ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА"

Факультет инфокоммуникационных сетей и систем
Кафедра программной инженерии и вычислительной техники

ЛАБОРАТОРНАЯ РАБОТА №5

«ШАБЛОННЫ КЛАССОВ»

по дисциплине «ООП»

Вариант №13

Выполнил студент группы
ИКПИ-12
Музычук Артем

Санкт-Петербург
2022

1. Постановка задачи

В настоящей лабораторной работе необходимо решить две задачи, связанные с организацией шаблонов классов. Первая из задач состоит в преобразовании в шаблон класс того числового класса, который был разработан в первой лабораторной работе. Вторая задача состоит в разработке шаблона класса для стека, построенного на основе массива с фиксированными размерами. При решении второй задачи следует предусмотреть обработку исключительных ситуаций.

2. Описание полей и методов класса Complex

template <typename F, typename P> - шаблон класса;

F – части комплексного числа

Complex() – конструктор по умолчанию;

P summ(F *real, F *im);

P dif(F *real, F *im);

P mult(F *real, F *im);

P div(F *real, F *im);

3. Описание полей и методов класса Stack

template <typename T> - шаблон класса;

Stack() - конструктор

T* stackPtr; // указатель на стек

size_t NowSize;// текущий размер стека

size_t MaxSize;// общий размер стека

Stack &push(T &value) – поместить элемент в стек

T &pop() – удалить из стека

Void printStack() - печать на экран

~Stack() – диструктор

4. Код программы

Lab5_1.cpp

```
#include <iostream>
#include <cmath>

using namespace std;

template <typename F, typename P>

class Complex
{
private:
   F real[2], im[2];

public:
   // Конструктор по умолчанию
   Complex()
   {

      cout << "Введите действительную часть комплексного числа: ";
      cin >> real[0];
      cout << "Введите мнимую часть комплексного числа: ";
      cin >> im[0];
      if (im[0] >= 0)
         cout << "Введенное число: " << real[0] << "+" << im[0] << "i" << endl;
      else
      {
         cout << "Введенное число: " << real[0] << im[0] << "i" << endl;
      }
      cout << "Модуль комплексного числа = |" << (sqrt((pow(real[0], 2)) + (pow(im[0], 2)))) << "|" << endl;

      cout << endl
          << "\t [Введите 2-ое комплексное число для проведения арифметических операций]:" << endl;
      cout << "\nДействительная часть: ";
      cin >> real[1];
      cout << "Мнимая часть: ";
      cin >> im[1];
      if (im[1] >= 0)
         cout << "Введенное число: " << real[1] << "+" << im[1] << "i" << endl;
      else
      {
         cout << "Введенное число: " << real[1] << im[1] << "i" << endl;
      }
      cout << "Модуль комплексного числа = |" << (sqrt((pow(real[1], 2)) + (pow(im[1], 2)))) << "|" << endl;

      cout << endl
          << "\t [Результаты]" << endl
          << endl;
      summ(real, im);
      dif(real, im);
      mult(real, im);
      div(real, im);
      cout << endl;
```

```
    };
    // Метод "+"
    P summ(F *real, F *im)
    {
        cout << "Сумма введенных чисел = " << real[0] + real[1] << "+(" << (im[0] + im[1]) << ")i" << endl;
        return (0);
    }
    // Метод "-"
    P dif(F *real, F *im)
    {
        cout << "Разность введенных чисел = " << real[0] - real[1] << "+(" << (im[0] - im[1]) << ")i" << endl;
        return (0);
    }
    // Метод "*"
    P mult(F *real, F *im)
    {
        cout << "Произведение введенных чисел = " << (real[0] * real[1] - im[0] * im[1]) << "+(" << im[0] * real[1] +
real[0] * im[1] << ")i" << endl;
        return (0);
    }
    // Метод "/"
    P div(F *real, F *im)
    {
        cout << "Частное введенных чисел = " << (real[0] * real[1] + im[0] * im[1]) / (real[1] * real[1] + im[1] * im[1]) <<
"+(" << (im[0] * real[1] - real[0] * im[1]) / (real[1] * real[1] + im[1] * im[1]) << ")i" << endl;
        return (0);
    }
    ~Complex(){};
};

int main()
{
    int ans = 1, ch = 0;
    while (ans != 0)
    {
        // Задача 16 (Комплексные числа) :
        cout << "\t |КОМПЛЕКСНЫЕ ЧИСЛА|" << endl;
        Complex<float, int> first; // создание объекта класса
        break;
    }
    return 0;
}
```

