МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

« МИРЭА Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Вычислительной техники

УЧЕБНОЕ ЗАДАНИЕ

по дисциплине

« Объектно-ориентированное программирование»

Наименование задачи:

« Задача 3_1_3 »

С тудент группы	ИНБО-07-21	Михайлюк Д.С.
Руководитель практики	Ассистент	Асадова Ю.С.
Работа представлена	«___»_____ 2022 г.	_____
		(подпись студента)
Оценка		_____
		(подпись руководителя)

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Постановка задачи

Создать класс для объекта стек. Стек хранит целые числа. Имеет характеристики: наименование (строка, не более 10 символов) и размер (целое). Размер стека больше или равно 1. Функционал стека:
- добавить элемент и вернуть признак успеха (логическое);
- извлечь элемент (НЕ вывести!) и вернуть признак успеха (логическое);
- получить имя стека (строка);
- получить размер стека (целое);
- получить текущее количество элементов в стеке (целое).

В классе определить параметризированный конструктор, которому передается имя стека и размер. При переполнении стека очередной элемент не добавлять и определяется соответствующий признак успеха.

В основной программе реализовать алгоритм:

1. Ввести имя и размер для первого стека.
2. Создать объект первого стека.
3. Ввести имя и размер для второго стека.
4. Создать объект второго стека.
5. В цикле:
    5.1. Считывать очередное значение элемента.
    5.2. Добавлять элемент в первый стек, при переполнении завершить цикл.
    5.3. Добавлять элемент во второй стек, при переполнении завершить цикл.
6. Построчно вывести содержимое стеков.

Описание входных данных

Первая строка:

«имя стека 1»«размер стека»

Вторая строка:

«имя стека 2»«размер стека»

Третья строка:

Последовательность целых чисел, разделенных пробелами, в количестве не менее чем размер одного из стеков + 1.

Описание выходных данных

Первая строка:

«имя стека 1»«размер»

Вторая строка:

«имя стека 2»«размер»

Третья строка:

«имя стека 1»«имя стека 2»

Каждое имя стека в третьей строке занимает поле длины 15 позиции и прижата к левому краю.

Четвертая строка и далее построчно, вывести все элементы стеков: «значение элемента стека 1»«значение элемента стека 2»

Вывод значений элементов стеков производиться последовательным извлечением.

Каждое значение занимает поле из 15 позиции и прижата к правому краю.

Метод решения

Для решения поставленной задачи необходимо использовать:

- объекты стандартных потоков класса iostream для ввода и вывода информации на экран (cin и cout);
- объекты s1 и s2 класса Stack.

Класс Stack:

- свойства/поля:
    - Поле - хранит длинну стека текущего объекта
        - Наименование - n;
        - Тип - целочисленный;
        - Модификатор доступа - private;
    - Поле - хранит имя стека
        - Наименование - name;
        - Тип - строковый;
        - Модификатор доступа - private;
    - Поле - хранит текущую длинну стека
        - Наименование - name;
        - Тип - сторковый;
        - Модификатор доступа - cur r;
    - Поле - Хранит текущий стек
        - Наименование - a;
        - Тип - целочисленный массив;
        - Модификатор доступа - private;
- Функционал:
    - Метод - Stack;

    - Функционал - параметризированный конструктор с целочисленными параметрами длинны стека n и стоковым параметром name - название стека
- Метод - addEl
    - Функционал - добавляет эдлемент в конец стека
- Метод - getName
    - Функционал -возвращаетпоследний элемент стека
- Метод - getLen
    - Функционал - возвращает длинну стека
- Метод - getCurLen
    - Функционал - возвращает текущую длинну стека
- Метод - ~Stack
    - Функционал - Деструктор удаляющий динамическуб память.

Описание алгоритма

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

Функция: main

Функционал: Основной алгоритм программы

Параметры: отсутствуют

Возвращаемое значение: Цклочисленное значение - код возврата

Алгоритм функции представлен в таблице 1.

