

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

ОТЧЁТ

«ЛАБОРАТОРНАЯ №11.2: СТЕК»

Дисциплина: «Программирование»

Выполнил:

Студент группы ИВТ-21-26

Безух Владимир Сергеевич

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь, 2022

Содержание

1.	Постановка задачи	3
2.	Анализ задачи.....	4
3.	Исходный код.....	5
4.	Анализ результатов.....	8

1. Постановка задачи

Продemonстрировать работу стека как структуры данных.

2. Анализ задачи

Стек можно реализовать на основе односвязного списка с ограниченным функционалом. Необходимо реализовать методы добавления узла в начало стека и удаления узла из начала стека для организации принципа LIFO (last in, first out).

3. Исходный код

```
#include <iostream>

template <typename T>
class Stack {
public:
    Stack();
    ~Stack();

    T top() const;
    size_t size() const;

    void push(const T& data);
    void pop();

    void clear();

private:
    struct Node {
        Node(T data = T(), Node* pointer_to_next_node = nullptr)
            : data(data), pointer_to_next_node(pointer_to_next_node) {}

        Node(const Node& copy)
            : data(copy.data), pointer_to_next_node(copy.pointer_to_next_node) {}

        Node& operator=(const Node& right) {
            if (this != &right) {
                data = right.data;
                pointer_to_next_node = right.pointer_to_next_node;
            }

            return *this;
        }

        T data;
        Node* pointer_to_next_node;
    };

    void pushFirstNode(Node* node);
    void pushFrontNode(Node* node);

    void popFirstNode();
    void popFrontNode();

    size_t stack_size;
    Node* head_node;
};

template<typename T>
Stack<T>::Stack()
    : stack_size(size_t{0}), head_node(nullptr) {}

template<typename T>
Stack<T>::~~Stack()
{
    clear();
}

template<typename T>
T Stack<T>::top() const
{
    return head_node->data;
}
```

```

template<typename T>
size_t Stack<T>::size() const
{
    return stack_size;
}

template<typename T>
void Stack<T>::push(const T& data)
{
    Node* new_node = new Node(data);
    stack_size ? pushFrontNode(new_node) : pushFirstNode(new_node);
    ++stack_size;
}

template<typename T>
void Stack<T>::pop()
{
    if (stack_size == size_t{0}) return;

    Node* remove_node = head_node;
    (stack_size == size_t{1}) ? popFirstNode() : popFrontNode();
    delete remove_node;
    --stack_size;
}

template<typename T>
void Stack<T>::clear()
{
    if (stack_size == size_t{0}) return;

    Node* remove;
    Node* next_node = head_node;

    while (stack_size) {
        remove = next_node;
        next_node = next_node->pointer_to_next_node;
        delete remove;
        --stack_size;
    }

    head_node = nullptr;
}

template<typename T>
void Stack<T>::pushFirstNode(Node* node)
{
    head_node = node;
}

template<typename T>
void Stack<T>::pushFrontNode(Node* node)
{
    node->pointer_to_next_node = head_node;
    head_node = node;
}

template<typename T>
void Stack<T>::popFirstNode()
{
    head_node = nullptr;
}

```

```

template<typename T>
void Stack<T>::popFrontNode()
{
    head_node = head_node->pointer_to_next_node;
}

int main()
{
    std::setlocale(LC_ALL, "Russian");

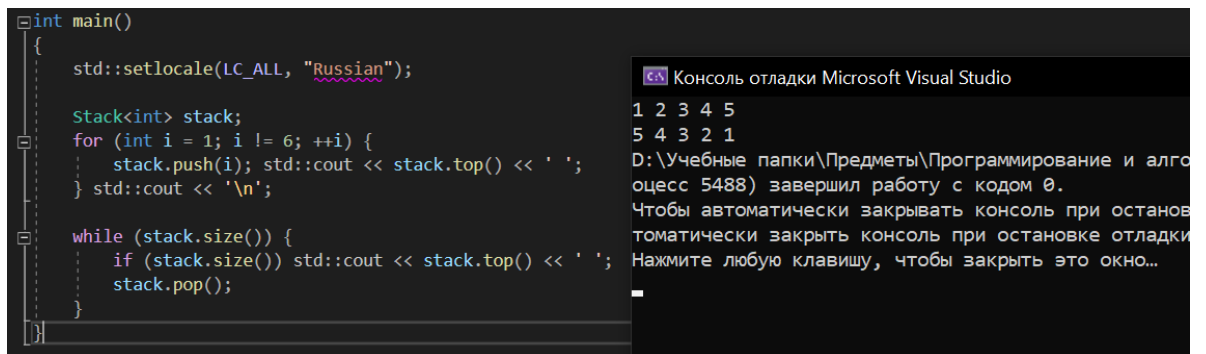
    Stack<int> stack;
    for (int i = 1; i != 6; ++i) {
        stack.push(i); std::cout << stack.top() << ' ';
    } std::cout << '\n';

    while (stack.size()) {
        if (stack.size()) std::cout << stack.top() << ' ';
        stack.pop();
    }
}

```

4. Анализ результатов

Результаты работы программы (рис. 1).



The image shows a screenshot of the Microsoft Visual Studio IDE. On the left, the C++ source code is displayed in a dark-themed editor. The code defines a `main` function that sets the locale to Russian, creates a stack, pushes numbers 1 through 5, and then prints the stack contents in reverse order. On the right, the 'Консоль отладки Microsoft Visual Studio' (Visual Studio Debug Console) shows the output of the program: the numbers 1 2 3 4 5 on the first line and 5 4 3 2 1 on the second line. Below the output, there is a message in Russian indicating that the process 5488 has finished with code 0, and instructions on how to close the console window.

```
int main()
{
    std::setlocale(LC_ALL, "Russian");

    Stack<int> stack;
    for (int i = 1; i != 6; ++i) {
        stack.push(i); std::cout << stack.top() << ' ';
    } std::cout << '\n';

    while (stack.size()) {
        if (stack.size()) std::cout << stack.top() << ' ';
        stack.pop();
    }
}
```

Консоль отладки Microsoft Visual Studio

1 2 3 4 5
5 4 3 2 1
D:\Учебные папки\Предметы\Программирование и алгоритмы 5488) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, нажмите любую клавишу, чтобы закрыть это окно...

Рисунок 1 — Результаты