Giacomini Stefano, matricola 829854

mail: s.giacomini3@campus.unimib.it

progetto-cpp

Stack con elemnti di tipo generico

La Struttura Dati

Ragionando sulla consegna del progetto ho deciso che il modo migliore per rappresentare uno stack con id generici (template) senza l'utilizzo di vector o liste fosse quello di creare una classe avente una array dinamico e una size indicante la dimensione dello stack.

In questa implementazione di uno stack la struct Stack contiene

- value_type di tipo T template che "salva" il tipo di dati dell'array dinamico
- Un unsigned int size_type contente la dimensione del tipo T
- Un array dinamico di tipo value_type
- Una variabile di tipo size_type contente la dimensione dell'array

Per la struct Stack vengono implementati:

- Un costruttore di default che mette il numero di nodi a stack a nullptr e size a 0
- Un costruttore secondario che prende in input una dimensione e un valore, vado a creare un array dinamico stack di dimensione data dove ogni elemnto è inizializzato al valore passato in input
- Un costruttore di conversione che prende in input un'altro stack avente lo tipo diverso
- Un costruttore di copia che prende in input un'altro stack avente lo stesso tipo
- Un operatore di assegnamento
- Un Distruttore che rimuove dall'heap l'array dinamico e setta stack a nullptr e size a 0
- Un metodo getter di size chiamato size()

- Un metodo getter dello stack chiamato value che prende in input un indice e ritorna il valore dello stack a tale indice
- Un metodo swap che prende in input un'altro satck e ne scambia stack e size
- Un costruttore secondario che prende in input due iteratori, uno di inizio e uno di fine e va a creare uno stack contente i valori compresi tra questo due iteratori
- Metodo begin che ritorna l'iteratore in fondo allo stack
- Metodo end che ritorno l'iteratore in cima allo stack

Funzioni Principali

Le funzioni princiapali sono:

- **push(const value_type &value)**: Il metodo che aggiunge in cima allostack l'elemento passato in input
- **pop()**: Metodo che ritorna l'elemento in cima allo stack e lo rimuove da esso
- **clean()**: Metodo che svuota lo stack e setta size a 0
- **fill(iterator begin, iterator end)**: Metodo che svuota lo stack (nel caso non fosse già vuoto) e lo riempie con i valori compresi tra begin e end (compresi)
- **removeif(Pred pr)**: Metodo che prende in input un predicato generico e che rimuove dalla stack tutti quegli elemnti che soddisfano tale predicato

In più è implementato l'operatore di stream di Stack che stampa l'array dinamico e la size.

Iteratore

All'interno della classe Graph viene implementata una sottoclasse rappresentante un iteratore di tipo forward costante che itera sugle elemnti dell'array dinamico stack ritornandone il valore.

Per l'iteratore vengono implementati seguenti metodi: * Costruttori di default e costruttore di copia

- Distruttore

- Operatore di assegnamento
- Operatore di dereferenziamento
- Operatore di puntamento
- Operatore pre-incremento e di post-incremento
- Operatore di uguaglianza e di diversità
- Costruttore privato di inizializzazione usato dalla classe container Graph

Test

I test vengono fatti utilizzando una struct punto che viene quindi definita in graph.h e implementata in main.cpp. Vengono in più implementati un predicato isEven e una struct even per testare removeif.