

Documentazione del esercizio

Descrizione: Esercizio affidatomi da parte di Coolshop.it per verificare le mie competenze

Autore: Istratoaie Mircea George

Data creazione: 19/12/2023

Data ultima modifica: 22/12/2023

Descrizione del problema:

Si implementi uno script da eseguire da linea di comando, utilizzando a piacimento uno tra **NodeJs** e **C#**.

Lo script riceve in input il percorso di un file csv da utilizzare, contenente la lista di ordini di un sito ecommerce.

Lo script deve dare in output i seguenti dati:

- Record con importo totale più alto
- Record con quantità più alta
- Record con maggior differenza tra totale senza sconto e totale con sconto

Risoluzione:

Ho scelto l'utilizzo di **C#** per il completamento di questo esercizio.

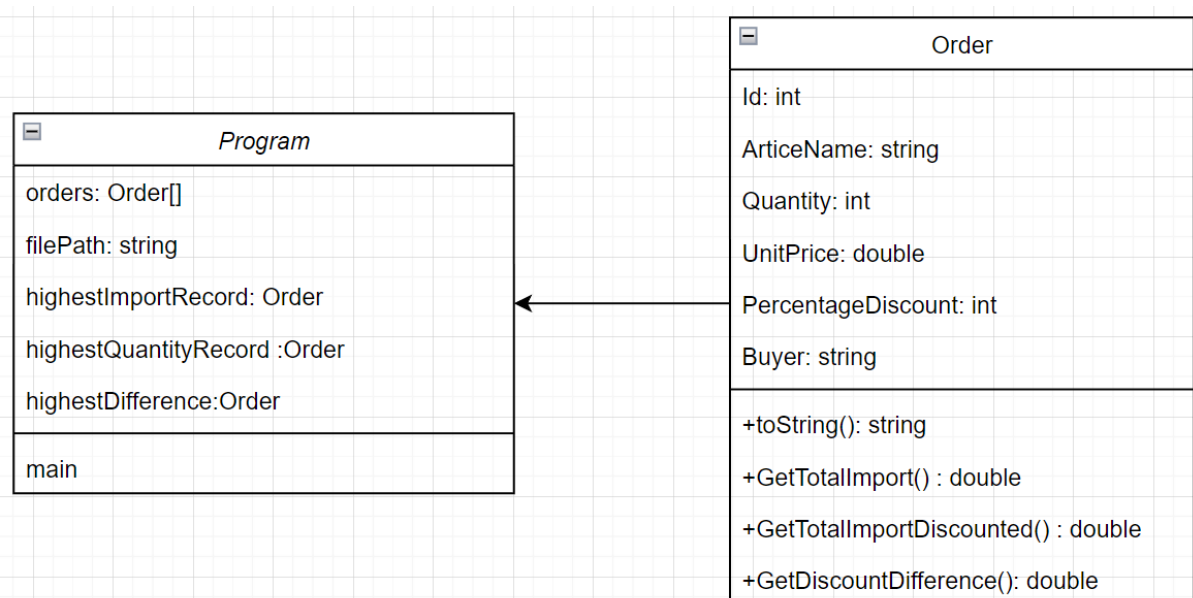
Come strumento ho usato **Visual Studio 2022** Community.

Il mio **GitHub** [<https://github.com/Lyfriss>]

Il perno centrale per la risoluzione verte sull'utilizzo della libreria **CsvHelper**

[<https://joshclose.github.io/CsvHelper/>] la quale fornisce strumenti semplici ed efficaci per poter leggere e scrivere i file .csv.

Struttura codice:



I singoli records del file vengono trattati come istanze della classe **Order**.

Per ogni parametro del record vi è un attributo nella classe provvisto di get e set.

Nota è necessario specificare il "[Name(")]" poiché i nomi degli attributi sono diversi dagli header del file csv

I metodi di Order sono:

- **ToString:** Restituisce una stringa con la quale visualizzare a schermo il record
- **GetTotalImport:** Restituisce l'importo senza alcuno sconto
- **GetTotalImportDisconted:** Restituisce l'importo a cui viene applicato la percentuale dello sconto
- **GetDiscountDifference:** Restituisce la differenza tra l'importo con sconto e quello senza

Nel main si procede

1. Creare il vettore di Order orders il quale conterrà i record del file csv
2. Creare una configurazione nella quale impostiamo HasHeaderRecord = true poiché nel file csv fornito il primo record contiene gli header.
3. Chiedere in input l'indirizzo del file e salvarlo nella variabile filePath.
4. Grazie alla libreria CsvHelper possiamo utilizzare il metodo csv.GetRecords<Order>().ToArray() col quale immagazziniamo i record nel vettore
5. Visualizziamo il vettore a schermo
6. Creiamo le tre variabili necessaria a ricavare i record richiesti in consegna
 - a. highestImportRecord = orders[0];
 - b. Order highestQuantityOrder = orders[0];
 - c. Order highestDifference = orders[0];
7. Nel ciclo assegnamo a ciascuna variabile il valore corretto
8. Mostriamo a schermo i tre record voluti