Report

Motivazione scelte ADT

Progettazione

Main

Il programma si apre con la visualizzazione di un menù interattivo che guida l'utente nella selezione delle varie funzionalità disponibili. Per garantire un'adeguata modularità e aderire al principio dell'information hiding, le funzionalità sono state suddivise logicamente in più file sorgente, ciascuno dedicato alla gestione di uno specifico aspetto dell'applicazione.

All'interno del file main.c, oltre alla chiamata alla funzione menu() che stampa il menù principale, sono presenti due funzioni ausiliarie:

- una per pulire il buffer di input (pulisci_input) per gestire correttamente l'interazione con l'utente,
- l'altra per inserire una pausa tra le operazioni (attendi_utente), bloccando temporaneamente l'esecuzione fino alla pressione del tasto INVIO.

La logica di selezione delle opzioni nel menù è gestita tramite una struttura switch, che associa a ciascun valore numerico l'operazione corrispondente.

Tutti i dati inseriti o modificati durante l'esecuzione del programma vengono persistiti su file, così da poter essere recuperati anche nelle sessioni successive.

abbonamenti.c

In questo modulo vengono implementate tutte le funzioni relative alla gestione degli abbonamenti. Gli abbonamenti sono memorizzati all'interno della struttura Cliente, che comprende sia i dati anagrafici sia due campi specifici per l'abbonamento:

- la data di inizio dell'abbonamento,
- e la durata in giorni.

Tutti i clienti vengono organizzati in un albero binario di ricerca (BST), ordinato in base al codice fiscale, scelto come chiave univoca.

Questa scelta strutturale consente accessi rapidi, inserimenti ordinati e ricerche efficienti. Sono state inoltre implementate funzioni per la cancellazione di nodi e la gestione del riordinamento dell'albero, oltre alla serializzazione dei dati in formato JSON, per facilitarne il salvataggio e la successiva lettura da file.

Lezioni.c

La gestione delle lezioni è affidata a un array dinamico, contenuto all'interno di una struttura CatalogoLezioni che ne memorizza anche la dimensione attuale e la capacità allocata.

Sono state implementate tutte le funzioni necessarie per:

- l'inserimento di nuove lezioni,
- la riallocazione automatica della memoria quando necessario,
- la rimozione di lezioni esistenti.

La scelta dell’array dinamico è stata motivata dal numero limitato e stabile di lezioni e dalla necessità di accessi frequenti e veloci agli elementi, rendendo questa struttura più adatta rispetto a liste o alberi.

Prenotazioni.c

Le prenotazioni sono gestite tramite una lista dinamica, in quanto tale struttura consente di gestire in modo flessibile aggiunte e rimozioni frequenti, che sono tipiche di questo tipo di dati (prenotazioni quotidiane, cancellazioni, aggiornamenti)

È stato inoltre stabilito un limite massimo di prenotazioni per fascia oraria, e sono state sviluppate funzioni booleane per verificare la disponibilità in una determinata fascia, prima di procedere all’aggiunta.

Il modulo include anche una funzione che mostra all’utente quali fasce orarie siano attualmente disponibili e quali invece siano al completo.

Utilities.c

Questo file contiene una raccolta di funzioni generiche e di utilità comune, come:

- la conversione e formattazione dell’orario,
- funzioni di gestione del tempo basate sulla libreria standard time.h,
- e altre operazioni ricorrenti impiegate in più moduli.

Persistenza_dati.c

Infine, la gestione della persistenza dei dati è stata centralizzata in un apposito modulo. Qui sono implementate tutte le funzioni necessarie per:

- il caricamento dei dati da file all’avvio del programma,

e il salvataggio automatico o manuale delle strutture aggiornate, assicurando che le informazioni inserite durante l’esecuzione vengano mantenute anche dopo la chiusura dell'applicazione.

Conclusione

Il programma è stato progettato con una forte attenzione alla modularità, alla chiarezza e alla manutenibilità del codice, rispettando i principi fondamentali della programmazione strutturata. La scelta delle strutture dati è stata effettuata in base alle caratteristiche specifiche di ciascun insieme di informazioni, con particolare attenzione alle operazioni più frequenti e ai vincoli funzionali del sistema.

Rappresentazione grafica di come i vari moduli funzionano tra di loro

File (Con Specifiche)