

FONDAMENTI DI INFORMATICA I

FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Una `CollezioneCD` è un contenitore fisico di CD dove può essere custodito un numero illimitato di CD musicali. Ciascun CD può essere vergine (-), oppure masterizzato. Nel secondo caso, il CD può contenere musica rock (R) oppure musica classica (C). La collezione mantiene i CD ordinati in base all'ordine di inserimento.

Le operazioni che si possono effettuare sulla collezione di CD sono le seguenti:

(PRIMA PARTE)

- ✓ **`CollezioneCD c;`**
Costruttore di default, che inizializza una `CollezioneCD c`. Inizialmente la collezione non contiene CD.
- ✓ **`c.aggiungi()`**
Operazione che aggiunge un CD vergine alla collezione.
- ✓ **`c.aggiungi(g)`**
Operazione che aggiunge un CD di genere `g` alla collezione, dove `g` è un carattere, e i caratteri validi sono solo 'C' e 'R'. Implementare una corretta gestione delle situazioni di errore.
- ✓ **`c.masterizza(g)`**
Operazione che masterizza il primo CD vergine trovato nella collezione (ossia il meno recente) con una musica di genere `g`. La funzione restituisce `vero` se esiste un CD vergine da masterizzare, `falso` altrimenti.
- ✓ **`cout << c`**
Operatore di uscita per il tipo `CollezioneCD`. L'uscita ha la forma seguente:
[R-CR]

I CD vengono visualizzati in ordine di inserimento. In questo esempio la collezione `c` è formata da quattro CD. Il primo ed il quarto sono di genere rock, il secondo è vergine e il terzo è di genere classico (l'operatore ***non*** deve aggiungere un *new line* dopo la parentesi quadra chiusa).

(SECONDA PARTE)

- ✓ **`c.elimina()`**
Operazione che elimina dalla collezione `c` il CD vergine inserito meno di recente.
- ✓ **`CollezioneCD c1(c)`**
Costruttore di copia che inizializza un oggetto `c1` con il valore dell'oggetto `c`.
- ✓ **`~c`**
Operatore di complemento bit a bit, che ritorna il numero di CD vergini presenti in `c`.
- ✓ **`c%g`**
Operatore di modulo, che ritorna il numero di CD di genere `g` presenti nella collezione `c`.
- ✓ **`~CollezioneCD()`**
Distruzione della classe `CollezioneCD`.

Mediante il Linguaggio C++, realizzare il tipo di dati astratti `CollezioneCD`, definito dalle precedenti specifiche, **utilizzando una implementazione a lista**. Individuare le eventuali situazioni di errore, e metterne in opera un corretto trattamento.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA

AVVIO E IDENTIFICAZIONE

- Avviare la macchina in modalità diskless, scegliere “Fondamenti di Informatica I” ed effettuare il login:
 nome: studenti
 password: studenti
- Aprire un terminale e spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.
- Dare il comando `$ ident`, sempre da dentro la cartella. Lo script richiede i propri dati (cognome, nome, numero di matricola e password (la password **non va dimenticata** in quanto è indispensabile per scaricare da internet il proprio elaborato a consegna avvenuta). Il comando `ident` crea il file `matricola.txt` nella cartella corrente. Lo script può essere lanciato più volte, in tal caso il file `matricola.txt` viene sovrascritto. Per verificare che il file sia stato creato e che il contenuto sia quello giusto dare il comando (la password è codificata):
`$ cat /home/studenti/elaborato/matricola.txt`
- A questo punto il docente verifica che tutti gli studenti abbiamo effettuato l’identificazione, dopodiché provvede a inviare i seguenti file nella cartella `elaborato` del proprio PC: `compito.h`, `compito.cpp`, `main.cpp`. Controllare pertanto che questi file, insieme al file `matricola.txt`, siano presenti sul proprio elaboratore.

SVOLGIMENTO DELLA PROVA

- Definire ed implementare il tipo di dato astratto richiesto e le relative funzioni nei file `compito.h` e `compito.cpp`. Il file `main.cpp` contiene la funzione principale `main()` ed è utilizzato dallo studente per testare la sua implementazione della classe. Il file `main.cpp` può essere modificato a piacere. In sede di valutazione dell’elaborato verrà considerato **esclusivamente il contenuto dei file `compito.h` e `compito.cpp`** ed è pertanto **vietato cambiare nome a tali file**.

Per compilare e linkare dare il comando:

```
$ g++ main.cpp compito.cpp (eseguibile invocabile tramite $ ./a.out)  
(utilizzare g++ -g per includere le informazioni di debug qualora si intenda debuggare con ddd) .
```

PER CONSEGNARE O RITIRARSI

Recarsi dal docente avendo preso nota dell’identificativo della macchina (g34, s23, ...).

REGOLA PER LA CORREZIONE

Per il superamento della prova pratica è **obbligatorio** aver definito ed implementato correttamente la classe e le seguenti funzioni:

- costruttore (`CollezioneCD c;`)
- aggiungi CD vergine (`c.aggiungi();`)
- aggiunge in CD di genere `g` (`c.aggiungi(g);`)
- masterizza un CD con musica di genere `g` (`c.masterizza(g)`)
- stampa a video (`cout<<c`)

Il codice di altre funzioni che impedisca la compilazione deve essere messo come commento. Tali funzioni verranno comunque valutate in sede di correzione della prova.