Таблица 1. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление строковых переменных name1 и name2	2	
2		Объявление целочисленных переменных n1 и n2	3	
3		Считывание с клавиатуры значения переменных name1, n1	4	
4		Объявление объекта s1 класса stack с параметрами name1, n1	5	
5		Считывание с клавиатуры значения переменных name2, n2	6	

6		Объявление объекта s2 класса stack с параметрами name2, n2	7	
7		Объявление целочисленной переменной next_el	8	
8		Считывание с клавиатуры значения переменной next_el	9	
9	Метод добавления элементов первого и второго и второго стека выполнился		9	
		Объявление целочисленных переменных m1 и m2,i. инициализация переменной i = 0	10	
10	i < минимальная длинна двух стеков	вызов метода getEl щбъектов s1 s2 с параметрами m1 m2	11	
			13	
11		вывод на экран значения переменных m1 m2	12	
12		Увеличение i на единицу	10	
13	Длинна первого стека больше длинны второго	вывод на экран результата метода getEl от объекта s1 со значением переменной m1	∅	
			∅	

Класс объекта: Stack

Модификатор доступа: public

Метод: Stack

Функционал: Создание объекта с текущими параметрами

Параметры: Строковое name - имя стека, целочисленный n - длинна стека, cur_n - текущая длина стека

Возвращаемое значение: ссылка на параметр

Алгоритм метода представлен в таблице 2.

Таблица 2. Алгоритм метода Stack класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1		Присвоение текущему объекту свойство name сооответствующего параметра. аналогично со свойством n	2	
2		инициализация переменной cur_n = 0	3	
3		выделение памяти для поля a	∅	

Класс объекта: Stack

Модификатор доступа: public

Метод: addEl

Функционал: Добаляет элемент в конец стека

Параметры: целочисленный el - добавляемый элемент

Возвращаемое значение: True - если операция была успешна, false - если нет

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода addEl класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1	Текущая длинна строки равнв заданной	Возвращает False	∅	
		Добавление el в конец стека	2	
2		Увеличение cur_n на единицу	3	
3		Возвращение true	∅	

Класс объекта: Stack

Модификатор доступа: public

Метод: getEl

Функционал: возвращает последний объект из стека

Параметры: целочисленная ссылка m - возвращаемый объект

Возвращаемое значение: True - если операция была успешна, false - если нет

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода getEl класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1	текущий размер массива больше нуля	Извлечение последнего жлемениа из стека	2	
		Возвращает false	∅	
2		Увеличивает cur_n на единицу	3	
3		Возвращение true	∅	

Класс объекта: Stack

Модификатор доступа: public

Метод: getCurLen

Функционал: Возвращает текущую длинну стека

Параметры: отсутствуют

Возвращаемое значение: Цеслочисленный cur_n - текущая длинна стека

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода getCurLen класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1		Возвращение значение поля cur_n	∅	

Класс объекта: Stack

Модификатор доступа: public

Метод: getLen

Функционал: возвращает общую длинну стека

Параметры: отсутствуют

Возвращаемое значение: целочисленная n - общая длинна

Алгоритм метода представлен в таблице 6.

Таблица 6. Алгоритм метода getLen класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1		Возвращает значение поля n	∅	

Класс объекта: Stack

Модификатор доступа: public

Метод: ~Stack

Функционал: Удаление динамически выделенной памяти

Параметры: отсутствуют

Возвращаемое значение: void

Алгоритм метода представлен в таблице 7.

Таблица 7. Алгоритм метода ~Stack класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1		Удаление динамически выделенной памяти	∅	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

