Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6.2

Дисциплина: Основы алгоритмизации и программирования

Тема: Стек через классы.

|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Бреднев М.П. | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

**Анализ предметной области**

**Постановка задачи**

Реализовать стеки через классы.

**Тестирование программы**

****

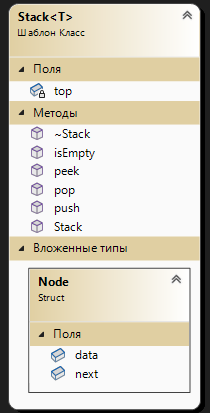
*Рис. 1 – Тестирование программы.*

**Заключение**

Была разработана программа, в которой реализованы стеки через классы.

**Приложения**

Приложение UML-диаграмма



Приложение Б – код программы

Main.cpp:

#include <iostream>

#include "CustomStack.h"

int main() {

Stack<int> stack;

stack.push(1);

stack.push(2);

stack.push(3);

std::cout << stack.peek() << std::endl;

stack.pop();

std::cout << stack.peek() << std::endl;

stack.pop();

std::cout << stack.peek() << std::endl;

stack.pop();

std::cout << stack.isEmpty() << std::endl;

return 0;

}

//В данной реализации стека с помощью односвязного списка создается класс Stack, содержащий приватное поле top, указывающее на вершину стека.

//Внутренний класс Node используется для хранения данных элемента стека и указателя на следующий элемент.

//Метод push добавляет новый элемент на вершину стека.

//Метод pop удаляет элемент с вершины стека и возвращает его значение.Метод peek возвращает значение элемента на вершине стека, не удаляя его.

//Метод isEmpty проверяет, пуст ли стек.

customstack.h:

#pragma once

template <typename T>

class Stack {

private:

struct Node {

T data;

Node\* next;

};

Node\* top;

public:

Stack() {

top = nullptr;

}

~Stack() {

while (!isEmpty()) {

pop();

}

}

void push(T item) {

Node\* newNode = new Node;

newNode->data = item;

newNode->next = top;

top = newNode;

}

T pop() {

if (isEmpty()) {

std::cerr << "Stack is empty" << std::endl;

exit(EXIT\_FAILURE);

}

Node\* temp = top;

T data = temp->data;

top = top->next;

delete temp;

return data;

}

T peek() {

if (isEmpty()) {

std::cerr << "Stack is empty" << std::endl;

exit(EXIT\_FAILURE);

}

return top->data;

}

bool isEmpty() {

return top == nullptr;

}

};