Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема : "Стек"

Выполнил работу:

Студент группы РИС-22-1Б

Батин В.В.

Проверил:

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

**Постановка задачи.**

Вариант 5:

Сформировать стек, добавить в него элементы, удалить из него элементы.

**Анализ задачи**

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.

**Код.**

#include <iostream>

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;

}

};

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;

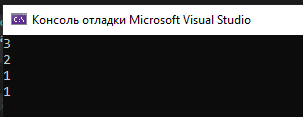
stack.pop();

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

return 0;

}

**Результат выполнения**

****