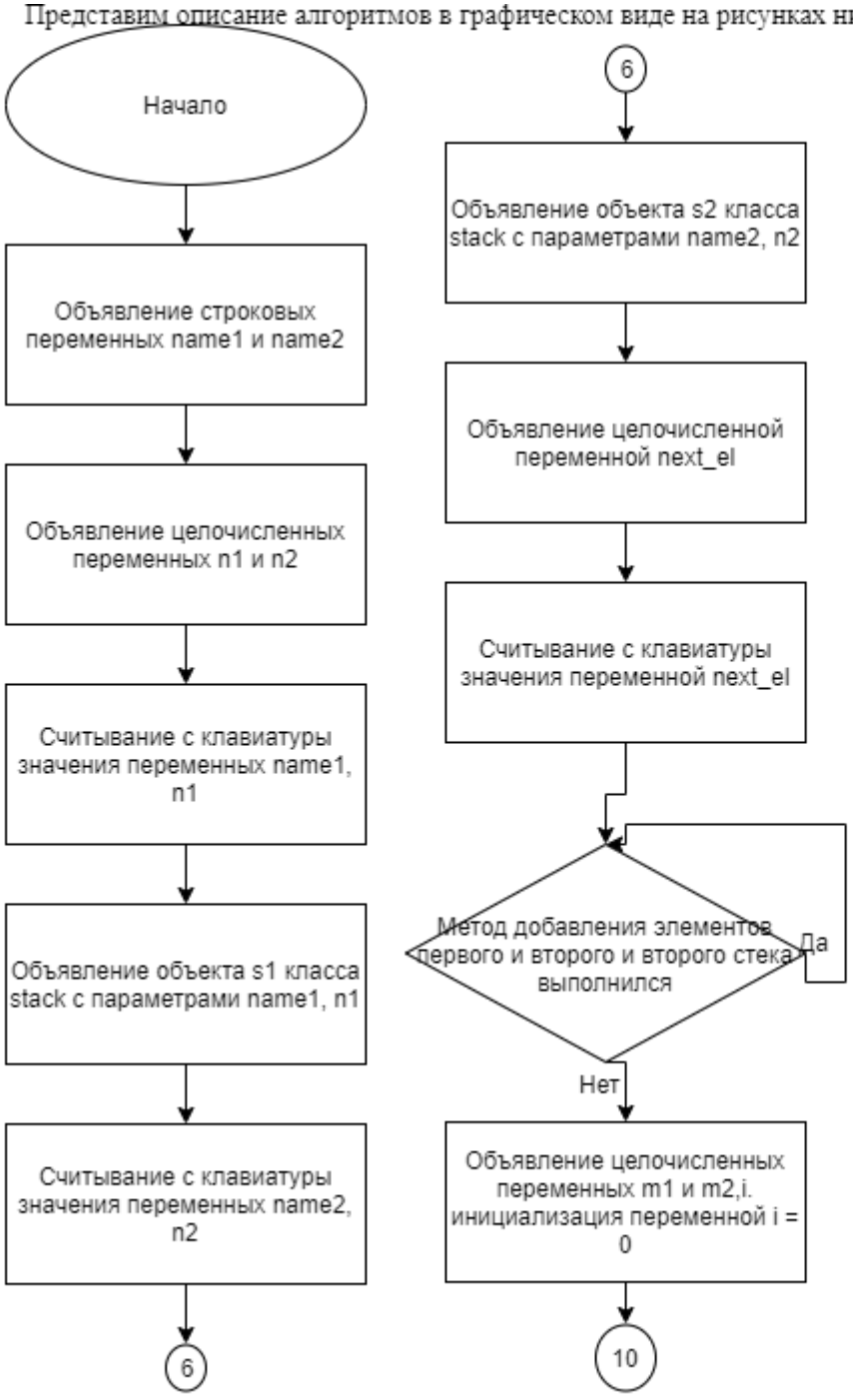

Рис. 1. Блок-схема алгоритма.

