Gestione di una Concessionaria di Auto

Esercizio di Programmazione in C++

29 Novembre 2024

Obiettivo

In questo esercizio, gli studenti devono implementare un programma C++ per gestire il magazzino e gli ordini di una concessionaria di auto. Il programma deve:

- Leggere i dati iniziali delle auto da un file di testo.
- Memorizzare i dati in una struttura interna.
- Permettere la visualizzazione del magazzino.
- Consentire la rimozione di un'auto dal magazzino in seguito a un ordine.
- Salvare lo stato aggiornato del magazzino in un file di testo.

Struttura del Programma

Il programma deve essere strutturato seguendo i principi della programmazione orientata agli oggetti (OOP). Gli studenti devono:

1. Creare una classe astratta `Vehicle`, che rappresenta un veicolo generico.
2. Implementare una classe concreta `Car`, che eredita da `Vehicle`.
3. Creare una classe `Dealership` per gestire il magazzino.

Suggerimenti per l'Implementazione

Per aiutare nella lettura dei dati dal file di testo, ecco il metodo `loadInventory`, che utilizza i token per leggere i campi separati da virgole.

Metodo `loadInventory`

```cpp
void loadInventory(const std::string& filename) {
    std::ifstream file(filename);
    if (!file) {
        std::cerr << "Errore nell'apertura del file: " <<
            filename << std::endl;
        return;
    }

    std::string line;
    while (std::getline(file, line)) {
        std::istringstream iss(line);
        std::string brand, model, yearStr, priceStr;

        // Estrae i campi separati da virgole
        if (std::getline(iss, brand, ',') &&
            std::getline(iss, model, ',') &&
            std::getline(iss, yearStr, ',') &&
            std::getline(iss, priceStr, ',')) {

            // Converte i campi numerici
            int year = std::stoi(yearStr);
            double price = std::stod(priceStr);

            // Crea un oggetto Car e lo aggiunge al
               magazzino
            inventory.push_back(std::make_shared<Car>(
               brand, model, year, price));
        } else {
            std::cerr << "Riga non valida nel file: " <<
               line << std::endl;
        }
    }
}
```

Esempio di File di Input

Il file di testo da leggere deve avere il seguente formato:

```
Ford,Focus,2018,15000
Toyota,Corolla,2020,18000
BMW,320i,2019,25000
```

`Audi,A4,2021,35000`

Ogni riga rappresenta un'auto con i seguenti campi separati da virgole:

1. Marca (`brand`)
2. Modello (`model`)
3. Anno (`year`)
4. Prezzo (`price`)

Compiti per gli Studenti

1. Creare le classi `Vehicle` e `Car` per gestire i dettagli delle auto.
2. Implementare la classe `Dealership`, che includa i metodi per:
   - Caricare il magazzino dal file (`loadInventory`).
   - Visualizzare il magazzino.
   - Rimuovere un'auto dal magazzino quando viene ordinata.
   - Salvare il magazzino aggiornato in un file.
3. Scrivere il `main` per gestire il programma attraverso un menu.

Estensioni Facoltative

Per gli studenti più esperti, si possono aggiungere le seguenti funzionalità:

- Ordinare il magazzino in base al prezzo o all'anno.
- Permettere la ricerca di auto per marca o modello.
- Gestire eccezioni specifiche durante la lettura del file.

Valutazione

Il programma verrà valutato in base a:

- Correttezza della struttura OOP.
- Funzionalità dei metodi richiesti.
- Gestione corretta dei file di input e output.
- Qualità del codice (chiarezza, commenti, stile).