

PROGETTO 3

ESAME DI PROGRAMMAZIONE 2

Studente: Biagio Cubisino

Matricola: X81001028

QUESITO:

Si consideri il file 'Studenti.txt' di dati relativi a degli studenti, contenente i seguenti campi per ciascun record: matricola, cognome, nome, città di residenza e media voti. La media voti è un valore decimale. Il file si suppone già creato ed il numero di studenti non è noto. Tutti i valori sono separati da un punto e virgola. Queste informazioni devono essere recuperate e memorizzate in opportune strutture dati dinamiche, tenendo conto delle operazioni che vanno fatte sui record descritte in seguito.

Lo scopo del programma è quello di definire dei gruppi di studenti per una esercitazione di laboratorio. I criteri per definire i gruppi di studenti rispettano le seguenti regole:

- - Gli studenti con media maggiore o uguale a 24 appartengono ad un insieme A.
- - Gli studenti con media minore di 24 appartengono ad un insieme B.

La procedura che crea i gruppi estrae gli studenti dalle rispettive strutture dati di appartenenza in questo modo: gli studenti dell'insieme A vanno estratti in ordine di media voto crescente, quelli dell'insieme B vanno estratti in ordine inverso rispetto a quello con cui sono stati letti dal file.

Sia a uno studente estratto dalla struttura A e $b1$ e $b2$ due studenti estratti da B:

- - Se la somma dei voti di $b1$ e $b2$ è almeno 40, creare un gruppo formato dagli studenti $b1$ e $b2$, altrimenti, creare un gruppo composto dagli studenti a , $b1$ e $b2$.

- - Se non si hanno abbastanza studenti in A e/o B per applicare le regole precedenti, creare gruppi formati da almeno due studenti ciascuno, considerando tutti gli studenti rimasti.
Numerare i gruppi creati e stamparli a video, mostrando per ciascun gruppo:
 - - Il numero progressivo che identifica il gruppo
 - - Il numero di studenti che fanno parte del gruppo
 - - Matricola, cognome e media voto di ciascun membro
 - - Media dei voti degli studenti che fanno parte del gruppo
- L'implementazione deve contenere le classi con tutti gli attributi e metodi necessari al corretto funzionamento. Completare le classi con opportuni attributi e metodi in maniera arbitraria.
Prevedere inoltre un metodo main() utile a verificare i metodi di cui 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.
- Nota:** vi ricordo che il progetto deve essere fatto in modo tale da poter eseguire la compilazione ed il building su una macchina generica con il comando g++. Implementazioni “s sofisticate” comporteranno altrettante capacità di utilizzo del programma g++ da parte dello studente che le propone.

SCELTE PROGETTUALI

Struttura dati utilizzata: List,Stack

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.

L'utilizzo della struttura dati Stack invece è stata usata per estrarre gli studenti dell'insieme B in ordine inverso rispetto a come sono stati letti dal file.

Librerie utilizzate:

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

`<fstream>` : Per la lettura del file txt

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

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

File Header:

Studenti.h

Lista.h

Pila.h

File Implementazioni:

Lista.cpp

Studenti.cpp

Main.cpp

La scelta della creazione dei file *Lista.cpp* e *Studenti.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, come è stato fatto con *Pila.h*,

Classi utilizzate:

-Studenti.h && Studenti.cpp

-Lista.h && Lista.cpp

-Pila.h

Nelle classe [Studenti.h](#) ho inizializzato tutta la classe cliente passandogli i parametri degli studenti(Matricola,Cognome,Nome,Città,Media voti) e ho dichiarato i metodi della classe.

Nella classe [Studenti.cpp](#) ho implementato i metodi contenuti nel file header .

Nella classe [Lista.h](#) ho inizializzato una lista ordinata doppiamente linkata con tutti i suoi metodi (inserimento, ricerca, rimozione)

Nella classe [Lista.cpp](#) ho implementato i metodi contenuti nel file header.

Nella classe [Pila.h](#) ho inizializzato una pila con tutti i suoi metodi (push, pop), e ho implementato i metodi direttamente nel file header.