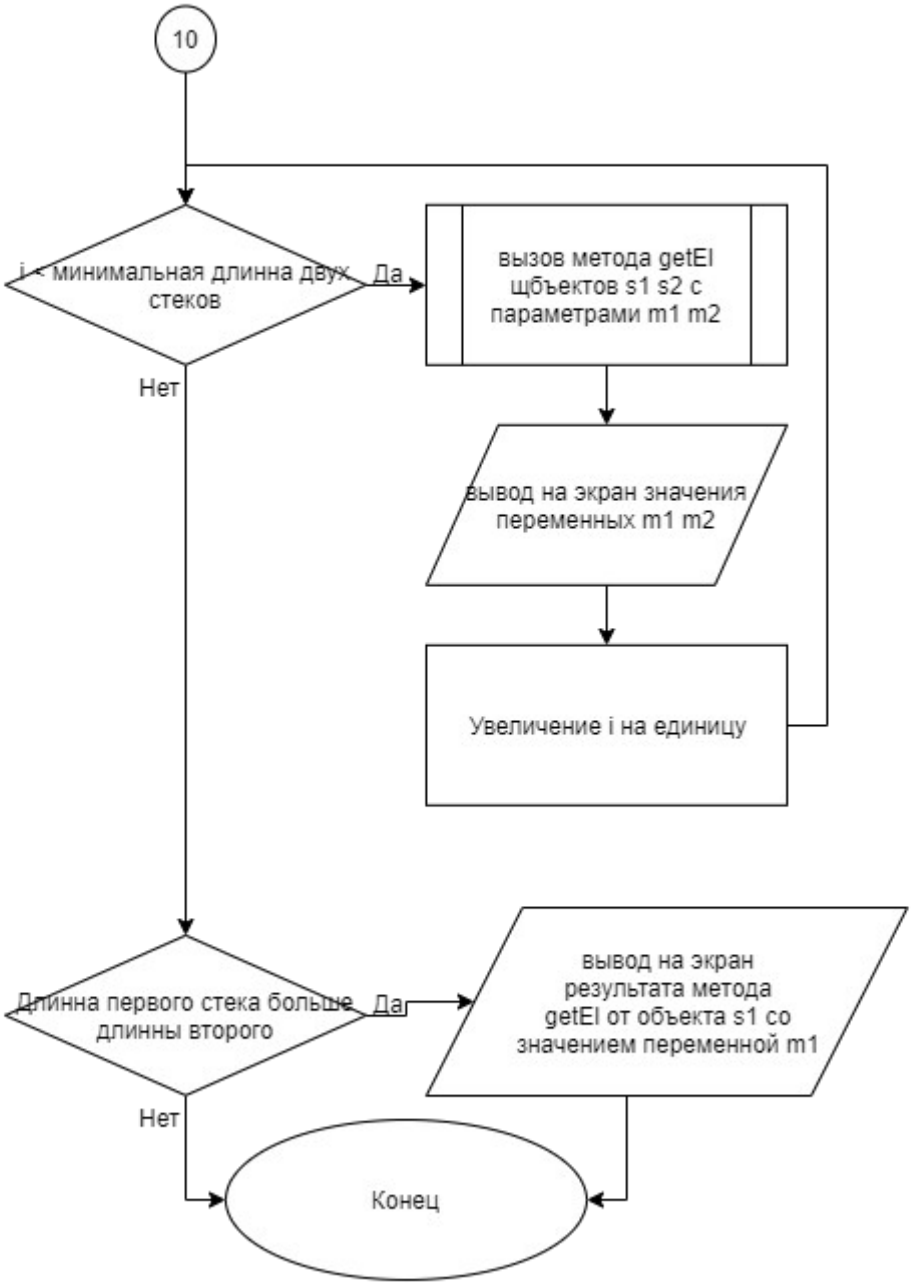

Рис. 2. Блок-схема алгоритма.

