Progetto LASD

Shopping ONLINE-Documentazione

Marco Pastore N86003875

Funzionalità di base

Ho sviluppato un servizio di shopping online. Prima di accedere al sito ci sta un check iniziale per capire se l'utilizzatore del software è un cliente o un amministratore. Se l'utente risponde che è un cliente, gli viene posta a scelta se si deve registrare o meno. Per l'amministratore invece ci sta direttamente la funzione di login perché si ipotizza che abbia già un account.

Le funzionalità per un cliente e un amministratore sono:

Per il cliente:

1. Caricare soldi
2. Svuotare il conto
3. Riempire il carrello
4. Acquistare il carrello
5. Svuotare il carrello
6. Effettuare il logout

Per l'amministrazione:

1. Aggiungere prodotti al carrello
2. Modificare le scorte o i prezzi di un prodotto

Pulizia del codice

Il codice, secondo me, è abbastanza pulito. Faccio un check iniziale per capire se è un cliente o un amministratore e gestisco tutto con un grande if-then-else. All'interno dell' if ci sta la gestione della parte della registrazione del cliente e i vari switch case. All'interno dell'else invece le funzionalità , siccome sono poche (sono 2), le ho gestite in maniera sequenziali, molto compatto.

Modularità

Ho gestito ogni funzione in un .c per essere il più modulare possibile, il .h dove ho messo la struct per la definizione della lista e tutte le funzioni per la gestione del mio programma, anche se è unico, ho cercato di mettere dei commenti per separare e spiegare le macrocategorie

Interfaccia Grafica

E' un interfaccia molto semplice e chiara.

Struttura dati utilizzata

Io ho scelto la lista singolarmente linkata, perché:

scegliere liste doppiamente linkate o circolari, mi sembrava inutile, perché a me basta scorrere semplicemente la lista per trovare gli elementi

Adesso il campo si restringe, e si poteva scegliere o la lista singolarmente linkata o gli alberi.

Utilizzare gli alberi, ovviamente, è istruttivo, è la struttura dati più difficile da implementare e da utilizzare (per ora), ma non è detto che sia la più efficiente (sempre secondo me). Una volta che noi andiamo ad

allocare l'abero, nessuno ci garantisce che sia ben bilanciato. Ipotizziamo che sia abbastanza bilanciato, avremmo un costo che va tra il log(n) e l'n. Se avessi saputo fare algoritmi di bilanciamento, avrei scelto l'albero.

In conclusione, ho scelto la lista singolarmente linkata.

Manuale d'uso

Check iniziale per chiedere se sei un cliente o un amministratore (si risponde con una c o con una a)

Per il cliente:

Appena metti la lettera c di cliente, il programma ti chiede se sei registrato o no, se la risposta è N di no, ti fa effettuare la registrazione e poi successivamente ti fa mettere i dati per effettuare il login. Altrimenti se precedentemente si è effettuato una registrazione, basta mettere i dati.

Una volta entrati ci sta riportato il saldo in alto e poi le varie opzioni

Se si sceglie la 1 (caricare il soldi): puoi inserire una cifra.

Se si sceglie la 2 (svuotare il conto): ti viene posta un ulteriore domanda di conferma per svuotare il conto, se si risponde si, allora viene effettuata l'operazione

Se si sceglie la 3 (Riempire il carrello): ti viene mostrata la lista dei prodotti disponibile. Se il prodotto ci sta, viene caricato nel carrello, altrimenti in un file d'attesa.

Se si sceglie la 4 (Acquistare il carrello): Ti viene proposto il carrello, e se lo voui comprare, lo compri

Se si sceglie la 5 (Rimuovere il carrello): Rimuove il carrello

Se scegli la 6 (Effettuare il logout) effettua il logout

Per il l'amministratore:

In maniera sequenziale chiede prima se si deve aggiungere un prodotto tra quelli già disponibili.

Se si risponde no: chiede se vuoi modificare qualcosa di un prodotto (o la quantità o il prezzo)

ATTENZIONE: si può fare solo una delle due operazioni alla volta