**Дисциплина: Объектно Ориентированное Программирование**

**Лабораторная работа №5**

**Тема: Шаблоны классов на языке С++**

Выполнил: Бардин М.И. Студент: 2-курса

Группа: ИТ-902

Проверил: ст. пр. Юрков В.А.

Луганск 2022

**Задание:**

1. Создать шаблон класса **Class\_Stack**, который должен содержать 2 метода: **Push** ( поместить элемент в стек ) и **Pop** ( извлечь элемент из стека ). При попытке добавить элемент в заполненный стек, метод **Push** должен выдавать сообщение: «Стек заполнен». При попытке извлечь элемент из пустого стека, метод **Pop** должен выдавать сообщение: «Стек пуст». Шаблон класса должен работать со встроенными типами данных языка С++. Класс **Class\_Stack** должен содержать 2 конструктора: конструктор по умолчанию ( в этом случае размер стека устанавливается равным 10 элементам ) и конструктор принимающий в качестве параметра количество элементов в стеке. В случае передачи некорректного количества элементов стека, размер стека устанавливается равным 10 элементам.

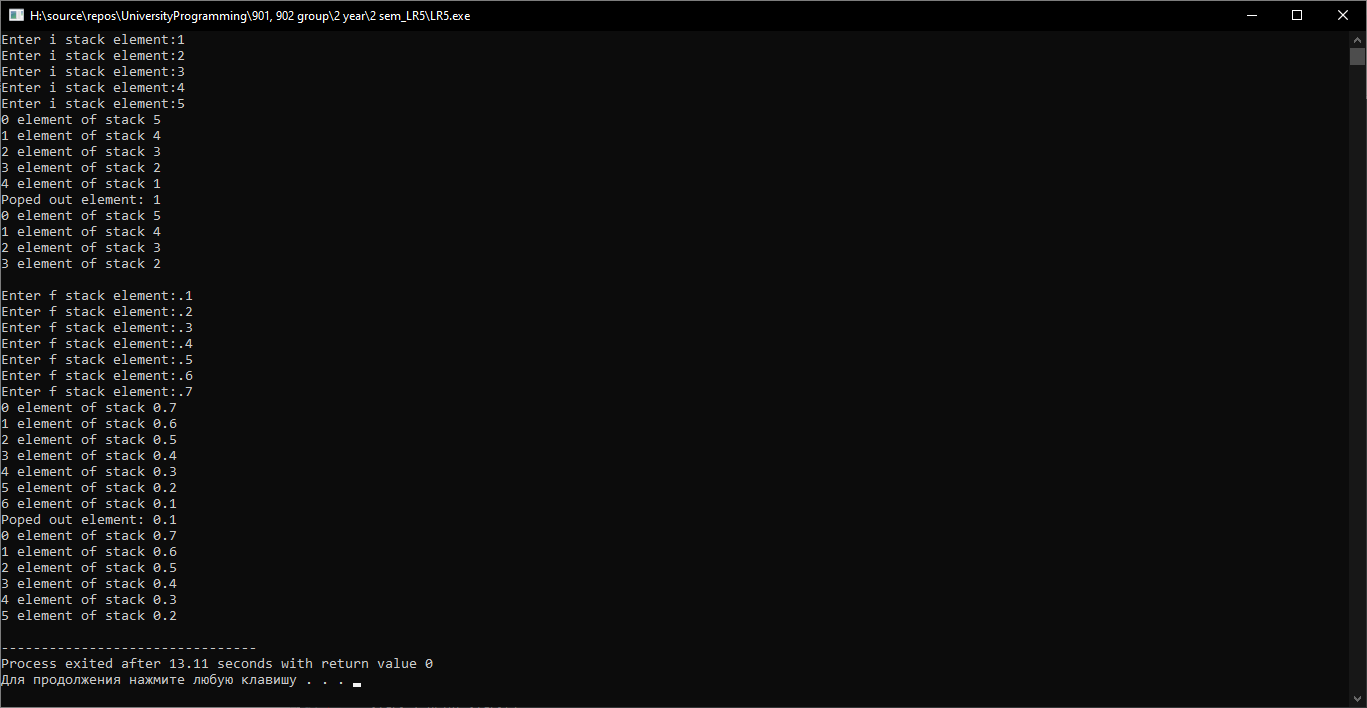
2. Написать консольную программу, демонстрирующую работу с экземплярами класса **Class\_Stack** на примере целого и вещественного типа данных. 

Рисунок 5.1 результат работы программы

Код:

#include <stdio.h>

#include <iostream>

#include <typeinfo>

#include <vector>

#include <algorithm>

class ErrorMsg{

public:

ErrorMsg(const char\* msg="Unknown Exception", int code=-1){

\_code=code;

\_msg=msg;

}

int GetMsgCode() {

return \_code;

}

char\* ShowMsg() {

printf("Caught error, code: %d, error: %s\n",GetMsgCode(),\_msg);

}

private:

const char\* \_msg;

int \_code;

};

template <typename T>

class Class\_Stack{

public:

Class\_Stack(){

initialize\_stack();

}

Class\_Stack(int size){

max\_size = size<=0?max\_size:size;

initialize\_stack();

}

void push(T element){

try{

if (stack.size() < max\_size)

stack.push\_back(element);

else

throw ErrorMsg("Stack overflow", 1);

}

catch(ErrorMsg &e){

e.ShowMsg();

exit(e.GetMsgCode());

}

}

T pop(){

try{

if (stack.size() > 0){

T temp = stack.back();

stack.pop\_back();

return temp;

}

else

throw ErrorMsg("Stack empty", 2);

}

catch(ErrorMsg &e){

e.ShowMsg();

exit(e.GetMsgCode());

}

}

void print\_stack(){

for (int i=0;i<stack.size();i++){

std::cout<<i<<" element of stack "<<stack[i]<<'\n';

}

}

private:

std::vector<T> stack;

int max\_size = 10;

void initialize\_stack(){

for (int i=0;i<max\_size;i++){

std::cout<<"Enter "<<typeid(T).name()<<" stack element:";

T temp;

std::cin>>temp;

push(temp);

}

std::reverse(stack.begin(),stack.end());

};

};

int main(){

Class\_Stack<int> stack\_i(5);

stack\_i.print\_stack();

std::cout <<"Poped out element: "<<stack\_i.pop()<<'\n';

stack\_i.print\_stack();

printf("\n");

Class\_Stack<float> stack\_f(7);

stack\_f.print\_stack();

std::cout <<"Poped out element: "<< stack\_f.pop()<<'\n';

stack\_f.print\_stack();

return 0;

}

Вывод: в ходе работы мы изучили создание шаблона класса на языке С++