

Esercitazione - Classi (300)

Fondamenti 2 - Corso di Laurea in Informatica

Esercizio 1: Realizzare una classe `Programmatore` che abbia i seguenti campi privati:

- **string nome**, che rappresenta il nome del programmatore;
- **unsigned eta**, che rappresenta l'età del programmatore;
- **unsigned stipendio**, che indica lo stipendio mensile percepito dal programmatore;
- **unsigned anni_esperienza**, che specifica gli anni di esperienza maturati dal programmatore;
- **bool senior**, true se il programmatore è di tipo senior, false se è di tipo junior.

Implementare, oltre ai metodi `get` e `set`, gli operatori di input e output. Realizzare, in seguito, un `main` che, utilizzando la classe definita precedentemente, legga in input 5 programmatori ed implementi i seguenti metodi:

- **unsigned getNumeroProgrammatoriPerNome(Programmatore programmatori[], string nome)** che, ricevuto in input l'array di programmatori e uno specifico nome, restituisca il numero di programmatori presenti nell'array aventi quel nome.
- **float getStipendioMedioSenior(Programmatore programmatori[])** che, ricevuto in input l'array di programmatori, restituisca lo stipendio medio percepito dai programmatori di tipo senior. Se non sono presenti programmatori di tipo senior, restituire 0.
- **const Programmatore& getJuniorPiuEsperto(Programmatore programmatori[])** che, ricevuto in input l'array di programmatori, restituisca il programmatore di tipo junior avente più esperienza. Se non esiste alcun programmatore di tipo junior, segnalare l'errore terminando l'esecuzione del programma.
- **unsigned maxDifferenzaEta(Programmatore programmatori[])** che, ricevuto in input l'array di programmatori, restituisca la massima differenza di età tra i componenti del gruppo.

- **bool anzianita(Programmatore programmatori[])** che, ricevuto in input l'array di programmatori, restituisca true se e solo se il criterio di anzianità risulta rispettato. In particolare, ogni programmatore deve percepire uno stipendio maggiore di tutti gli altri aventi meno anni di esperienza.

Esercizio 2: Realizzare una classe `Studente` che abbia i seguenti campi privati:

- **string matricola**, che rappresenta la matricola dello studente;
- **unsigned eta**, che rappresenta l'età dello studente;
- **unsigned esami_sostenuti**, che indica il numero di esami sostenuti dallo studente;
- **string nomi_esami[24]**, che in posizione i riporta il nome dell' i -esimo esame sostenuto;
- **unsigned voti_esami[24]**, che in posizione i riporta il voto ottenuto all' i -esimo esame sostenuto;
- **unsigned cfu_esami[24]**, che in posizione i riporta il numero di CFU dell' i -esimo esame sostenuto;

Dare la possibilità di costruire un'istanza di `Studente` specificando matricola ed età.

Implementare, oltre ai metodi `get` e `set`, gli operatori di input e output e i seguenti metodi:

- **void aggiungiEsame(string esame, unsigned voto, unsigned cfu);**
- **void rimuoviUltimoEsame();**
- **float getMediaVoti() const;**
- **float getMediaPonderata() const** (la media ponderata si calcola come somma pesata dei voti rispetto ai CFU dividendo, infine, per il numero totale di CFU acquisiti);
- **unsigned getVotoI(unsigned i) const** che restituisce il voto i -esimo.

Realizzare, in seguito, un `main` che, utilizzando la classe definita precedentemente, legga in input un intero k e, successivamente, k studenti, implementando i seguenti metodi:

- **bool bestStudents(Studente studenti[], unsigned numeroStudenti)** che, ricevuto in input un array di studenti con la relativa dimensione, restituisca true se e solo se la media ponderata di tutti gli studenti risulta non inferiore a 28.

- **unsigned numeroEsamiAllaLaurea(Studente studenti[], unsigned numeroStudenti)** che, ricevuto in input un array di studenti con la relativa dimensione, restituisca il numero di esami mancanti allo studente che risulta essere più prossimo alla laurea. Si supponga che il numero di esami totali sia pari a 24.
- **bool stessiVoti(const Studente& s1, const Studente& s2)** che restituisca true se e solo se gli studenti s1 ed s2 hanno ottenuto lo stesso numero di volte ogni votazione.
Ad esempio, il metodo deve restituire true se s1 ha sostenuto l'esame di Inglese con voto pari a 27 e quello di Analisi con voto pari a 25, mentre s2 ha sostenuto l'esame di Matematica Discreta con voto pari a 25 e quello di Inglese con voto pari a 27.
- **string getMatricolaStudenteGiovaneEBravo(Studente studenti[], unsigned numeroStudenti)** che, ricevuto in input un array di studenti con la relativa dimensione, tra tutti gli studenti aventi al più 20 anni e che hanno sostenuto almeno 3 esami, restituisce la matricola di quello con la media dei voti maggiore.
A parità di media voti, restituire lo studente più giovane. Se non esiste alcuno studente con tali caratteristiche, restituire la stringa "- -".