

Nome e cognome: _____ Matricola: _____

Programmazione ad oggetti mod.1, esame del 14/01/2022

PROVA DI PROGRAMMAZIONE

Durante questa prova si possono utilizzare il libro di testo, appunti o altro materiale (come, ad esempio, la stampa delle slide del corso). È invece proibito l'utilizzo di dispositivi elettronici quali, ad esempio, computer portatili, smartphone ed e-reader. Le risposte devono essere date scrivendo il codice sotto il testo dell'esercizio e, se necessario, sul retro del foglio. La durata della prova è di 60 minuti.

Nello svolgimento degli esercizi:

- È necessario massimizzare sempre l'information hiding,
- Le classi possono implementare metodi aggiuntivi se ritenuto utile o necessario, ma questi non devono essere visibili al di fuori della classe stessa.

Esercizio 1 (7 punti): GreenPass

Si implementi una classe astratta GreenPass che contenga il codice fiscale di una persona (rappresentato attraverso una stringa) e la data di scadenza (rappresentato come un oggetto di tipo Date del package java.util). Una volta creato un green pass, non deve essere più possibile modificare né il codice fiscale né la data di scadenza. Deve però esserci la possibilità di accedere alle informazioni (codice fiscale e scadenza) anche al di fuori della classe.

Si implementino poi una classe VaccinoGreenPass che estende la classe GreenPass e rappresenta un GreenPass ottenuto tramite vaccino. Tale classe verrà utilizzata all'interno di collezioni (Collection) successivamente.

Negli esercizi seguenti si assuma che oltre alla classe VaccinoGreenPass ne sia stata implementata un'altra chiamata TamponeGreenPass che rappresenta un green pass ottenuto tramite tampone, e con un'implementazione analoga a quella di VaccinoGreenPass.

Esercizio 2 (4 punti): PersonaConGreenPass

Si implementi la classe `PersonaConGreenPass` che contiene il codice fiscale (sempre rappresentato come una stringa) di una persona, e una collezione (`Collection`) di green pass. Quando istanziata, la classe non contiene alcun `GreenPass`. La collezione di `GreenPass` deve poter essere accessibile pubblicamente tramite un metodo `getter`.

Si implementi poi un metodo pubblico `addGreenPass` che riceve un oggetto di tipo `GreenPass` e lo aggiunge alla collezione. Se uno stesso `GreenPass` viene aggiunto due volte deve essere contenuto una sola volta nella collezione. Nel caso in cui il green pass aggiunto fa riferimento ad un codice fiscale diverso, deve essere lanciata una eccezione `PersonaSbagliataException` implementata come segue.

```
public class PersonaSbagliataException extends Exception {}
```

Esercizio 3 (4 punti): ControlloGreenPass

Si implementi una classe `ControlloGreenPass` che implementa due metodi statici `haGreenPassNormale` e `haGreenPassSuper` che ricevono una `PersonaConGreenPass` e ritornano un valore booleano `true` se e solo se tale persona è coperta da un qualsiasi `GreenPass` o se è coperta da un `VaccinoGreenPass` per la data odierna, rispettivamente.

La data odierna può essere ottenuta invocando la seguente catena di metodi `Calendar.getInstance().getTime()` che ritorna un oggetto di tipo `Date`. Il confronto tra date invece (per sapere se una data è prima o dopo di un'altra) può essere effettuato attraverso i metodi seguenti della classe `Date`.

`boolean`

`after(Date when)`

Tests if this date is after the specified date.

`boolean`

`before(Date when)`

Tests if this date is before the specified date.