

PROGETTO 1

ESAME DI PROGRAMMAZIONE 2

Studente: *Nunzio Depetro*

Matricola: *X81000913*

Quesito:

Il file 'Comande.txt' contiene le informazioni sugli ordini effettuati in una pizzeria nel seguente formato

COGNOME_CLIENTE, PIZZA, QUANTITA, PREZZO_UNITARIO

I campi `COGNOME_CLIENTE` e `PIZZA` sono dati di tipo testuale, il campo `QUANTITA` è un intero e

il `PREZZO_UNITARIO` è un numero con la virgola. Si assuma che non ci siano ripetizioni nei cognomi.

Il candidato scriva un programma in C++ che, utilizzando le opportune strutture dati, sia in grado di:

1. Caricare le informazioni presenti nel file;
2. Raggruppare le ordinazioni per cliente;
3. Cercare tutte le ordinazioni di un dato cliente, stamparle a video e calcolare il totale speso;
4. Eliminare dalle ordinazioni tutti i clienti che hanno speso meno della media; - (BONUS) Salvarle su un file;
5. BONUS: Stilare una classifica delle pizze più vendute.

Fornire una funzione `main` che esegua correttamente i metodi richiesti nei punti sopra.

Fornire inoltre un documento sintetico in formato PDF che spieghi le scelte progettuali, motivando le

scelte relative alle strutture dati e agli algoritmi utilizzati.

NB: Il progetto verrà compilato ed eseguito utilizzando il comando `g++`.

Implementazioni “s sofisticate”, o compilazioni gestite da IDE esterni, comporteranno altrettante capacità di utilizzo di `g++` da parte dello studente che le propone.

SCELTE PROGETTUALI

Struttura dati utilizzata: List.

L'utilizzo della struttura [List](#) nasce dal fatto che è fondamentale utilizzare una struttura dati dinamica in modo da allocare all'interno della stessa gli oggetti contenuti nel file.txt.

La necessità di dover comparare più elementi suggerisce l'overload degli operatori di comparazione.

Librerie utilizzate:

`<iostream>` : Libreria per l'utilizzo degli oggetti cout, cin, etctera...

`<fstream>` : Per la lettura del file txt e per la scrittura del file "Clienti Eliminati"

`<string>` : Libreria per l'utilizzo delle variabili string

`<sstream>` : Libreria per lo stream delle stringhe.

File Header:

Cliente.h
ListaDL

File Implementazioni:

ListaDL.cpp
Cliente.cpp
main.cpp

La scelta della creazione dei file [ListaDL.cpp](#) e [Cliente.cpp](#) è stata fatta in maniera arbitraria e per mantenere un determinato ordine nella scrittura, L'alternativa era dichiarare ed implementare la classe direttamente all'interno dell'header.

Classi utilizzate:

-*Cliente.d* && *Cliente.cpp*

-ListaDL.h && ListaDL.cpp

Nella classe *Cliente.h* ho inizializzato tutta la classe cliente passandogli i parametri delle ordinazioni (Cognome,Pizza,Quantita,Prezzo) e ho dichiarato i metodi della classe.

Nella classe *Cliente.cpp* ho implementato i metodi contenuti nel file header .

Nella classe *ListaDL.h* ho inizializzato una lista doppiamente linkata con tutti i suoi metodi (inserimento,ricerca,rimozione)

Nella classe *ListaDL.cpp* ho implementato i metodi contenuti nel file header.