

Prova pratica del 2 settembre 2020

Ricordarsi di consegnare il codice sorgente di tutte le classi e della classe main salvando il progetto Code::Blocks in Z:

Simulazione agenzia di viaggi

Si scriva un programma per simulare la prenotazione di viaggi in un'agenzia. L'agenzia vende **Viaggi**. Ogni Viaggio è identificato da un **ID**, una **destinazione finale**, un **costo** in euro, una **data di inizio**, una **data di fine**, da un **numero di posti disponibili** e da una **lista di Viaggiatori** (si veda dopo) che si sono prenotati per il Viaggio.

Esistono due tipologie di Viaggio: **Crociera** e **Tour**. La **Crociera** ha anche l'indicazione se c'è o meno l'**animazione a bordo**. Un **Tour** ha invece l'indicazione se è prevista o meno l'**assicurazione** e una **lista di Guide** (si veda dopo) che accompagnano i Viaggiatori nel Tour e che può essere anche vuota (cioè il Tour non è guidato).

La classe Viaggio (e quindi anche le sue classi derivate) deve prevedere, come minimo, i seguenti metodi:

- Un metodo **prenota** a cui il Viaggiatore che effettua la prenotazione e il numero di viaggiatori (es: nel caso di un Viaggio prenotato da una famiglia, prenota il padre e il numero dei viaggiatori è 4, lui, moglie e i due figli); il metodo deve verificare se ci sono ancora posti disponibili a sufficienza e se così non è ritornare *false*. Se invece ci sono posti a sufficienza il metodo deve inserire il Viaggiatore nella lista di Viaggiatori e decrementare correttamente il numero di posti disponibili per il Viaggio, ritornando come valore *true*;
- Un metodo **cancella** a cui passare il Viaggiatore e il numero previsto di viaggiatori e provvede a cancellare il Viaggiatore dalla lista e ad aggiornare il numero di posti disponibili; il metodo deve comunque prima verificare che risulti prenotato quel Viaggio da quel Viaggiatore.

Inoltre, la classe Tour deve anche prevedere un metodo **guidato** che ritorna un booleano ad indicare se il Tour è con guida (almeno una) o meno.

Nel sistema devono essere modellate le **Persone**. Ogni Persona è caratterizzata da un **ID**, un **nome**, un **cognome**, l'**età** e il **sex**. Si vogliono modellare due tipologie di Persona: **Viaggiatore** e **Guida**. Il Viaggiatore ha anche una **nazionalità**, mentre la Guida ha anche una **lista di lingue conosciute** (ciascuna memorizzata come stringa).

(CONTINUA NEL PROSSIMO FOGLIO)

Il main deve inizializzare una serie (non ne servono molti) di Viaggi di ogni tipologia, assegnando ad essi, dove necessario, delle Guide. Quindi deve chiedere all'utente se:

- a) Vuole registrare un nuovo Viaggiatore e nel caso chiedere i dati necessari (l'ID può essere chiesto o generato in modo progressivo) e creare il relativo oggetto, per poi visualizzare i dati del Viaggiatore creato
- b) Vuole prenotare un Viaggio e nel caso chiedere l'ID del Viaggio da prenotare, l'ID del Viaggiatore principale (che deve essere stato prima creato) e il numero di partecipanti (incluso il Viaggiatore principale)
- c) Vuole cancellare un Viaggio e nel caso chiedere l'ID del Viaggio da cancellare, l'ID del Viaggiatore principale (che deve avere una prenotazione per quel Viaggio) e il numero di partecipanti da cancellare (non è necessario effettuare controlli su tale numero).

Sviluppare le classi che permettono una soluzione basata sulla programmazione Object Oriented al problema.