ‘a capo’ (tranne l’ultima). Attraverso il metodo `viaggiPerTratta()` è possibile ottenere l’elenco dei viaggi organizzati il cui nome tratta corrisponde a quello passato come parametro, nel formato utilizzato dal metodo precedente.