Міністерство освіти і науки України

Львівський національний університет імені І. Франка

Факультет прикладної математики та інформатики

Кафедра дискретного аналізу та інтелектуальних систем

Звіт

Про виконання лабораторної роботи №2

“Стек, стек STL”

Виконав:

Студент групи ПМІ-13

Тимчишин Ярема

Львів 2022

**Мета:** навчитися писати програми для роботи зі стеком на базі масиву чи однозв’язного списку та з використанням контейнера <stack> з STL.

**Теоретичні відомості**

**Стек** в інформатиці та програмуванні — різновид лінійного списку, структура даних, яка працює за принципом «останнім прийшов — першим пішов» (LIFO, англ. *last in, first out*). Всі операції (наприклад, видалення елемента) в стеку можна проводити тільки з одним елементом, який міститься на верхівці стека та був уведений в стек останнім.

Стек можна уявити як стопку тарілок, з якої можна взяти верхню, і на яку можна покласти верхню тарілку.

Стек можна організувати як масив або множину комірок у певній ділянці пам'яті комп'ютера з додатковим зберіганням ще й вказівника на верхівку стека. Заштовхування елемента в стек збільшує адресу вказівника, виштовхування елемента зменшує її. Таким чином, адреса вказівника завжди відповідає комірці масиву, в якій зараз міститься верхівка стека.

**Операції зі стеком:**

* **push** («заштовхнути елемент»): елемент додається в стек та розміщується в його верхівці. Розмір стека збільшується на одиницю. При перевищенні граничної величини розміру стека, відбувається переповнення стека (англ. *stack overflow*).
* **pop** («виштовхнути елемент»): повертає елемент з верхівки стека. При цьому він видаляється зі стека і його місце у верхівці стека займає наступний за ним відповідно до правила LIFO, а розмір стека зменшується на одиницю. При намаганні «виштовхнути» елемент зі вже пустого стека, відбувається ситуація «незаповненість» стека (англ. *stack underflow*).

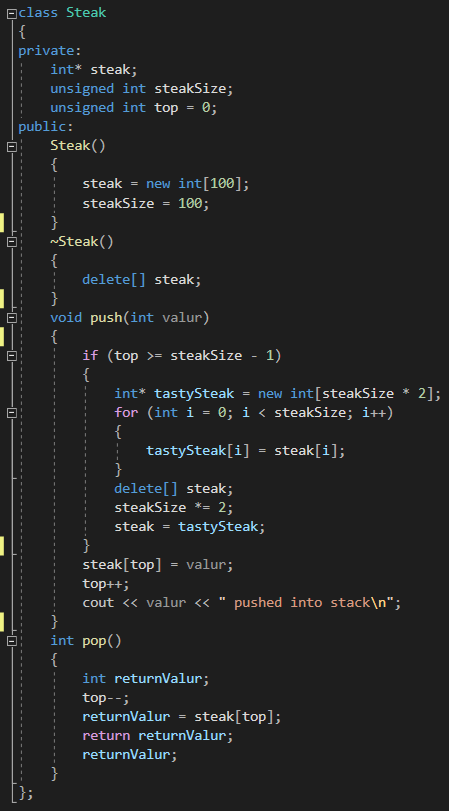
**Додаткові операції (присутні не у всіх реалізаціях стека):**

* **isEmpty**: перевірка наявності елементів у стеку; результат: істина (true), коли стек порожній.
* **isFull**: перевірка заповненості стека. Результат: істина, коли додавання нового елементу неможливе.
* **clear**: звільнити стек (видалити всі елементи).
* **top**: отримати верхній елемент (без виштовхування).
* **size**: отримати розмір (кількість елементів) стека.
* **swap**: поміняти два верхніх елементи місцями.

**Хід роботи:**

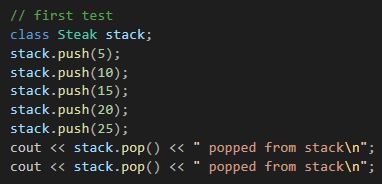
**Стек**

1. Створюю клас стек разом з основними операціями push та pop а також деструктором. Хочу зазначити, що не звіжаючи на те що у програмі я написав стейк, ми працюємо з стеком.

