

# Esame di Programmazione II

Garro Simone

Num. Di Matricola: **1000014539**

**13/07/2021**

## Analisi del Problema

L'esercizio richiede di implementare diverse strutture dati per portare a termine il problema, che consentano di:

- Inserire elementi nelle strutture dati;
- Estrarli e riportarli in altrettante strutture dati.

Da ciò che viene dato nella consegna, trasuda che:

- Bisognerà leggere e caricare le informazioni da file;
- Estrarre i dati da essa e trasferirle in altre strutture dati, seguendo delle regole precise.

## Considerazioni Progettuali

- 1) La richiesta di inserire gli elementi da file in una struttura dati in modo inverso rispetto alla lettura del file;
- 2) L'inserimento in ordine crescente in base alla media;

Tenendo conto di tutto ciò, ho scelto di utilizzare come struttura dati una **Lista Doppia** **Concatenata Ordinata**, che contenga un riferimento alla **testa** e alla **coda**.

## Classi e File Usati

- **Studente**: classe base con attributi tipici di uno studente e relativi metodi per poterli richiamare.
  - **Overloading** dell'operatore << che permette di visualizzare correttamente tutti i dati relativi allo studente;
  - **Overloading** degli *operatori logici di comparazione*, usando come parametro di confronto la **media**.
- **Nodo\_DL<T>**: classe *Nodo* con implementazione template contenente gli indirizzi (**succ**, **pre**) ed il valore generico **T**, con annessi metodi per la gestione di essi;
- **ListaDL<T>**: classe *ListaDL* con implementazione template con richiamo della classe *Nodo\_DL<T>* per oggetti del tipo **testa** o **coda**, e i diversi metodi per la corretta realizzazione e uso della lista;

- **Gruppo:** classe *Gruppo* che include come header “**Studiante.h**” e “**Liste4.h**”, così da giostrare con le funzioni delle altre classi, in questa classe verranno implementati alcuni metodi per la creazione dei diversi gruppi da stabilire secondo vari criteri dati.

Come file escluso dall'utilizzo di classi, ho realizzato “**Richieste.h**” il quale contiene tutti i metodi necessari per portare a termine tutte le richieste del progetto e gli header delle precedenti classi per poter lavorare al meglio.

Tutte le operazioni effettuate nella lista avranno una complessità del tipo **O(n)**, ricordando che stiamo lavorando con una *Lista Doppiaemente Concatenata Ordinata*.