

Esame di Programmazione Orientata agli Oggetti - 3 febbraio 2023

Si definisca un insieme di classi e interfacce per rappresentare la gestione della visite di un Polo Museale

Ogni visitatore è caratterizzato dai dati personali **nome**(String), **età** (int), un **biglietto** (int), un **identificatore** univoco assegnato progressivamente in fase di acquisto, un **prezzo base** del biglietto e deve essere possibile ottenere in ogni momento il **costo complessivo** dell'utente.

Il costo complessivo del biglietto si calcola in base al prezzo base, servizi a pagamento acquistati ed eventuali agevolazioni.

I visitatori si differenziano nelle seguenti categorie:

- **adulti** (età in anni compresa tra 10 e 72 inclusi),
- **anziani** (età>72),
- **bambini** (età<10,)

I bambini e gli anziani, possono essere Agevolati, cioè il costo del biglietto è scontato secondo il seguente schema:

- Ai bambini si applica una detrazione del 50% del costo complessivo del biglietto
- Agli anziani si applica una detrazione del 30% solo sul prezzo base del biglietto

Sono previsti i servizi a pagamento che possono essere in alcuni casi accessibili solo a specifiche categorie di utenti. Qui consideriamo solo i servizi:

1. **PercorsoGuidato**: è un servizio a pagamento che può essere solo per adulti e anziani. Il tragitto guidato può essere di 3 livelli diversi L (L = 1..3 che corrispondono a visita guidata breve di gruppo, visita guidata lunga di gruppo, visita guidata individuale) con costi diversi: 5*L euro.
Solo i visitatori che ne hanno diritto possono invocare un metodo **setGuida** che semplicemente incrementa il costo del biglietto in base al livello scelto (e per nostra comodità di test esegue la scrittura su System.out di " accompagnatore livello <n° livello> per il visitatore " + <nomeVisitatore>) e un metodo **getLivello** che restituisce la tipologia di servizio (il livello) acquistato.
nota: si può restituire 0 se il visitatore non ha acquistato questo servizio.
2. **Audioguida**: è un servizio a pagamento (2 euro) solo per gli adulti e gratuito per bambini e anziani:
il servizio descriverà le opere in mostra in modalità diversa per bambini e altri e può essere avviato solo dagli utenti che ne hanno diritto in invocando il metodo opportuno che qui è semplicemente descritto dalla scrittura su System.out di una frase personalizzata di benvenuto (es. "buongiorno piccolo <nome> oggi ti raccontiamo la storia di <String che rappresenta il topic della mostra> (es. "vecchi faraoni d'Egitto...."), oppure analogamente "buongiorno sig <nome>....") prevedere metodi **setAudioguida** (per ottenere il servizio), e **getAudioguida** (per conoscere se questo visitatore utilizza il servizio)
3. **AccompagnatorePedagogo**: i bambini possono accedere ad un servizio di accompagnamento e guida fatto apposta per permettere una fruizione adeguata alle loro esigenze. Questo servizio è a pagamento (4 euro). Dunque il visitatore bambino che chiede questo servizio (**setPedagogo**) incrementa il costo del biglietto ed esegue la scrittura su System.out di " accompagnatore pedagogo: " +

<nomeVisitatore>.

Anche in questo caso occorre un metodo per sapere se il visitatore ha acquistato questo servizio

Gestione della biglietteria

Supponiamo per semplicità che l'acquisto del biglietto segni anche l'inizio della visita al museo. Il polo museale mantiene tutte le informazioni relative ai visitatori attualmente in visita al museo in una mappa con chiave il numero biglietto e valore il Visitatore.

Ogni volta che un visitatore accede al polo museale viene staccato un nuovo biglietto per il visitatore e viene inserita nella Mappa una nuova entry con chiave il nuovo numero biglietto e il visitatore con il biglietto assegnato. L'aggiunta dei servizi (se c'è) sarà una operazione successiva. Quando il visitatore lascia il museo il suo biglietto viene estratto dalla Map.

Scrivere le operazioni per

1. Costruire la Map
2. Inserire visitatori delle diverse tipologie trattate
3. Spampare a video il contenuto della Map
4. Restituire (se c'è) il visitatore che ha un biglietto con un dato identificativo
5. Estrarre dalla Map il visitatore che ha un biglietto con un dato identificativo
6. Valutare l'attuale flusso di visitatori (quanti visitatori popolano il polo adesso)
7. Valutare il numero di bambini attualmente in visita al polo e stamparli

Comportamento Multithread rappresentativo del flusso di visitatori nel museo:

Per la gestione Multithread semplifichiamo il modello di visitatore riducendolo a una classe caratterizzata di identificatore biglietto id (ancora una volta un intero assegnato progressivamente) e nome del visitatore assegnato come stringa "Visitatore"+id e definiamo il flusso dei visitatori come una lista che è un bounded buffer di capacità massima 15.

Supponiamo ci siano più punti di accesso e più punti di uscita dal museo.

il flusso dei visitatori, visitatori gestito da cinque thread:

- 2 input gate thread (porte di accesso al polo) si comportano da Produttori:
 - inseriscono nel buffer un numero random compreso tra 1 e 10 di nuovi visitatore
 - vanno in sleep per un periodo random compreso tra 3 e 5 sec
- 1 thread si comporta da Lettore (monitora il flusso di visitatori attuale):
 - legge e stampa a schermo l'intero contenuto della collection
 - va in sleep per un periodo 20 sec
- 3 output gate thread (porte di uscita dei visitatori) si comportano da Cancellatori:
 - cancellano un numero random compreso tra 1 e 5 visitatori contenuti nella collection (stampandoli il output)
 - vanno in sleep per un periodo random compreso tra 3 e 5 sec

Prevedere la gestione di almeno 3 tipi (definizione, sollevamento e gestione) di eccezione

Prevedere opportuni main che dimostrino la correttezza delle soluzioni proposte

ATTENZIONE! LA PROVA VA SALVATA DENTRO UNA CARTELLA CON NOME:
[NUM MATRICOLA][COGNOME][NOME] (es. 046000001ROSSIPAULO)
e inviata su STUDIUM entro il tempo di scadenza della prova
