Programmazione ad Oggetti
Traccia Laboratorio 15 novembre 2019

Esercizio 1.
Realizzare la classe Utente che contiene almeno i seguenti campi:

- int numero;
- int tipologiaOperazione;

dove è possibile avere solo tre tipologie di operazioni: 0 per spedizione, 1 per pagamento, 2 per riscossione. La scelta dei metodi da inserire nella classe Utente è libera.

Realizzare la classe Posta che permetta di gestire i clienti in coda in un ufficio postale. La classe deve contenere al suo interno un campo di tipo list<Utente>.

a) void aggiungiUtente(int tipologiaOperazione); aggiunge un utente in lista di attesa. Il numero dell'utente è progressivo rispetto al numero di utenti in attesa per quella determinata operazione.
b) Utente prossimoUtente(); restituisce il prossimo utente e lo rimuove dalla lista di attesa. La precedenza tra gli utenti è stabilita come segue:
    - Prima tutti gli utenti che devono effettuare una spedizione in ordine di arrivo.
    - Se non ci sono utenti che devono effettuare spedizioni, si alternano gli utenti che devono effettuare pagamenti e riscossioni.
    - Nel caso in cui tutti gli utenti rimasti in attesa debbano effettuare la stessa operazione, si servono in base all'ordine di arrivo.
c) void stampaUtentiInCoda() const; stampa tutti gli utenti in ordine di priorità.
d) void stampaUtenti(int tipologiaOperazione) const; stampa tutti gli utenti in attesa per una determinata tipologia di operazione.
e) void reset(); rimuove tutti gli utenti.

Infine, realizzare un main di prova che permetta l'interazione con un utente utilizzando un'interfaccia testuale. La schermata iniziale deve contenere la seguente interfaccia:

==== Menu ====
Premi 1 per aggiungere un utente specificando la tipologia di operazione
Premi 2 per selezionare il prossimo utente da servire
Premi 3 per stampare gli utenti in coda
Premi 4 per stampare gli utenti in coda per una determinata operazione
Premi 5 per effettuare un reset
Premi 9 per uscire

CONTINUA A PAGINA 2.

Dipartimento di Matematica e Informatica
Università della Calabria
Via P. Bucci, Cubo 30B
87036 Rende (Cs)

Tel. (+39) 0984.49.6402 - 6400
Fax (+39) 0984.49.6410
direttore.demacs@unical.it
dipartimento.demacs@pec.unical.it

www.unical.it

Esercizio 2.
Realizzare una classe Generatore che gestisca la generazione casuale di numeri. La classe deve contenere un campo privato list<int> e i seguenti metodi:

a) int generaNumero(); genera un numero casuale tra -1000 e +1000, lo aggiunge alla lista interna e lo restituisce. Nel caso in cui un numero sia già stato inserito 10 volte nella lista, allora ne genera uno nuovo. Nel caso non sia possibile generare nuovi numeri, restituisce INT_MAX.
b) int numeroFrequente() const; restituisce il numero che appare più frequentemente nella lista. Nel caso in cui ci siano più numeri con questa proprietà, restituire il primo in ordine di apparizione nella lista. Se non ci sono numeri restituire INT_MAX.
c) int generaNumeroPrimo () const; genera un numero casuale tra 0 e 10000 che sia primo. Il numero generato non deve essere inserito all'interno della lista.
d) void cancellaNumero(int num, int n); elimina le prime n occorrenze del numero num all'interno della lista.

Esercizio 3.
Realizzare la classe Film che contenga i seguenti campi privati: string titolo; int anno; int incasso; string regista; Genere genere, dove Genere può essere un enumerativo con i seguenti valori: NONDEFINITO, ANIMAZIONE, COMICO, COMMEDIA, DRAMMATICO, FANTASY, HORROR, STORICO.

Realizzare la classe Cinema contenente un campo privato list<Film*> e i seguenti metodi:

- void aggiungi(Film*); aggiunge un elemento all'interno della lista.
- Genere migliorGenere() const; restituisce il genere di film che ha incassato di più tra quelli presenti nella lista (escludendo i film per cui il genere è NONDEFINITO). Se due generi diversi hanno incassato lo stesso importo massimo restituire il primo in ordine alfabetico (esempio, ANIMAZIONE precede COMICO). Se non sono presenti film o se tutti i film hanno il genere NONDEFINITO, restituire NONDEFINITO.
- string registaStanco() const; restituisce il regista meno attivo di recente. In particolare, si consideri A come l'anno più recente di un film nella lista. Restituire il regista che ha diretto meno film nell'anno A. Se due o più registi hanno diretto lo stesso numero minimo di film nell'anno A, restituire il primo in ordine alfabetico. Se non sono presenti film, restituire "-1".
- int registiSettoriali() const; restituisce il numero di registi settoriali. Un regista è detto settoriale se almeno il 70% dei suoi film sono dello stesso genere (includendo il genere NONDEFINITO). Se non sono presenti film, restituire -1. Esempio: Un regista che ha diretto due film di ANIMAZIONE e un film COMICO non è settoriale (67% dei film di ANIMAZIONE e 33% COMICO). Un regista che ha diretto tre film di tipo COMICO e un film COMMEDIA è settoriale (75% dei film di COMICO e 25% COMMEDIA).
- int differenzaIncassoMaggiore() const; restituisce la differenza di incasso più alta tra i film presenti nella lista. Data una qualsiasi coppia di film F1, F2 la differenza di incasso tra i due film è l'incasso di F1 - l'incasso di F2. Se sono presenti meno di 2 film nella lista, restituire -1.