**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №3

«Стек»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-21Б |  | преподаватель каф. ИУ5 |
| Михалёв Я.М. |  | Козлов А.Д. |
| Подпись и дата: |  | Подпись и дата: |
|  |  |  |



Москва, 2022 г.

Постановка задачи

Для заданного целого **n**>1 наименьшее целое **d**>1, на которое делится **n** – **простой множитель**. Разложение **n** на простые множители можно произвести поиском **d** с дальнейшей заменой **n** на частное от деления **n** на **d** и повторяя этот процесс, пока **n** не станет равным 1. Написать программу разложения **n** на простые множители этим способом с печатью простых множителей сначала по убыванию, а потом – по возрастанию. **ИСПОЛЬЗУЙТЕ СТЕК!** Например, для **n=3960** Ваша программа должна дать

**11 \* 5 \* 3 \* 3 \* 2 \* 2 \* 2**

и затем

**2 \* 2 \* 2 \* 3 \* 3 \* 5 \* 11**

Разработка алгоритма

**Класс Stack:**

Свойства:

* Elem\* Current – указатель на текущий элемент стека
* int n – количество элементов стека

Методы:

* push() – добавляет элемент в стек
* pop() – возвращает верхний элемент стека
* empty() – возвращает true если стек пустой, иначе false
* size() – возвращает количество элементов стека

**Список функций**

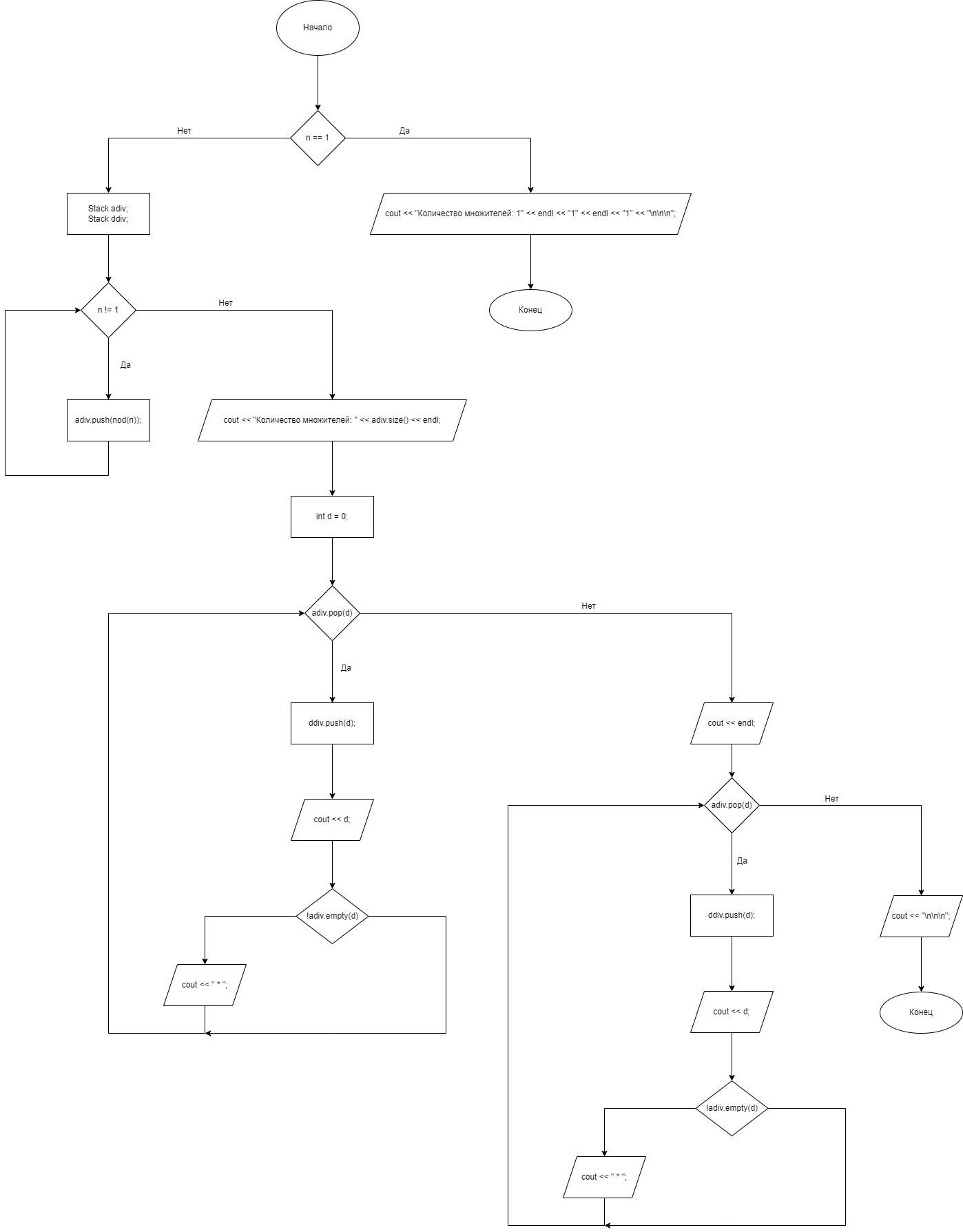
**getInt –** принимает целое положительное число от пользователя

**Nod –** возвращает наибольший общий делитель двух чисел

**Calculate** – принимает целое положительное число и выводит на экран его простые делители

**Схема алгоритма**

Calculate

****

Текст программы

Main.cpp

#include <iostream>

#include "Stack.h"

