

MOTIVAZIONE DELLA SCELTA DELL'ADT

Sommario

MOTIVAZIONE DELLA SCELTA DELL'ADT	1
ADT.....	2
STRUTTURE DATI SCELTE	2
HASHTABLE CLIENTI	2
LISTA PER CORSI E PRENOTAZIONI.....	2

ADT

L'utilizzo degli ADT (Tipi di Dato Astratti) nel progetto ci consente di rappresentare in modo strutturato, modulare e sicuro le diverse strutture presenti nel programma, come iscritto, corso e prenotazioni.

L'utilizzo degli ADT inoltre offre vantaggi come:

- **L'Astrazione dei Dati:** consente di nascondere i dettagli interni e gestire i dati tramite operazioni ben definite.
- **Modularità:** facilita la manutenzione del codice e semplifica la ricerca degli errori.
- **Riusabilità del codice:** gli ADT creati per questo progetto, potrebbero essere riutilizzati in altri progetti, a seguito di qualche modifica.
- **Scalabilità:** l'aggiunta o la modifica di funzioni non richiede la riscrittura dell'intero programma.

STRUTTURE DATI SCELTE

HASHTABLE CLIENTI

Per la gestione dei clienti abbiamo scelto di utilizzare come struttura dati la tabella hash. Queste ci permette di avere accesso diretto agli elementi della tabella, tramite l'algoritmo di hashing è infatti possibile ottenere la posizione del dato nella tabella, dalla chiave primaria che abbiamo scelto, che nel nostro caso è "IDCliente".

Garantendo un accesso diretto, la tabella hash garantisce facilità nell'inserimento, cancellazione e ricerca, conoscendo la chiave, ma è possibile effettuare operazioni di ricerca anche senza conoscere la chiave stessa, ma scorrendo l'intera tabella, di fatti nel programma è presente più di una ricerca (per nome, cognome e durata dell'abbonamento) al di fuori della classica ricerca per ID.

Per quanto riguarda l'algoritmo di hashing abbiamo optato per un algoritmo già collaudato, l'algoritmo FNV-1a, scelto per l'essere deterministico, ovvero dallo stesso input produrrà sempre lo stesso output, ma anche per la sua generazione di chiavi, che riduce al minimo le collisioni.

Le collisioni sono gestite come delle liste, nel caso in cui due clienti producano lo stesso indice nella tabella, si andrà a creare una lista, il tutto è implementato già all'interno del codice.

LISTA PER CORSI E PRENOTAZIONI

Per i corsi e le prenotazioni abbiamo invece scelto di implementare delle liste, la scelta è dovuta al numero elevato di prenotazioni e di lezioni che si potevano presentare, la lista difatti risolve il problema dato che è un tipo di struttura che viene allocato dinamicamente, ma soprattutto può crescere e diminuire nel tempo. Altro vantaggio è il poter inserire e rimuovere un elemento dalla lista, conoscendo la

posizione, funzione usata soprattutto per la cancellazione, dato che per l'inserimento abbiamo optato per inserire in cima alla lista il nuovo elemento.

L'unica nota "dolente" nella lista è nella complessità di ricerca, nel peggior dei casi pari a $O(n)$, in quanto nel caso in cui dovessimo ricercare un elemento della lista, ci toccherà scorrerla per intero, a differenza della tabella hash che per l'accesso diretto ha una complessità pari a $O(1)$.