

FONDAMENTI DI INFORMATICA I

FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Un `Autonoleggio` gestisce un parco auto dove i clienti possono recarsi per noleggiare auto e restituirle successivamente. Il gestore dell'`Autonoleggio` deve poter inserire nuove auto nel parco macchine, oltre che permettere ad un cliente di noleggiare e successivamente restituire auto.

Implementare le seguenti operazioni che possono essere effettuate su un `Autonoleggio`, assumendo che la targa (identificatore univoco per la macchina) sia una stringa di 7 caratteri; che i clienti siano identificati dal `NOME.COGNOME` (una stringa di 20 caratteri al massimo) e che il numero di posti massimo per un'auto sia 9.

- ✓ **`Autonoleggio a`**
Costruttore che inizializza un `Autonoleggio a`. Inizialmente l'`autonoleggio` non contiene auto.
- ✓ **`a.insauto(s, n)`**
Funzione che aggiunge un'auto all'`autonoleggio`. L'auto ha targa `s` e numero di posti `n`. Se l'operazione fallisce, la funzione lascia l'`autonoleggio` inalterato e restituisce `false`. Restituisce `true` nei casi di successo.
- ✓ **`a.noleggia(c, n)`**
Funzione che realizza il noleggio da parte di un cliente identificato da `c` di un'auto di esattamente `n` posti. Restituisce `false` nel caso non ve ne sia alcuna disponibile con tale numero di posti, `true` negli altri casi. Uno stesso cliente può noleggiare successivamente altre auto.
- ✓ **`cout << a`**
Operatore di uscita per il tipo `Autonoleggio`. Per ogni auto dell'`autonoleggio` viene stampata la targa, il numero di posti e se e' presente o meno nell'`autonoleggio` e, qualora sia assente, il nome ed il cognome di chi l'ha noleggiata. *Le auto debbono essere mostrate ordinate secondo la targa.*

```
ET341CV, 7, ASSENTE (LUIGI.BIANCHI)
GG112RS, 9, ASSENTE (GIOVANNI.VERDI)
RT844ST, 3, PRESENTE
RT986ZH, 3, PRESENTE
SR444AB, 5, ASSENTE (MARIO.ROSSI)
```

In questo esempio l'`autonoleggio` ha cinque auto, di cui due da 3 posti, una da 5, una da 7 ed una da 9. Entrambe le auto da 3 posti sono presenti nell'`autonoleggio`, mentre le auto da 5, 7 e 9 posti sono state noleggiate, rispettivamente, da `MARIO.ROSSI`, `LUIGI.BIANCHI` e `GIOVANNI.VERDI`.

- ✓ **`a.riconsegna(c)`**
Funzione che riconsegna tutte le auto noleggiate dal cliente `c`.
- ✓ **`a2(a)`**
Costruttore di copia.
- ✓ **`!a`**
Operatore di negazione logica che restituisce un `autonoleggio` con tutte e solo le auto di `a` noleggiate.
- ✓ **`~Autonoleggio()`**
Distruttore.

Utilizzando il linguaggio C++, realizzare il tipo di dati astratti `Autonoleggio` definito dalle precedenti specifiche. Individuare 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, dopodichè 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 (`Autonoleggio a`)
- inserisci auto (`a.insauto(s,n)`)
- noleggia auto (`a.noleggia(c,n)`)
- stampa a video (`cout<<a`)

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.