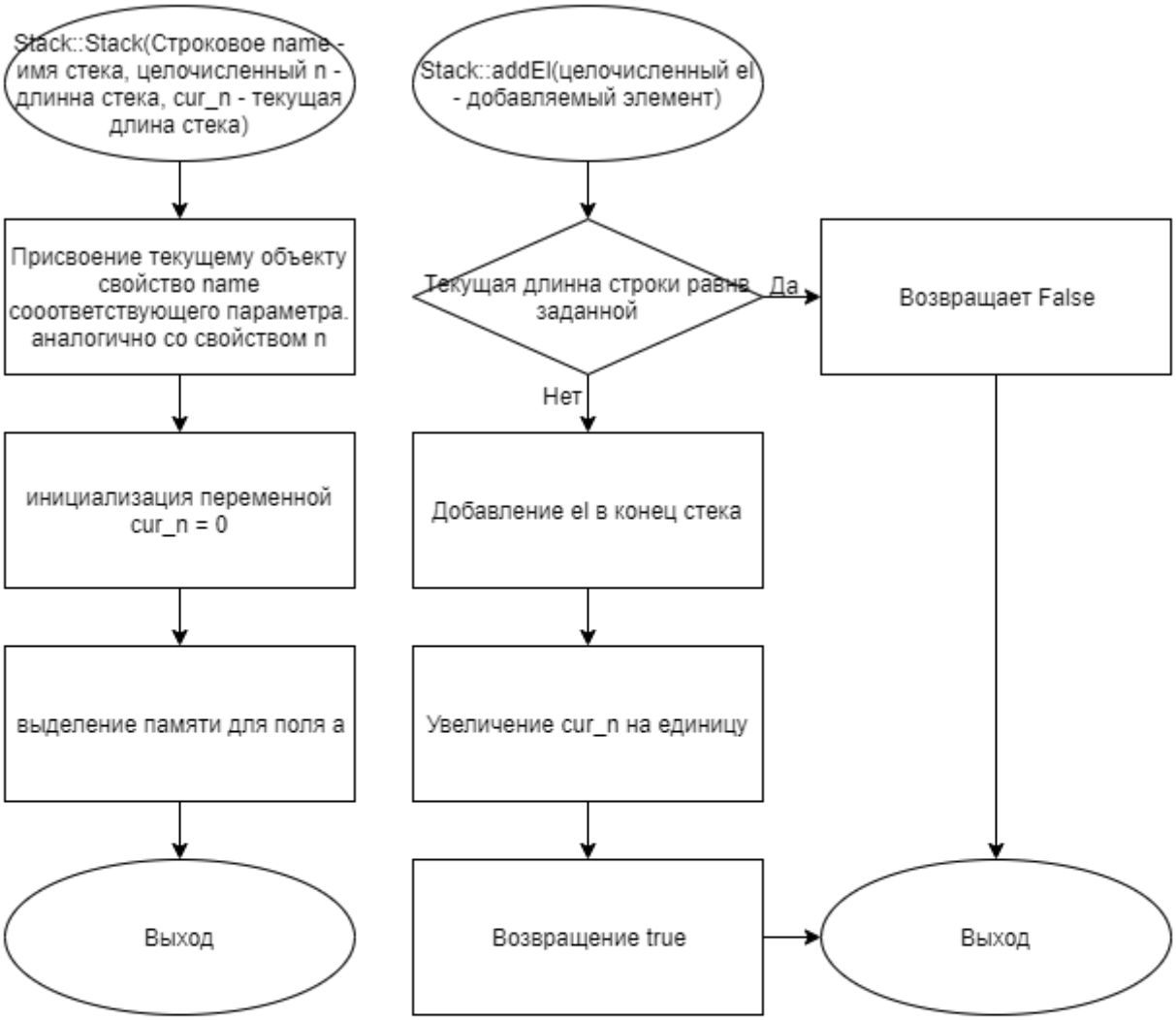

Рис. 3. Блок-схема алгоритма.

