

Università degli studi di Firenze



Laboratorio di programmazione

1° anno 2022/2023

Esercitazione n.9 gestione finanziaria

Cristiano Gaetano

07/09/2023

ANALISI DEL PROGETTO:	3
Scopo dell'esercitazione:	3
Funzionalità del progetto:	3
Analisi dei requisiti:	4
ARCHITETTURA DEL PROGETTO:	4
Organizzazione del progetto	4
Le classi principali e le relazioni	5
IMPLEMENTAZIONE:	6
Processo di scrittura del codice:	6
Problemi e soluzioni:	7
TEST E DEBUG:	8
Eventuali bug del programma:	8
LEZIONI APPRESE DAL PROGETTO:	8

ANALISI DEL PROGETTO:

Scopo dell'esercitazione:

Sviluppare classi che rappresentano transazioni finanziarie su un conto corrente (ingresso e uscita). Le classi devono essere in grado di leggere e salvare i dati su file.

Funzionalità del progetto:

Il progetto avrà varie funzioni, le fondamentali per il suo funzionamento sono:

1. **Iscrizione utente** -> Ci dovrà essere la possibilità di far iscrivere l'utente inserendo i suoi dati prima di creare una carta, per sapere a chi verrà intestata.
2. **Creazione carta** -> Collegata al primo punto, la creazione della carta ci permetterà di inserire i dati della carta voluta dall'utente e di eseguire le funzioni per la risoluzione dello scopo dell'esercitazione.
3. **Operazioni deposito/prelievo** -> Queste operazioni dovranno essere disponibili una volta inseriti tutti i dati dei punti precedenti. Qui l'utente potrà decidere se fare un'operazione di entrata o uscita su una determinata carta.
4. **Salvataggio dati in un file** -> Ogni volta che l'utente inserirà dei nuovi dati che siano dell'iscrizione user o della creazione di una carta, oltre a salvare i dati nel programma tramite certi strumenti, li salveremo anche in un file di testo all'interno della cartella del progetto.

Analisi dei requisiti:

Per la costruzione di questo progetto useremo soprattutto funzioni "basilari" del C++ come if-else, switch, do-while ecc... Dall'altra parte usufruiremo anche della OOP, ci baseremo principalmente sull'utilizzo dei vector per la loro dimensione dinamica con cui riusciremo a memorizzare i dati inseriti dall'utente per tutta l'esecuzione del programma.

ARCHITETTURA DEL PROGETTO:

Le classi principali e le relazioni

Per la programmazione del progetto abbiamo avuto bisogno di 4 classi:

1. **Finanza** -> E' la classe principale del progetto, al suo interno troviamo un vettore che può contenere oggetti di tipo Account; quest'ultimo verrà usato per memorizzare i dati degli utenti iscritti. A livello di metodi invece presenterà le funzioni per creare, modificare, cercare ed eliminare elementi (Utenti) dal vettore.
2. **Account** -> Questa classe invece avrà come attributi i dati che dovranno essere inseriti per l'iscrizione, modifica, eliminazione di un utente. Inoltre conterrà anche un vettore che potrà contenere oggetti di tipo Carta per memorizzare i dati delle carte. Anche questa avrà i metodi per creare, modificare, cercare ed eliminare elementi (Conti) dal vettore.
3. **Carta** -> Contiene gli attributi per andare a operare sulle carte, un vettore di tipo Transazione per memorizzare le transazioni eseguite e i getter/setter.
4. **Transazione** -> L'ultima classe invece, contiene solo gli attributi che vanno a costituire una transazione.
5. **Data** -> Classe aggiuntiva per salvare la data della transazione.

IMPLEMENTAZIONE:

Processo di scrittura del codice:

Per l'organizzazione della del codice ho seguito delle tappe precise:

1. Inizializzazione di tutte le classi e scrittura dei nomi dei metodi:

Ho iniziato il lavoro dichiarando tutte le classi che mi sarebbero servite con l'intestazione dei metodi al loro interno.

2. Completamento metodi registrazione / modifica / ricerca utente:

Come prima cosa ho iniziato a lavorare nella parte dedicata all'utente, ho implementato le funzioni registrazione, modifica e ricerca con l'inserimento dei dati anche su file.

3. Completata eliminazione utente, creazione / modifica / eliminazione e ricerca Carte:

Qui ho concluso la parte dell'utente e ho iniziato quella per le carte, implementate le stesse funzioni dell'utente.

4. Inizio lavoro per le transazioni, creazione della classe Data:

Dopo aver costruito le basi per andare a risolvere lo scopo del progetto, ho finalmente iniziato a lavorare sulla sezione delle transazioni. E' stata una parte abbastanza lunga ma senza troppi problemi, completata la fase della ricerca utente e ricerca carta.

5. Completamento della sezione transazioni:

Sono riuscito a inserire la transazione all'interno di un file txt e a cambiare i dati all'interno del programma.

6. Controllato il funzionamento attraverso il main:

Una volta concluso tutto il programma nel generale, ho costruito il main per verificare il corretto funzionamento di tutti i metodi.

7. Controllo generale del progetto:

Per concludere, ho fatto una revisione generale del progetto (Per i passaggi di valori e i const nei metodi).

8. Testing dei metodi principali:

Infine, ho costruito il testing su alcuni metodi.

Problemi e soluzioni:

Durante la programmazione del progetto ho riscontrato pochi problemi ma quelli che mi hanno bloccato per più tempo alla ricerca di una soluzione sono stati sicuramente quelli legati alle operazioni sui dati delle carte nel file di testo siccome non mi trovava la riga da modificare/eliminare. Alla fine ce l'ho fatta, sono riuscito a capire il problema vedendo la riga che il mio ciclo while andava a cercare nel file di testo, essa infatti presentava dei difetti che la rendevano impossibile da trovare nel file.

TEST E DEBUG:

Eventuali bug del programma:

Il programma in sé riesce ad eseguire tutte le funzioni richieste per la risoluzione del problema principale. Gli unici bug che possiamo trovare durante l'esecuzione del programma sono legati ad un inserimento sbagliato sulle variabili Int/String da parte dell'utente.

LEZIONI APPRESE DAL PROGETTO:

Come novità ho appreso principalmente il funzionamento del salvataggio dei dati su file di testo, una cosa che non avevo mai fatto fino a quel momento.