Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

**Звіт**

про виконання лабораторної роботи № 7

з курсу «Алгоритми та програмування, ч.2»

**Виконав**:

студент групи ФІ-11

Діяк І.В.

**Перевірила:**

Рижа І.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата) (підпис викладача)

Львів-2023 р.

**Лабораторна №7**

**Тема роботи: Стандартна бібліотека шаблонів**

**Перелік ідентифікаторів:**

**StringStack{} – клас контейнера типу стек.**

**Myifstream{} – клас файлового потоку, що читає стрічки з файлу.**

**Myoftream{] – клас файлового потоку, що пише стрічки у файл.**

**Getline( ) – метод, що читає стрічку за допомою ітератора.**

**Writeline() – метод, що пише у файл за допомогою ітератора.**

**Reverse\_stack() – метод, що перевертає стек.**

**Print\_stack() – метод, що друкує стек.**

**modifishion\_stack() – метод, що модифікує стек добавляючи йому номери.**

**Write\_stack\_to\_file**(const string& fname) – **метод, що пише стек у файл.**

**Print\_file\_list() - функція, що друкує всі файли.**

**Код програми** **(**Написано мовою C++). Програма написання— Visual Studio 2022.

#include<string>

#include<iostream>

#include <fstream>

#include<sstream>

#include"Myifstream.h"

#include"Myofstream.h"

#include<windows.h>

#include<stack>

#include<algorithm>

using namespace std;

#pragma warning (disable:4996)

enum

{

\_print\_file = 1, \_print\_stack, \_from\_file\_to\_stack, \_modifishion\_stack, \_reverse\_stack, \_write\_to\_file, END

};

class StringStack : public stack<string>

{

public:

void print\_stack() const;

void modifishion\_stack();

void delete\_stack();

void reverse\_stack();

void write\_stack\_to\_file(const string& fname) const;

};

int from\_file\_to\_stack(StringStack& stack , const string& fname);

int print\_file(const string& fname);

int menu();

int inputNumber(const char\*);

void print\_file\_list();

int menu()

{

int choice;

cout << "\n\_\_Menu\_\_" << endl;

cout << \_print\_file << '.' << "Print file " << endl;

cout << \_print\_stack << '.' << "Print stack " << endl;

cout << \_from\_file\_to\_stack << '.' << "From file to stack " << endl;

cout << \_modifishion\_stack << '.' << "Modifi stack " << endl;

cout << \_reverse\_stack << '.' << "Reverse stack " << endl;

cout << \_write\_to\_file << '.' << "Write stack to file" << endl;

cout << END << '.' << "End " << endl;

choice = inputNumber("\nEnter number: ");

return choice;

}

int inputNumber(const char\* msg)

{

cout << msg;

const int size = 5;

char chyslo[size + 1];

cin.getline(chyslo, size);

return atoi(chyslo);

}

void print\_file\_list()

{

WIN32\_FIND\_DATAA file\_info;

HANDLE file\_handle;

file\_handle=FindFirstFileA("\*.txt", &file\_info);

if(file\_handle!=INVALID\_HANDLE\_VALUE)

do

{

cout << file\_info.cFileName << endl;

} while (FindNextFileA(file\_handle,&file\_info));

}

int print\_file(const string& fname)

{

Myifstream fin(fname);

string str = "";

if (!fin.is\_open())

{

cout << "File read Error" << endl;

return -1;

}

while (fin.getline(str))

{

cout << str << endl;

str = "";

}

cout <<str<< endl;

fin.close();

return 0;

}

int from\_file\_to\_stack(StringStack& stack, const string& fname)

{

Myifstream fin(fname);

string str;

int n = 0;

if (!fin)

{

cout << "File read Error" << endl;

return -1;

}

while (getline(fin, str))

{

stack.push(str);

n++;

cout << str << endl;

}

cout << endl;

fin.close();

return n;

}

void StringStack::print\_stack() const

{

//for\_each(c.rbegin(), c.rend(), [](const string& str) {cout << str << endl; }); // лямбда-вираз

copy(c.begin(), c.end(), ostream\_iterator<string, char>(cout, "\n"));

}

void StringStack::delete\_stack()

{

while (!empty())

pop();

}

void StringStack::reverse\_stack()

{

reverse(c.begin(), c.end());

}

void StringStack::modifishion\_stack()

{

int n = 0;

transform(c.begin(), c.end(), c.begin(), [&n](const string& str) {ostringstream string\_stream; string\_stream << ++n << ".\t" << str; return string\_stream.str(); });

}

void StringStack::write\_stack\_to\_file(const string& fname) const

{

Myofstream fout(fname);

copy(c.begin(), c.end(), ostream\_iterator<string, char>(fout, "\n"));

fout.close();

}

int main()

{

string file\_name;

int n = 0;

StringStack stack;

int choice;

print\_file\_list();

cout << "enter file name from list: ";

getline(cin,file\_name);

while ((choice = menu()) != END)

{

system("cls");

switch (choice)

{

case \_print\_file:

print\_file(file\_name);

break;

case \_print\_stack:

stack.print\_stack();

break;

case \_from\_file\_to\_stack:

if (!stack.empty())

{

stack.delete\_stack();

}

n = from\_file\_to\_stack(stack, file\_name);

if (n > 0)

{

cout << "Stack created with " << n << " rows" << endl;

}

break;

case \_modifishion\_stack:

stack.modifishion\_stack();

cout << "Modified stack: " << endl;

stack.print\_stack();

break;

case \_reverse\_stack:

cout << "Your stack: " << endl;

stack.reverse\_stack();

stack.print\_stack();

break;

case \_write\_to\_file:

stack.write\_stack\_to\_file("numbered" + file\_name);

break;

default:

cout << "Error: unknown menu item" << endl;

break;

}

}

if (!stack.empty())

{

stack.delete\_stack();

}

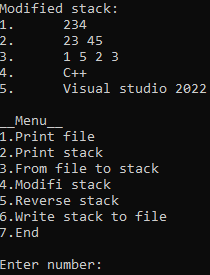
cout << "Ends of program" << endl;

system("pause");

system("cls");

}

**Контрольний приклад:**



**Висновок:** на цій лабораторній я навчився працювати з контейнерами стандартної бібліотеки шаблонів, використовувати ітератори для потокового вводу/виводу і використовувати стрічкові потоки.