ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

за курсом «Програмування»

студент(а) групи ПА-22-1

Бердик Роман Вікторович

кафедра комп’ютерних технологій, ДНУ

2023/2024 н.р.

1. Постановка задачі

Метою лабораторної роботи було розробити об’єктно-орієнтовану бібліотеку для роботи з виразами, представленими в постфіксній формі. Для реалізації обчислень потрібно використовувати стек для збереження чисел та виконання операцій над ними. Завдання включає:

Реалізацію стеку з використанням класів C++.

Перевантаження операторів та підтримка збереження і відновлення даних зі стеку.

Обробка виразу в постфіксній формі та виконання обчислень.

2. Опис розв’язку

Був розроблений клас Stack, що дозволяє зберігати числа та виконувати операції над ними. Основна логіка полягає у тому, що числа додаються до стеку, а при зустрічі оператора знімаються два елементи зі стеку, виконується операція, і результат повертається назад у стек.

Клас Stack підтримує перевантаження операторів, таких як [] для доступу до елементів стеку, оператор = для присвоєння та оператор -> для "розумного доступу".

Також реалізовано можливість збереження стеку у файл та відновлення його з файлу.

3. Вихідний текст програми розв’язку задачі (основні фрагменти з коментарями

Main.cpp

#include <iostream>

#include <fstream>

#include "stack.h"

int main() {

Stack<double> stack;

std::string expression;

std::cout << "Enter postfix expression: ";

std::getline(std::cin, expression);

std::istringstream iss(expression);

std::string token;

while (iss >> token) {

if (isdigit(token[0])) {

stack.push(std::stod(token));

} else {

double b = stack.pop();

double a = stack.pop();

if (token == "+") stack.push(a + b);

else if (token == "-") stack.push(a - b);

else if (token == "\*") stack.push(a \* b);

else if (token == "/") stack.push(a / b);

}

}

std::cout << "Result: " << stack.top() << std::endl;

return 0;

}

Stack.h

#ifndef STACK\_H

#define STACK\_H

#include <vector>

#include <iostream>

template <typename T>

class Stack {

private:

std::vector<T> elements;

public:

void push(const T& element) {

elements.push\_back(element);

}

T pop() {

if (elements.empty()) throw std::out\_of\_range("Stack is empty");

T element = elements.back();

elements.pop\_back();

return element;

}

T top() const {

if (elements.empty()) throw std::out\_of\_range("Stack is empty");

return elements.back();

}

bool empty() const {

return elements.empty();

}

void saveToFile(const std::string& filename) const {

std::ofstream ofs(filename);

for (const auto& el : elements) {

ofs << el << " ";

}

}

void loadFromFile(const std::string& filename) {

std::ifstream ifs(filename);

T element;

elements.clear();

while (ifs >> element) {

elements.push\_back(element);

}

}

};

#endif

4. Опис інтерфейсу програми (керівництво користувача)

Запуск програми: Користувач вводить вираз у постфіксній формі (наприклад, 2 3 10 \* +).

Обробка виразу: Програма обчислює результат за допомогою стеку і виводить його на екран.

Збереження стеку: Користувач може зберегти поточний стан стеку у файл, використовуючи метод saveToFile.

Відновлення стеку: Стек може бути відновлений з файлу за допомогою методу loadFromFile.

5. Опис тестових прикладів

Тест 1: Вираз 2 3 10 \* +.

Результат: 32.

Тест 2: Вираз 5 1 2 + 4 \* + 3 -.

Результат: 14.

6. Аналіз помилок (опис усунення зауважень)

Під час тестування програми виникла помилка, пов'язана з некоректним вилученням елементів зі стеку при обробці порожнього стеку. Ця помилка була виправлена шляхом додавання перевірки на порожність стеку перед виконанням операцій.