

Esercitazione n. 3

Sviluppare una classe che gestisca uno *stack*: una struttura dati lineare di tipo LIFO a cui si può accedere soltanto mediante uno dei suoi capi sia per memorizzare sia per estrarre dati.

L'implementazione deve utilizzare una struttura concatenata composta da una collezione di *nodi* che contengono dati e riferimenti ad altri nodi.

Il *nodo* deve essere una classe con due attributi privati: il valore memorizzato e un puntatore al nodo successivo. Il valore memorizzato è di tipo `char`.

Lo *stack* deve essere una classe con un solo attributo privato, il puntatore al nodo in testa allo stack. L'interfaccia pubblica della classe *stack* deve fornire, oltre al costruttore e al distruttore, i seguenti metodi:

`Push`: inserisce un valore nella pila.

`Pop`: estrae il valore di testa della pila.

`Top`: ritorna il valore di testa della pila senza estrarlo.

`Clear`: svuota la pila.

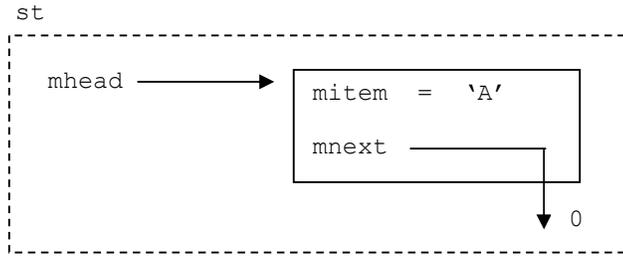
`IsEmpty`: segnala se la pila è vuota.

Verificare la correttezza dell'implementazione: si esegua una verifica per ogni metodo pubblico e una verifica di funzionamento completo. Per esempio: (i) inserire, estrarre; (ii) inserire, svuotare, inserire, estrarre; (sequenze di caratteri).

Nota:

In fase di sviluppo dell'applicazione è utile poter verificare il numero di oggetti *nodo* correntemente allocati (per controllare la presenza di eventuali memory leak). Si suggerisce di utilizzare un attributo privato statico nella classe *nodo*.

```
CMyStack st;  
st.Push('A');
```



```
st.Push('B');
```