**using** **namespace** std**;**

int nod**(**int**&** n**)** **{**

int d **=** 2**;**

**while** **(**n **%** d **!=** 0**)** **{**

d**++;**

**}**

n **/=** d**;**

**return** d**;**

**}**

void Calculate**(**int n**)**

**{**

**if** **(**n **==** 1**)**

**{**

cout **<<** "Количество множителей: 1" **<<** endl **<<** "1" **<<** endl **<<** "1" **<<** "\n\n\n"**;**

**return;**

**}**

Stack adiv**;**

Stack ddiv**;**

**while** **(**n **!=** 1**)**

**{**

adiv**.**push**(**nod**(**n**));**

**}**

cout **<<** "Количество множителей: " **<<** adiv**.**size**()** **<<** endl**;**

int d **=** 0**;**

**while** **(**adiv**.**pop**(**d**))**

**{**

ddiv**.**push**(**d**);**

cout **<<** d**;**

**if** **(!**adiv**.**empty**(**d**))**

cout **<<** " \* "**;**

**}**

cout **<<** endl**;**

**while** **(**ddiv**.**pop**(**d**))**

**{**

cout **<<** d**;**

**if** **(!**ddiv**.**empty**(**d**))**

cout **<<** " \* "**;**

**}**

cout **<<** "\n\n\n"**;**

**}**

int getInt**()**

**{**

double num**;**

cout **<<** "Введите число для разложения\n> "**;**

cin **>>** num**;**

**while** **(**num **!=** **static\_cast<**int**>(**num**)** **||** num **<=** 0**)**

**{**

cout **<<** "Число должно быть целым и положительным!\n> "**;**

cin **>>** num**;**

**}**

cin**.**ignore**(**80**,** '\n'**);**

**return** **static\_cast<**int**>(**num**);**

**}**

int main**()**

**{**

system**(**"chcp 1251 > nul"**);**

setlocale**(**LC\_ALL**,** "rus"**);**

**while** **(true)**

**{**

int n **=** getInt**();**

Calculate**(**n**);**

**}**

**}**

Stack.h

#include <cstddef>

class Elem **{**

public**:**

int Info**;**

Elem**\*** Next**;**

Elem**()** **:**Next**(NULL),** Info**(**0**)** **{}**

Elem**(**int info**)** **:**Next**(NULL),** Info**(**info**)** **{}**

**};**

class Stack

**{**

private**:**

Elem**\*** Current**;**

int n**;**

public**:**

Stack**()** **{**

Current **=** **NULL;**

**}**

inline **~**Stack**();**

inline void push**(**int info**);**

inline bool pop**(**int**&** info**);**

inline bool empty**(**int info**);**

inline int size**();**

**};**

Stack**::~**Stack**()** **{**

**while** **(**Current**)** **{**

Elem**\*** temp **=** Current**->**Next**;**

**delete** Current**;**

Current **=** temp**;**

n**--;**

**}**

**}**

void Stack**::**push**(**int info**)** **{**

Elem**\*** temp **=** **new** Elem**(**info**);**

temp**->**Next **=** Current**;**

Current **=** temp**;**

n**++;**

**}**

bool Stack**::**pop**(**int**&** info**)** **{**

**if** **(**Current **==** **NULL)**

**return** **false;**

info **=** Current**->**Info**;**

Elem**\*** temp **=** Current**->**Next**;**

**delete** Current**;**

Current **=** temp**;**

n**++;**

**return** **true;**

**}**

bool Stack**::**empty**(**int info**)** **{**

**return** Current **==** **NULL;**

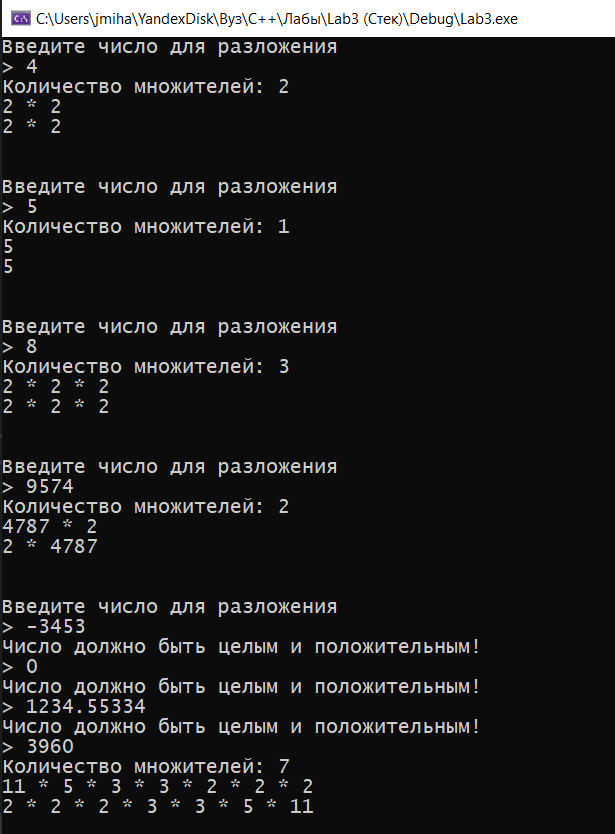
**}**

int Stack**::**size**()** **{**

**return** n**;**

**}**

Анализ результатов



Вывод

Я научился

* Работать со стеком
* Обрабатывать данные, полученные от пользователя