

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе № 3

«Шаблоны классов»

Выполнил:

студент группы ИУ5-23

Терентьев Владислав

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Козлов А. Д.

Подпись и дата:

Москва, 2019 г.

1. Постановка задачи

Разработать шаблонный класс "MyStack" для хранения простых множителей целых чисел на базе односвязного списка (шаблонный класс `ListNode`) и глобальную функцию разложения числа на простые множители. Вывести простые множители числа 3960 из стека сначала по убыванию, а потом по возрастанию.

2. Разработка алгоритма

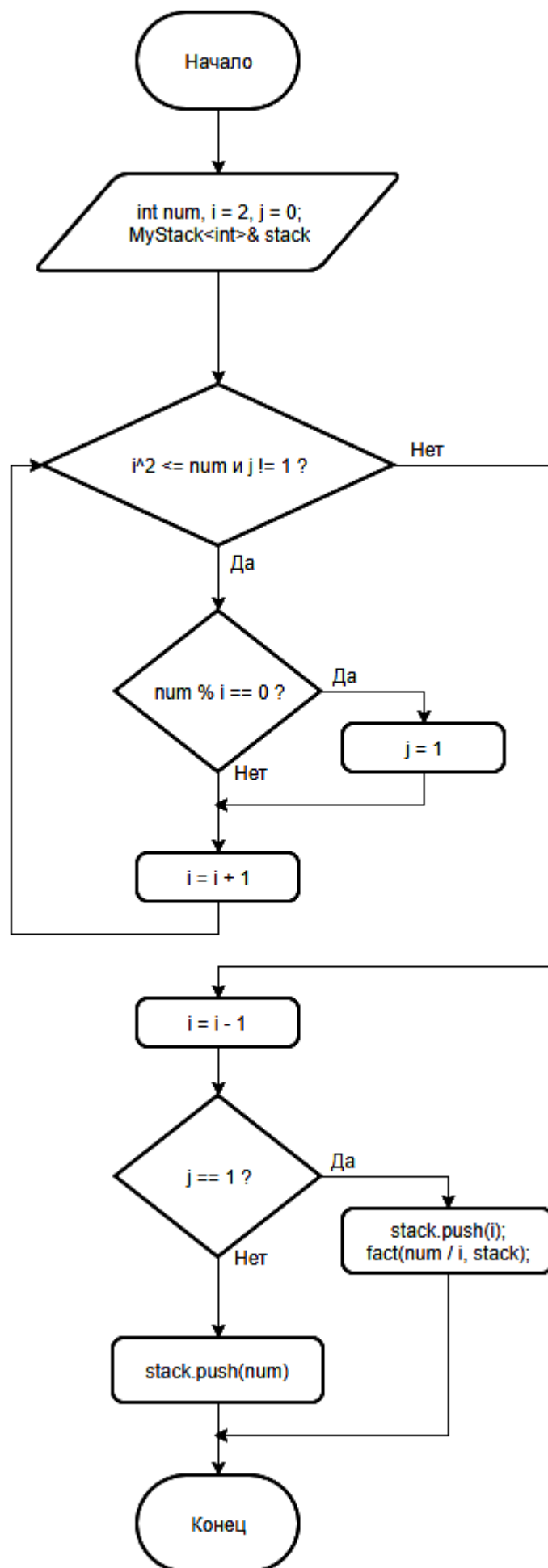
В программе описан шаблонный класс `ListNode` (параметры шаблона: `class INF`, `class FRIEND`) с переменными-членами (спецификатор доступа `private`): `d` типа `INF` для хранения значения; `next` указатель на объект класса `ListNode` для хранения адреса следующего элемента списка; `default` конструктор; и дружественный класс `FRIEND`.

Разработан шаблонный класс `MyStack` (параметры шаблона: `class INF`) с псевдонимом `Node` для шаблонного класса `ListNode` `<INF, MyStack <INF>>` и переменной-членом (спецификатор доступа `private`) `top` указатель на тип `Node` для хранения адреса "верхнего" объекта списка. Класс содержит (спецификатор доступа `public`): `default` конструктор; деструктор для очищения стека.

Методы: `empty` типа `bool` для проверки на пустоту стека; `push` типа `void` и параметром типа `INF` для добавления в стек нового элемента; `pop` типа `void` для удаления верхнего элемента из стека; `top_inf` типа `INF` для считывания информации из вершины стека.

Описана глобальная функция `fact` типа `void` и параметрами типа `(int, MyStack<int>&)` для разложения числа на простые множители и их запись в стек.

Схема алгоритма функции fact:



3. Текст программы

```
#include <iostream>

using namespace std;

template <class INF, class FRIEND>
class ListNode {
private:
    INF d;
    ListNode* next;
    ListNode() :next(NULL) {};
    friend FRIEND;
};

template <class INF>
class MyStack {
    typedef class ListNode <INF, MyStack <INF>> Node;
private:
    Node* top;
public:
    MyStack() :top(NULL) {}

    ~MyStack() {
        while (empty() != 1) {
            pop();
        }
    }

    bool empty() {
        if (top == NULL) {
            return 1;
        }
        else {
            return 0;
        }
    } // стек пустой?

    void push(INF n) {
        Node* temp = new Node;
        temp->d = n;
        temp->next = top;
        top = temp;
    }

    void pop() {
        Node* temp = top->next;
        delete top;
        top = temp;
        temp = NULL;
    }

    INF top_inf() {
        return top->d;
    }
};

void fact(int, MyStack<int>&);

void main()
{
    int number;
    MyStack<int> stack1, stack2;
    cin >> number;
    fact(number, stack1);
    cout << number << "=";
    cout << stack1.top_inf();
    stack2.push(stack1.top_inf());
    stack1.pop();
}
```

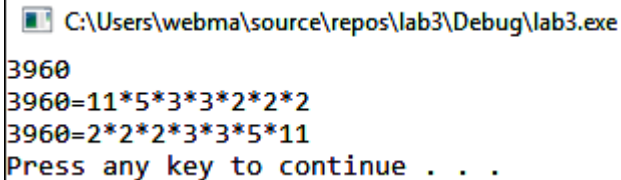
```

while (stack1.empty() != 1) {
    cout << "*" << stack1.top_inf();
    stack2.push(stack1.top_inf());
    stack1.pop();
}
cout << endl << number << "=";
cout << stack2.top_inf();
stack2.pop();
while (stack2.empty() != 1) {
    cout << "*" << stack2.top_inf();
    stack2.pop();
}
cout << endl;
system("pause");
}

void fact(int num, MyStack<int>& stack) {
    int i = 2, j = 0;
    while ((i * i <= num) && (j != 1)) {
        if (num % i == 0) {
            j = 1;
        }
        i = i + 1;
    }
    i--;
    if (j == 1) {
        stack.push(i);
        fact(num / i, stack);
    }
    else {
        stack.push(num);
    }
}
}

```

4. Анализ результатов



```

C:\Users\webma\source\repos\lab3\Debug\lab3.exe
3960
3960=11*5*3*3*2*2*2
3960=2*2*2*3*3*5*11
Press any key to continue . . .

```

Программа работает исправно.