

Progetto – "Gestione biblioteca personale"

Specifiche funzionali

Realizzare un programma C++ che offra funzioni di base per la gestione di una semplice biblioteca personale. Precisamente, il programma deve permettere la registrazione e manipolazione di *informazioni* relative a (propri):

- **libri**
- **film** (su DVD)
- **album musicali** (su CD)
- **amici**.

Ciascun dato dei primi tre tipi deve contenere: nome e cognome autore/i (max. 5),¹ titolo, casa editrice/produttrice, anno pubblicazione. Inoltre, ciascun libro comprende anche (almeno) il numero di pagine; ciascun film comprende anche (almeno) un elenco di attori (max 5) e una durata in minuti; ciascun album musicale comprende anche (almeno) un elenco di singoli brani musicali (max. 30) ciascuno con la propria durata, e una durata totale.

Ciascun dato relativo agli amici dovrà contenere: nome, cognome, indirizzo, numeri di telefono (max. 3), email, account facebook, note (testo libero, max. 100 caratteri).

Per ogni dato dei tipi sopra elencati il programma deve permettere l'esecuzione delle seguenti operazioni:

1. aggiunta di un nuovo dato di tipo libro/film/album/amico (i campi sottolineati--vedi sopra--sono obbligatori; gli altri sono facoltativi);
2. ricerca e visualizzazione (su std output) di tutte le informazioni relative a un libro/film/album individuato tramite sotto-stringa del cognome di uno degli autori o del titolo;²
3. ricerca e visualizzazione (su std output) di tutte le informazioni relative a un amico individuato tramite sottostringa del nome+cognome;
4. modifica di alcuni campi (indirizzo, note, ...) di un dato di tipo amico individuato come al punto (3);
5. cancellazione di un dato di tipo libro/film/album/amico, individuato come ai punti (2) e (3);
6. aggiunta di un **prestito** di un libro/film/album ad un amico; l'oggetto prestato e l'amico sono individuati come ai punti (2) e (3); ad ogni prestito è associata una data;
7. cancellazione di un prestito, individuato come al punto (2);
8. visualizzazione (su std output) di tutti i prestiti attivi;
9. salvataggio su file di tutti i dati e i prestiti.

L'interazione con l'utente avviene tramite **menù a più livelli**. Il programma provvede anche ad effettuare controlli sulla **correttezza dei dati** immessi (ad esempio, codice dell'operazione scelta, validità delle date, ecc.).

Note di progettazione

- Le entità *libro/film/album musicale* devono essere definite come altrettante **classi**, **derivate** da una classe comune, denominata ad esempio *pubblicazione*, che contiene informazioni su: autori, titolo, casa editrice/produttrice, anno pubblicazione. La nozione di *amico* deve essere definita come una classe, **derivata** dalla classe *persona*, contenente informazioni su: nome e cognome, indirizzo, numeri di telefono.
Ciascuna di queste classi deve fornire (almeno) una o più funzioni di **ricerca** di sottostringhe su uno o più dei suoi campi dati, come previsto nelle specifiche (ad es., la funzione `ricerca_nel_titolo(s)` per la ricerca della sottostringa `s` nel titolo di una pubblicazione).
- Gli insiemi di dati di tipo libro/film/album/amico devono essere definiti come altrettante **classi** (ad es. di nome `insieme_libri`, `insieme_film`, ...).

¹ Nel caso di film si intende il regista, nel caso di album musicale il/i cantante/i

² Nel caso in cui più dati soddisfino i criteri di ricerca, i dati trovati vengono visualizzati tutti, in successione.

Ciascuna di queste classi deve fornire (almeno) una o più funzioni di **ricerca** di sottostringhe sull'insieme dei dati gestiti e una o più funzioni per il **salvataggio** e la lettura dell'insieme di dati su/da file.

- Tutti i dati relativi a libri, film, album, amici e prestiti vengono salvati su altrettanti **file** su richiesta esplicita dell'utente e/o automaticamente al termine del programma, e caricati da file in memoria principale all'inizio dell'esecuzione del programma.

Note di implementazione

- Si richiede di realizzare la gestione di eventuali situazioni anomale tramite i meccanismi di **gestione eccezioni** del C++ ove opportuno.
- Le diverse operazioni (aggiunte, ricerche, modifiche, ...) devono essere svolte su dati memorizzati in **memoria principale** (ad esempio, array) e non direttamente su file.
- E' vietato: usare dichiarazioni `friend`; usare `goto`; usare variabili globali; definire classi con solo parte `public`; usare ereditarietà a sproposito; usare librerie grafiche.
- I dati devono essere memorizzati su file secondo un opportuno **formato**, come ricavabile dal seguente esempio relativo all'insieme dei dati di tipo *libro*:

Luis Joyanes	(nome)
Aguilar	(cognome)
#	(indica la terminazione della lista degli autori)
Fondamenti di programmazione in C++	(titolo)
McGraw-Hill	(casa editrice)
2008	(anno pubblicazione)
663	(numero di pagine)
Harvey M.	(nome)
Deitel	(cognome)
Paul J.	(nome)
Deitel	(cognome)
#	(indica la terminazione della lista degli autori)
C++ Fondamenti di programmazione	(titolo)
*	(indica informazione mancante – di tipo stringa)
2005	(anno pubblicazione)
0	(indica informazione mancante – di tipo numero)