

Fondamenti di Programmazione 2

November 4, 2022

Esercizio 1

Realizzare la classe `Documento` che contiene almeno i seguenti campi:

- `string proprietario`
- `string descrizione`
- `int dimensioneFile`

Realizzare la classe `GestoreDocumenti` che permette di gestire alcuni documenti di diverse tipologie. La classe deve contenere al suo interno un campo di tipo `list<Documento*>`:

- `void aggiungiDocumento(Documento*)`: aggiunge un documento alla lista dei documenti
- `void rimuoviDocumento(Documento*)`: rimuove un documento dalla lista dei documenti
- `void stampaDocumenti() const`: stampa il proprietario e la descrizione di tutti i documenti
- `void ordinaDocumenti(int valore)`: ordina i documenti come segue:
 - se `valore` è pari a 1, ordina in base al proprietario
 - se `valore` è pari a 2, ordina in base alla descrizione
 - se `valore` è pari a 3, ordina in base alla dimensione del file
 - in tutti gli altri casi (`valore < 1` o `valore > 3`) non fa operazioni

Realizzare le classi `DocumentoPDF` o `DocumentoMP3` che ereditano da `Documento`. La classe `DocumentoPDF` deve contenere anche un valore booleano che indichi se il documento è stato firmato oppure no. La classe `DocumentoMP3` deve contenere anche un valore `float` che indichi la durata.

Infine, realizzare un `main` di prova che interagisca con la classe `GestoreDocumenti`. In particolare nel `main` si dovranno creare 5 documenti di tipo `DocumentoPDF` e 5 di tipo `DocumentoMP3` e usare tutti i metodi della classe `GestoreDocumenti`.

Esercizio 2

Implementare una gerarchia di classi che modelli le figure geometriche **Cerchio**, **Quadrato**, **Rettangolo** implementando (almeno) la seguente interfaccia:

```
class Shape {
public:
    /* Restituisce il perimetro della figura */
    virtual double perimeter() const = 0;

    /* Restituisce l'area della figura */
    virtual double area() const = 0;
};
```

Scrivere successivamente un **main** di prova che, letti in input due interi n , k :

- Legga da input n figure geometriche e le salvi in un **vector<Shape*>**
- Stampi le prime k figure geometriche con *densità* maggiore, dove per densità si intende il rapporto tra area e perimetro della figura

Esercizio 3

La nota catena alberghiera “Olliver Hint” ha intenzione di aprire una nuova filiale in Calabria. L’Università della Calabria offre il proprio contributo implementando il sistema di gestione della nuova filiale, tenendo conto delle seguenti specifiche.

L’hotel ha 3 tipologie di camere: singola, doppia, tripla. Ogni camera ha un proprio numero identificativo. Utilizzando ereditarietà e polimorfismo, realizzare delle classi rappresentanti i concetti di camera e le diverse tipologie.

A tal proposito dotare le classi di due metodi virtuali:

- **calcolaCosto**, per calcolare il costo delle camere, che varia in base alla tipologia: 30 euro per una camera singola, 50 euro per una camera doppia, 75 euro per una camera tripla.
- **tipologia**, che restituisca un numero identificativo della tipologia: 1 per le camere singole, 2 per le camere doppie, 3 per le camere triple.

I gestori dell’hotel hanno poi bisogno di registrare le prenotazioni effettuate dai propri clienti. Ogni prenotazione riguarda una camera, per cui la classe **Prenotazione** avrà come dato un puntatore a camera, per poter usufruire del polimorfismo. Inoltre contiene alcune informazioni sul cliente che ha effettuato la prenotazione: nome, cognome, codice della carta di credito, e se ha fatto richiesta per la colazione. Tale richiesta comporta una maggiorazione di 10 euro aggiuntivi al prezzo della camera. Modellare, quindi il concetto di prenotazione tramite un’apposita classe.

Realizzare infine, una classe **GestoreHotel**, che abbia come dati due liste:

- Una lista di camere, rappresentanti le camere disponibili: in particolare, dovrà essere una lista di puntatori a camere, per poter usufruire del polimorfismo;
- Una lista di prenotazioni.

La classe deve permettere di:

- Aggiungere e rimuovere camere all'hotel;
- Aggiungere una prenotazione per un tipo specifico di camera;
- Rimuovere una prenotazione relativa ad un cliente;
- Calcolare l'importo di una prenotazione;
- Calcolare il totale degli importi di tutte le prenotazioni;
- Visualizzare informazioni riguardanti le camere disponibili e le prenotazioni.

Verificare attraverso un **main**, che il sistema per la gestione funzioni correttamente, creando diversi tipi di camere e alcune prenotazioni. Se nel main si fa uso di memoria dinamica per la creazione delle camere, è bene che sia il main ad occuparsi anche della de-allocazione.

Esercizio 4

Progettare e implementare una classe **AnagraficaUniversitaria** che si occupi di gestire gli studenti e i professori di un'università. La classe deve avere i seguenti metodi:

- **void aggiungiStudiante(string codiceFiscale, string nome, string cognome, float isee):** aggiunge lo studente all'anagrafica se non è già presente il codice fiscale
- **void aggiungiProfessore(string codiceFiscale, string nome, string cognome, float stipendio):** aggiunge il professore all'anagrafica se non è già presente il codice fiscale
- **Persona* getPersona(string codiceFiscale) const:** restituisce la persona con il codice fiscale passato come parametro. Nel caso non ci siano persone con quel codice fiscale, restituisce **nullptr**
- **void stampa() const:** stampa tutte le persone presenti nell'anagrafica

Progettare e implementare una classe **Mensa** che permetta di gestire i dati riguardanti le operazioni giornaliere di una mensa universitaria. La classe deve avere i seguenti metodi:

- **void accesso(Persona* p):** aggiunge la persona **p** alle persone che hanno usufruito della mensa in quel giorno. Una persona non può accedere due volte alla mensa lo stesso giorno
- **void stampa() const:** stampa tutte le persone che hanno usufruito della mensa in quel giorno

- `void nuovoGiorno()`: resetta i dati della mensa per far partire un nuovo giorno
- `float calcolaIncassoGiornaliero() const`: calcola l'incasso giornaliero tenendo conto delle seguenti tariffe:
 - Studenti:
 - * Per `ISEE <= 10000` la tariffa è 0€
 - * Per `10000 < ISEE <= 20000` la tariffa è 2€
 - * Per `20000 < ISEE <= 40000` la tariffa è 3€
 - * Per `ISEE > 40000` la tariffa è 4€
 - Professori:
 - * Per `Stipendio <= 30000` la tariffa è 2.5€
 - * Per `30000 < Stipendio <= 40000` la tariffa è €
 - * Per `20000 < ISEE <= 40000` la tariffa è 3€
 - * Per `Stipendio > 40000` la tariffa è 5€
 - Tutte le altre persone: 7€

Nota: la classe **Mensa** deve essere abbastanza flessibile da non essere modificata nel caso in cui nuove figure universitarie abbiano accesso alla mensa ad un prezzo scontato.