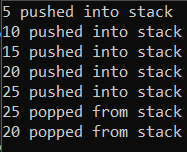


2. Написав три різні тести для перевірки роботи стеку.

У першому тесті я передав у функцію push цифри 5, 10, 15, 20 і 25 відповідно, і потім викликав функцію pop два рази:

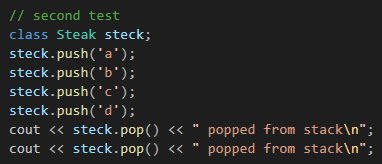


І після виконання програми отримав наступний результат:

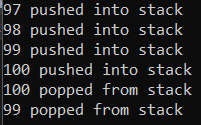


Як бачимо, з першим тестом програма справилась успішно.

У другому тесті я передав у функцію push не цифри, а символи a, b, c і d відповідно, і потім викликав функцію pop два рази:

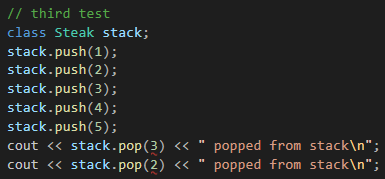


І після виконання програми отримав наступний результат:

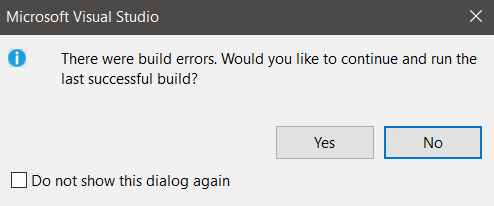


Як видно на консолі, замість того щоб запихати і забирати з стеку задані користувачем параметри, в цьому випадку це символи a, b, та d, програма запихала і витягувала зі стеку цифри 97, 98, 99 та 100. Це тому що функції у цій програмі приймають значення int, а ми запихали туди значення char. Тож програма, за таблицею ascii, перевела символи a, b, c та d у цифри, які відповідають даним символам.

У третьому тесті я передав у функцію push цифри 1, 2, 3, 4 і 5 відповідно, і потім викликав функцію pop два рази, проте цього разу я передав параметри також і у функцію pop, 3 і 2 відповідно:



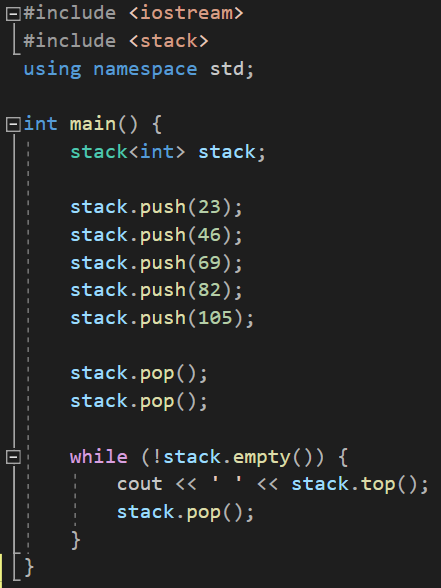
Передавши параметри у функцію pop, я мав на меті витягнути з стеку відповідно 3 та 2 елементи. У роботі з стеком заборонено витягувати якийсь елемент з середини стеку, можна витягувати лише той, який ми поклали у стек останнім. При написанні функції pop я не передав в функцію нічого так як це і має бути, а у третьому тесті ми попробували все ж таки передати у функцію якісь параметри. Програма таке не дозволяє, тому програма не запустилась.



**Стек STL**

1. Програмісти, як усім певно відомо, є дуже лінивими людьми. Тож щоб спростити роботу зі стеком, у бібліотеці STL є уже створений клас стек з готовими методами, конструкторами та операторами. Для того щоб скористатись цим стеком досить лиш написати #include <stack>.

Ось написана сама програма:



І ось результат її виконання:



Для перевірки справності цієї програми можна виконати ті самі тести, що й для простого стеку, які я написав вище і переписувати знову не мав бажання.

**Висновок:** В процесі виконання роботи я навчився складати програми для роботи зі простим стеком та стеком з бібліотеки STL, а також перевірив ці програми на працездатність.