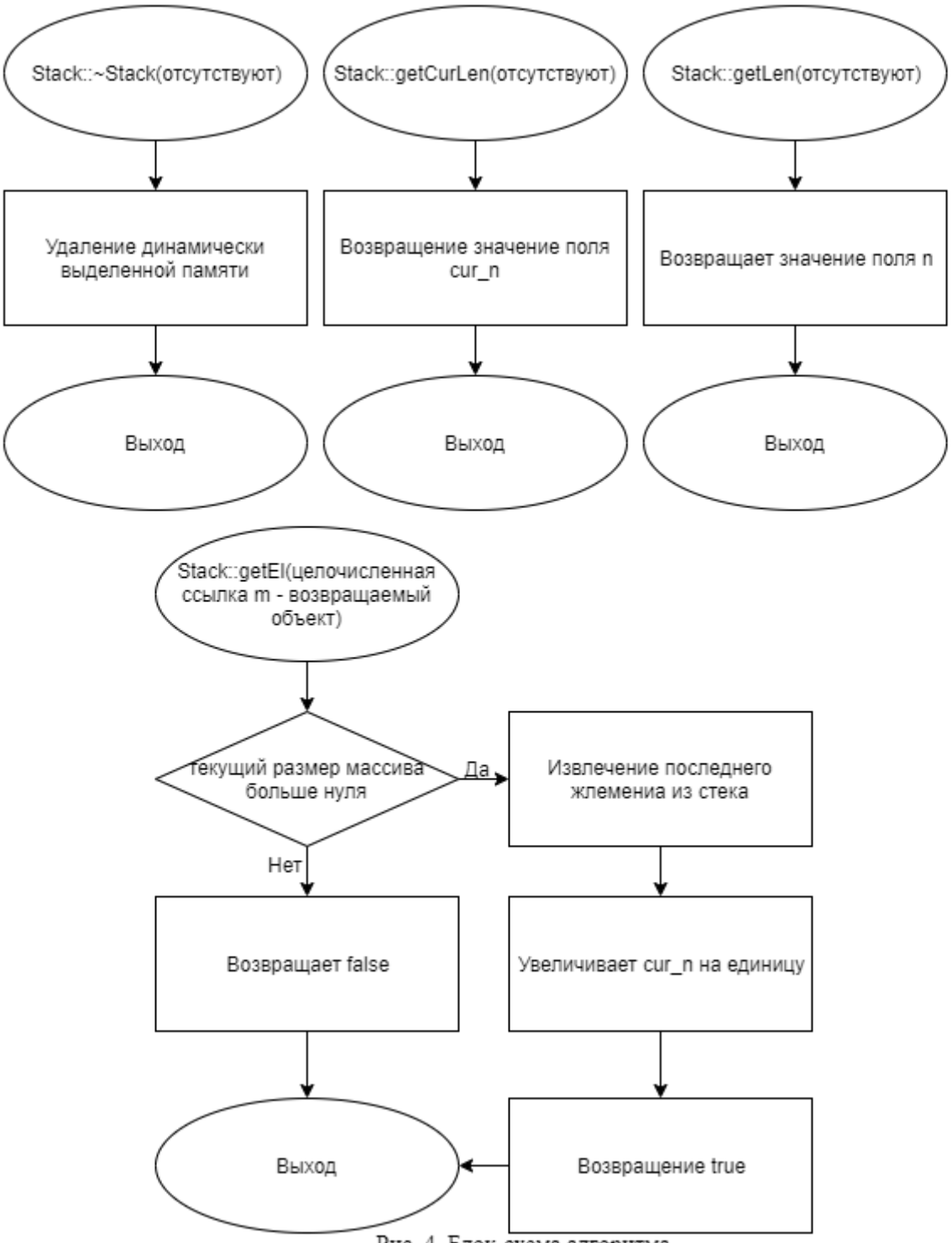

Рис. 4. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <iomanip>
#include <string>
#include "Stack.h"
using namespace std;
int main()
{
        string name1, name2;
        int n1, n2;
        cin >> name1 >> n1;
        Stack s1(name1, n1);
        cin >> name2 >> n2;
        Stack s2(name2, n2);
        int next_el;
        cout << name1 << " " << n1 << endl;
        cout << name2 << " " << n2 << endl;
        cin >> next_el;
        while(s1.addEl(next_el) and s2.addEl(next_el))
        {
                cin >> next_el;
        }
        int m1, m2;
        cout << left << setw(15) << name1;
        cout << left << setw(15) << name2;
        cout << endl;
        for (int i=0; i < min(s1.getLen(), s2.getLen()); i++)
        {
                s1.getEl(m1);
                cout << right << setw(15) << m1;
                s2.getEl(m2);
                cout << right << setw(15) << m2;
                if (i != min(s1.getLen(), s2.getLen()) - 1)
                {
                        cout << endl;
                }
        }
        if (s1.getLen() > s2.getLen())
        {
                s1.getEl(m1);
                cout << endl;
                cout << right << setw(15) << m1;
        }
        return(0);
}
```

Файл Stack.cpp

```
#include "Stack.h"
#include <iostream>
#include <string>
using namespace std;

Stack::Stack(string name,const int n)
{
        this -> name = name;
        this -> n = n;
        this -> cur_n = 0;
        a = new int[n];
};
bool Stack::addEl(int el)
{
        if(this -> getLen() == this -> getCurLen())
        {
                return false;
        }
        this -> a[this -> getCurLen()] = el;
        this -> cur_n++;
        return true;
};
bool Stack::getEl(int &m)
{
        if(this -> getCurLen() > 0)
        {
                m = this -> a[this -> getCurLen()- 1];
                this -> cur_n -= 1;
                return true;
        }
        return false;
};
int Stack::getCurLen()
{
        return cur_n;
};
int Stack::getLen()
{
        return n;
};
string Stack::getName()
{
        return name;
};
Stack::~Stack()
{
        delete a;
};
```

Файл Stack.h

```
#ifndef STACK_H
#define STACK_H
#include <string>
using namespace std;
class Stack
{
        private:
                string name;
                int n;
                int cur_n;
                int *a;
        public:
                Stack(string name, int n);
                bool addEl(int el);
                bool getEl(int &m);
                string getName();
                int getLen();
                int getCurLen();
                ~Stack();
};
#endif
```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
s1 1 s2 1 1 2	s1 1 s2 1 s1 s2 1 1	s1 1 s2 1 s1 s2 1 1
s1 3 s2 4 1337 8 567 2007	s1 3 s2 4 s1 s2 567 567 8 8 1337 1337	s1 3 s2 4 s1 s2 567 567 8 8 1337 1337
s1 2 s2 1 1 2	s1 2 s2 1 s1 s2 2 1 1	s1 2 s2 1 s1 s2 2 1 1

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ)

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.

3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).

4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».

6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).