Lab5_2.cpp

```
#include <iostream>
#include <cmath>

using namespace std;

#include <iomanip>

template <typename T>

class Stack {

private:
    T* stackPtr; // указатель на стек
```

```
    size_t NowSize;// текущий размер стека
    size_t MaxSize;// общий размер стека

public:
    //  конструктор по умолчанию
    Stack(int arg) {
        MaxSize = arg;   // инициализировать размер стека
        NowSize = 0;
        stackPtr = new T[MaxSize]{0}; // выделить память под стек

    }

    // деструктор
    ~Stack() {
        delete[] stackPtr; // удаляем стек
    }

    size_t size() { return NowSize; }

    // поместить элемент в стек
    Stack &push(T &value) {
        try
        {
            if (NowSize < MaxSize)
            {
                stackPtr[NowSize] = value;
                NowSize++;
                return *this;
            }
            else
            {
                throw 1;
            }
        }
        catch (int arg)
        {
            cout << "Ошибка! стек переполнен. максимальный размер = " << MaxSize << endl;
            return *this;
        }


    }

    // удалить из стека элемент
    T &pop()
    {
        T value = 0;
        if (NowSize > 0) {
            value = stackPtr[NowSize - 1], --NowSize;
        }
        return value;
    }

    // вывод стека
    void printStack() {
        for (int i = NowSize - 1; i >= 0; i--)
            cout << "|" << setw(4) << stackPtr[i] << endl;
```

```
    }
};


int main() {

    int n;
    cout << "Введите размер стека: ";
    cin >> n;

    double temp;
    Stack <double> myStack(n);
    // заполняем стек
    int count = 0;
    while (count++ != n+2) {
        cout << "Текущий размер стека = " << myStack.size() << endl;
        cout << "Максимальный размер стека = " << n << endl;
        cout << "----------------" << endl;
        cout << "Введите элементы стека: ";
        cin >> temp;


        myStack.push(temp);
        myStack.printStack();
    }

    cout << "----------------" << endl;
    cout << "Удаляем 2 верхних элемента стека" << endl;
    myStack.pop();
    myStack.pop();
    myStack.printStack();
    cout << "Текущий размер стека = "<< myStack.size() << endl;
    cout << "----------------" << endl;
    cout << "Введите число: ";
    cin >> temp;
    myStack.push(temp);
    myStack.printStack();
    cout << "Текущий размер стека = "<< myStack.size() << endl;
    cout << "Максимальный размер стека = " << n << endl;
    cout << "Конец работы" << endl;
    return 0;
}
```

5. Тесты:

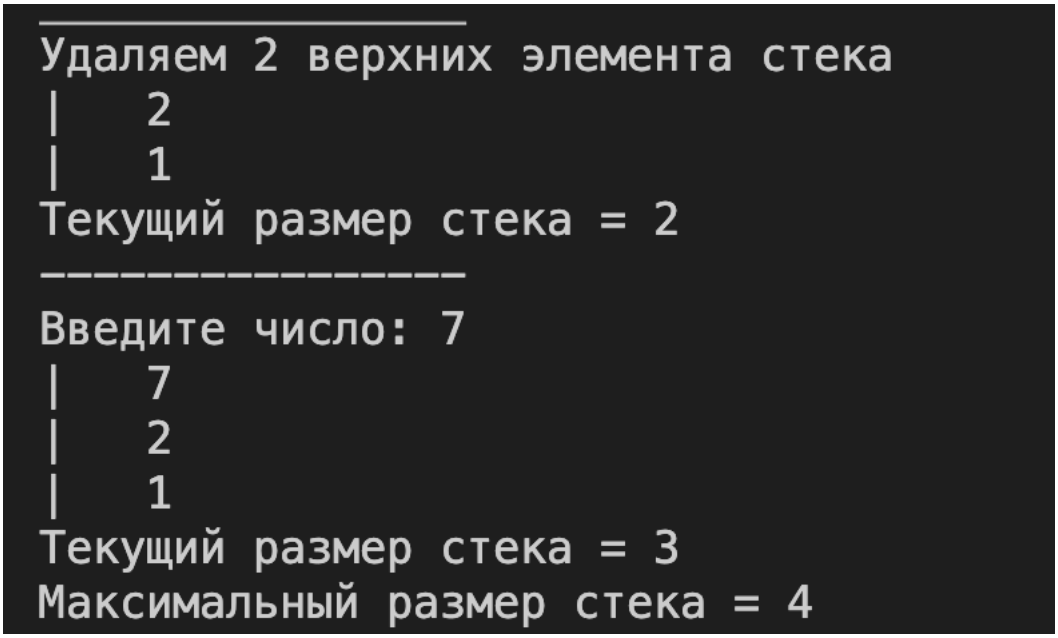
Lab5_1

```
         |КОМПЛЕКСНЫЕ ЧИСЛА|
Введите действительную часть комплексного числа: 3
Введите мнимую часть комплексного числа: 2
Введенное число: 3+2i
Модуль комплексного числа = |3.60555|

         [Введите 2-ое комплексное число для проведения арифметических операций]:

Действительная часть: 4
Мнимая часть: 5
Введенное число: 4+5i
Модуль комплексного числа = |6.40312|

         [Результаты]

Сумма введенных чисел = 7+(7)i
Разность введенных чисел = -1+(-3)i
Произведение введенных чисел = 2+(23)i
Частное введенных чисел = 0.536585+(-0.170732)i
```

Lab5_2

```
Введите размер стека: 4
Текущий размер стека = 0
Максимальный размер стека = 4
----------------
```

```
Введите элементы стека: 1
|   1
Текущий размер стека = 1
Максимальный размер стека = 4
----------------
Введите элементы стека: 2
|   2
|   1
Текущий размер стека = 2
Максимальный размер стека = 4
----------------
Введите элементы стека: 3
|   3
|   2
|   1
Текущий размер стека = 3
Максимальный размер стека = 4
----------------
Введите элементы стека: 4
|   4
|   3
|   2
|   1
Текущий размер стека = 4
Максимальный размер стека = 4
----------------
Введите элементы стека: 5
Ошибка! стек переполнен. максимальный размер = 4
```



6. Выводы

В ходе лабораторной работы, мы разработали шаблон для числового класса Complex, а также шаблон класса для стека, построенного на основе однонаправленного списка. Написанные тесты показали, что программа написана и выполнена верно.