

Programmazione 2

Appello 07/07/2021

Progetto 3

Analisi del problema:

Fornito un file .txt contenente dati relativi ad alcuni studenti e dato in input al programma, la finalità del progetto è quella di creare un numero indefinito di gruppi di studio per una esercitazione di laboratorio. Seguendo però delle istruzioni ben precise:

- Una prima divisione deve essere fatta sulla base della loro media voti. Apparterranno all'insieme A coloro i quali hanno una media maggiore o uguale a 24, altrimenti apparterranno all'insieme B.
- Sulla base di questa prima separazione, bisogna costituire i gruppi di studio attingendo da entrambi gli insiemi A e B. Con l'aiuto quindi di funzioni apposite, estrarre gli studenti secondo le regole riportate nella consegna.
- Infine, costituiti i gruppi di studio, riportarli a video in modo distinto e con le informazioni riguardanti gli studenti che vi fanno parte.

Considerazioni Progettuali:

- La richiesta di costruire due insiemi A e B e da questi estrarre secondo regole ben precise, suggerisce la creazione di due liste da trattare come Pile. In questo modo si renderà omogeneo il codice dal punto di vista

delle strutture dati. Sia da A che da B si estrarrà dalla coda. Implementazione(LIFO). Al fine di evitare l'inizializzazione di una lista di supporto, la distribuzione nei rispettivi gruppi A e B avverrà in modo diretto durante la lettura in input di "Studenti.txt".

- La richiesta di creare gruppi formati da studenti, suggerisce la creazione di una Lista di Liste (Listone), il cui "NodoListone" ha come key una lista. In questo modo si avranno nodi contenenti liste che definiranno i rispettivi gruppi di studenti.
- La necessità di svuotare entrambe le Pile (A e B) al fine di creare i gruppi, spinge alla realizzazione di una piccola funzione di supporto che si accorga quando non è più possibile estrarre. Procedendo nello svuotare, come da consegna, si passerà a formare gruppi di almeno due studenti ciascuno tra quelli rimasti. Utilizzando un "approccio in Modulo 2", si estrarrà a gruppi di due studenti per volta con la possibilità di resto 0 oppure 1, come da modulo. Qualora rimanesse un solo studente, verrebbe puntualmente ricondotto attraverso un puntatore alla coda del "Listone" nel corretto gruppo.
- La consegna di inviare a video tutto l'elaborato, spinge alla manipolazione della Lista di Liste (Listone). Quindi si dovrà iniziare dalla testa di "Listone" ed accedere alla Lista all'interno del proprio "NodoListone" partendo anche in essa dalla testa (Listone->testa->key->testa). Utilizzando due while annidati, scorrere prima le liste interne ai nodi e solo alla fine passare al nodo successivo della struttura dati più esterna.

Classi utilizzate:

- **Nodo_Studente:** contiene, come dati "Private", tutte le informazioni necessarie a definire uno studente. In "Public", oltre al costruttore, tutte le funzioni di utilità atte a manipolare i dati privati.
- **Lista_Studenti:** contiene, come dati "Private", un puntatore alla testa e uno alla coda. In "Public" oltre a funzioni di manipolazione dati membro e costruttore, due funzioni di inserimento. "Insert_decr()" di Tempo Lineare, ed una "Insert()" di Tempo Costante. Per la distribuzione nei gruppi infine una funzione "Estrai()" di Tempo Costante.
- **Nodo_Listone:** contiene, come dati "Private", tutte le informazioni necessarie a definire una Lista di Studenti. In "Public", oltre al costruttore, tutte le funzioni di utilità atte a manipolare i dati privati.
- **Listone:** contiene, come dati "Private", un puntatore in testa e uno in coda. In "Public" oltre a funzioni di manipolazione dati membro e costruttore, una funzione "InsertListone()" di Tempo Costante.

Il "main()" si limita a inizializzare le Strutture Dati utili allo svolgimento del programma, prendere in Input il file "Studenti.txt" e richiamare le funzioni globali elencate di seguito:

- **Opera()** : gestisce l'input.
- **Estrai()**: gestisce i gruppi.

- `Print_Gruppi()`: manda a video l'elaborato.