

Esercitazione - Classi (310)

Fondamenti 2 - Corso di Laurea in Informatica

Esercizio 1: Progettare e implementare una classe `AnagraficaUniversitaria` che si occupi di gestire gli studenti e i professori di un'università (Si possono riutilizzare le classi viste a lezione). La classe deve avere i seguenti metodi:

- `void aggiungiStudiante(string codiceFiscale, string nome, string cognome, float isee);` aggiunge lo studente all'anagrafica se non è già presente il codice fiscale.
- `void aggiungiProfessore(string codiceFiscale, string nome, string cognome, float stipendio);` aggiunge il professore all'anagrafica se non è già presente il codice fiscale.
- `Persona* getPersona(string codiceFiscale) const;` restituisce la persona con il codice fiscale passato come parametro. Nel caso non ci siano persone con quel codice fiscale, restituisce `nullptr`.
- `void stampa() const;` stampa tutte le persone presenti nell'anagrafica.

Progettare e implementare una classe `Mensa` che permetta di gestire i dati riguardanti le operazioni giornaliere di una mensa universitaria. La classe deve avere i seguenti metodi:

- `void accesso(Persona* p);` aggiunge la persona `p` alle persone che hanno usufruito della mensa in quel giorno. Una persona non può accedere due volte alla mensa lo stesso giorno.
- `void stampa() const;` stampa tutte le persone che hanno usufruito della mensa in quel giorno.
- `float calcolaIncassoGiornaliero() const;` calcola l'incasso giornaliero, tenendo in conto le seguenti tariffe:
 - La tariffa per gli studenti si calcola in base al valore dell'ISEE:
 - * $ISEE \leq 10000$ euro, tariffa: 0.0 euro
 - * $ISEE > 10000$ e ≤ 20000 euro, tariffa: 2.0 euro
 - * $ISEE > 20000$ e ≤ 40000 euro, tariffa: 3.0 euro
 - * $ISEE > 40000$ euro, tariffa: 4.0 euro
 - La tariffa per i docenti si calcola in base allo stipendio:

- * Stipendio ≤ 30000 euro, tariffa: 2.5 euro
- * Stipendio > 30000 e ≤ 40000 euro, tariffa 3.5 euro
- * Stipendio > 40000 , tariffa: 5.0 euro
- La tariffa per le altre persone è di 7 euro.
- void nuovoGiorno(); resetta i dati della mensa per far partire un nuovo giorno.

NOTA: La classe Mensa deve essere abbastanza flessibile da non essere modificata nel caso in cui nuove figure universitarie abbiano accesso alla mensa ad un prezzo scontato.

Esercizio 2: Utilizzando le classi Vincitore.h, Atletica.h e Atletica.cpp allegate alla traccia, implementare i 4 metodi in Atletica.cpp.