

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Московский государственный технический университет  
имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления  
Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №8  
«Разработка шаблона класса»  
по курсу: «Языки и методы программирования»

Выполнил:  
Студент группы ИУ9-21Б  
Старовойтов А. И.

Проверил:  
Посевин Д. П.

Москва, 2022

## Цели

Целью данной работы является изучение шаблонов классов языка C++.

## Задачи

IntStack<L, N> – стек целых чисел из диапазона от L до N, имеющий стандартный для стека набор операций. Если размер диапазона не превышает 256, для представления стека должен использоваться массив char'ов. В случае диапазона, имеющего размер, не превышающий 65536, должен использоваться массив short'ов.

## Решение

### int\_stack.hpp

```
#ifndef LAB8_INT_STACK_HPP_
#define LAB8_INT_STACK_HPP_

#include <iostream>
#include <limits>
#include <stack>
#include <type_traits>

template <int L, int R,
          typename T = typename std::conditional<R - L <=
    ↪ std::numeric_limits<unsigned char>::max(),
          unsigned char,
          typename std::conditional<R - L <=
    ↪ std::numeric_limits<unsigned short>::max(),
          unsigned short,
          int>::type>::type>
class IntStack {
private:
    std::stack<T> stack;

public:
    void push(const int v);
```

```

    int top();
    void pop();
};

template <int L, int R, typename T>
void IntStack<L, R, T>::push(int v)
{
    stack.push(static_cast<T>(v - L));
}

template <int L, int R, typename T>
int IntStack<L, R, T>::top()
{
    return stack.top() + L;
}

template <int L, int R, typename T>
void IntStack<L, R, T>::pop()
{
    stack.pop();
}

template <int L, int R>
class IntStack<L, R, int> {
private:
    std::stack<int> stack;

public:
    void push(int v);
    int top();
    void pop();
};

template <int L, int R>
void IntStack<L, R, int>::push(int v)
{
    stack.push(v);
}

template <int L, int R>

```

```
void IntStack<L, R, int>::pop()
{
    stack.pop();
}
```

```
template <int L, int R>
int IntStack<L, R, int>::top()
{
    return stack.top();
}
```

```
#endif // LAB8_INT_STACK_HPP_
```

## main.cpp

```
#include "int_stack.hpp"
```

```
int main()
{
    {
        IntStack<0, 5> test;
        test.push(2);
        test.push(3);
        std::cout << test.top() << ' ';
        test.pop();
        std::cout << test.top() << '\n';
    }
    {
        IntStack<0, 10000> test;
        test.push(300);
        test.push(400);
        std::cout << test.top() << ' ';
        test.pop();
        std::cout << test.top() << '\n';
    }
    {
        IntStack<0, 123456789> test;
        test.push(999999);
        test.push(88888888);
        std::cout << test.top() << ' ';
    }
}
```

```
        test.pop();  
        std::cout << test.top() << '\n';  
    }  
    return 0;  
}
```