PROVA PRATICA 17 SETTEMBRE 2013

FONDAMENTI DI INFORMATICA I
FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Un `Vagone` è in grado di contenere al più `N` casse di vino. Le casse possono essere di vino bianco oppure rosso. Un `Treno` può contenere un numero illimitato di vagoni. Le operazioni che possono essere effettuate su di un `Treno` sono le seguenti:

PRIMA PARTE ***(qualora siano presenti errori di compilazione, collegamento o esecuzione in questa prima parte, l'intera prova verrà considerata insufficiente e pertanto non verrà corretta)***

- ✓ **`Treno t;`**
  Costruttore di default, che inizializza un treno `t`. Inizialmente in treno è privo di vagoni.
- ✓ **`t.aggiungiVagone();`**
  Operazione che aggiunge un vagone vuoto in testa al treno `t`.
- ✓ **`t.aggiungiCassa(c);`**
  Operazione che aggiunge una cassa di vino di colore `c` al vagone che contiene il minor numero di casse, dove `c` è una costante enumerata di tipo `colore`:

  ```
  enum colore {ROSSO, BIANCO};
  ```

  Nel caso in cui il numero di casse minimo sia presente in più vagoni, la nuova cassa deve essere inserita nel vagone inserito meno di recente.
- ✓ **`cout << t`**
  Operatore di uscita per il tipo `Treno`. L'uscita ha la forma seguente:

  ```
  <r0,b1>-<r1,b0>-<r1,b1>
  ```

  Ogni coppia di parentesi angolari aperte/chiuse rappresenta un vagone. In questo esempio, il `Treno` è composto da tre vagoni, di cui il vagone di testa (inserito per ultimo) ha una cassa di vino bianco, quello centrale una di vino rosso e quello di coda (inserito per primo) ha una cassa di rosso ed una di bianco. L'operatore non deve aggiungere una *new line* alla fine.

SECONDA PARTE ***(si invita a commentare le operazioni di questa seconda parte che dovessero impedire la compilazione, il collegamento o la corretta esecuzione del codice)***

- ✓ **`t.eliminaCassa();`**
  Operazione che toglie una cassa di vino dal vagone con maggior numero di casse di vino. Se in tale vagone il numero dei casse di vino di colore rosso è maggiore del numero dei casse di vino bianco, viene eliminata una cassa di vino rosso, altrimenti (*minore o uguale*) una di vino bianco. Nel caso in cui il maggior numero di casse sia lo stesso per più di un vagone, ne viene tolta una da quello inserito più di recente.
- ✓ **`t.togliVagone();`**
  Operazione che toglie dal treno il vagone con minor numero di casse di vino. Nel caso in cui più di un vagone abbia lo stesso numero minimo di casse, toglie quello inserito più di recente.
- ✓ **`t1(t);`**

  Costruttore di copia, che inizializza un treno `t1` col valore del treno `t`.
- ✓ **`~Treno();`**
  Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto `Treno`, definito dalle precedenti specifiche. 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, …).