

Relazione Progetto C++ Stack

Introduzione progetto

Il progetto richiede la progettazione e realizzazione di una classe generica che implementa uno stack di elementi di tipo T. Uno stack è una struttura dati dove gli elementi sono organizzati in una pila. Un elemento che viene inserito nello stack con il metodo push è posto "sopra" gli altri già nello stack. E' possibile prelevare dallo stack, mediante il metodo pop, l'elemento "in cima" alla pila. Una volta che un elemento è prelevato, deve essere rimosso dallo stack.

Struttura dello stack :

_arr Puntatore allo stack

_size Numero di elementi dello stack

Definizioni :

- il costruttore default nel quale la _size viene impostata a 0 (numero di elementi dello stack) e generato il puntatore.
- Il costruttore di copia nel quale viene copiato uno stack in un altro
- La getSize per sapere la dimensione dello stack
- La clear custom per eliminare lo stack
- La push per inserire valori in cima allo stack
- La pop per rimuovere valori dalla cima dello stack. Se si esegue la pop sullo stack vuoto verrà comunicato un errore e verrà interrotta l'esecuzione del programma
- Il metodo removeif che, dato un predicato generico P su un elemento di tipo T elimina dallo stack tutti gli elementi che soddisfano il predicato P.

Test :

Nel file main.cpp sono riportati metodi per il test di tutte le principali operazioni della classe MyStack.

Note generali :

Visto che lo stack è una struttura dati LIFO, Last In First Out, è stato scelto di invertire gli iteratori.

Eseguendo la pop su uno stack vuoto allora verrà eseguita clear e poi bloccato il programma tramite assert

Valgrind:

```
==218==  
==218== HEAP SUMMARY:  
==218==    in use at exit: 0 bytes in 0 blocks  
==218==   total heap usage: 73 allocs, 73 frees, 74,960 bytes allocated  
==218==  
==218== All heap blocks were freed -- no leaks are possible  
==218==  
==218== For lists of detected and suppressed errors, rerun with: -s  
==218== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```