Специальность: 09.02.07 Информационные системы и программирование

Выполнила:

Студент группыY2333

Максимов Д.О

ОТЧЕТ

о лабораторной работе № 7

по теме: Классы

по дисциплине: Разработка программных модулей

Санкт-Петербург 2019

Проверил:

Антонов. М.Б.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

УНИВЕРСИТЕТ ИТМО

Факультет среднего профессионального образования

1. Цель и задачи

Цель: реализация работы с классами, освоение основных принципов ООП.

Задачи:

1. Изучить классы.
2. Задание

Общее ограничение на лабораторную работу:

* использовать разбитие на файлы основной программы (в main.cpp только функция main.cpp);
* документировать весь код используя аннотации Doxygen.

Требования к лабораторной (Вариант 1):

* Создать шаблон класса A (Stack) с использование класса B (std::vector);
* Предусмотреть генерацию исключений;
* Показать использование класса.

1. КОд программы

Основной выполняемый файл main.cpp:

#include <iostream>

#include <vector>

#include <stdexcept>

using namespace std;

template<typename T> class **Stack**

{

**std**::**vector**<**T**> **storage**;

public:

inline bool **empty**() const;

inline size\_t size() const;

void **push**(T element);

T **pop**();

T **top**() const;

};

template<typename T> inline bool Stack<T>::**empty**() const

{ return storage.empty();

}

template<typename T> inline size\_t Stack<T>::size() const

{ return storage.size();

}

template<typename T> void Stack<T>::**push**(T element)

{ storage.push\_back(element);

}

template<typename T> T Stack<T>::**pop**()

{ if(empty())

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

T last = storage.back();

storage.pop\_back();

return last;

}

template<typename T> T Stack<T>::**top**() const

{ if(empty())

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

else return storage.back();

}

int **main**()

{

Stack<int> si;

for(int i = 0; i < 10; ++i)

si.push(10-i);

while(!si.empty())

cout << si.pop() << ' ';

cout << endl;

Stack<double> sd;

for(int i = 0; i < 5; ++i)

sd.push(i\*0.5), cout << sd.top() << ' ';

cout << endl;

Stack<std::string> ss;

try

{ ss.pop();

}

catch(exception &e)

{ cout << "Error: " << e.what();

}

}

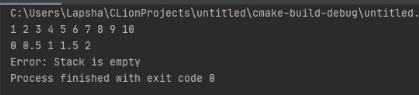


Рисунок 3 – Протокол программы

ВЫВОД

В процессе выполнения лабораторной работы были получены навыки по созданию самописных контейнеров.