



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

ИКБО-13-21

Дамарад Д.В.

Руководитель практики

Ассистент

Асадова Ю.С.

Работа представлена

«__»_____ 2022 г.

(подпись студента)

Оценка

(подпись руководителя)

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	
Постановка задачи.....	
Метод решения.....	
Описание алгоритма.....	
Блок-схема алгоритма.....	
Код программы.....	
Тестирование.....	
ЗАКЛЮЧЕНИЕ.....	
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	

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 параметризованный 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 производится последовательным извлечением.

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

Метод решения

Для решения поставленной задачи используются:

- Объекты стандартных потоков `cin` и `cout`. Используется для ввода с клавиатуры и вывода на экран.
- Объекты `name1`, `name2` класса `string` для хранения наименования.
- Объекты `s1`, `s2` класса `Stack`.
- Условный оператор `if` и операторы сравнения.
- Унарный оператор `"++"` для увеличения значения переменной.
- Унарный оператор `"--"` для уменьшения значения переменной.
- Манипулятор `setw` для установки ширины вывода.

Класс `Stack`:

- Поля:
 - Поле, отвечающее за хранения наименования:
 - Наименование - `name`;
 - Тип - `string`;
 - Модификатор доступа - `private`.
 - Поле, отвечающее за хранение размера:
 - Наименование - `n`;
 - Тип - `int`;
 - Модификатор доступа - `private`.
 - Поле, отвечающее за хранение текущей длины массива:
 - Наименование - `cur_n`;
 - Тип - `int`;
 - Модификатор доступа - `private`.
 - Поле, отвечающее за хранение указателя на массив:
 - Наименование - `a`;

- Тип - int*;
- Модификатор доступа - private.
- Методы:
 - Конструктор Stack:
 - Функционал - параметризованный конструктор, которому передается имя стека и размер.
 - Метод addEl:
 - Функционал - добавляет элемент и возвращает логический признак успеха.
 - Метод getEl:
 - Функционал - извлекает элемент и возвращает логический признак успеха.
 - Метод getName:
 - Функционал - получает имя стека.
 - Метод getLen:
 - Функционал - получает размер стека.
 - Метод getCurLen:
 - Функционал - получает текущее количество элементов в стеке.
 - Деструктор ~Stack:
 - Функционал - деструктор без параметров.

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: string name1,name2(Переменные для хранения имен стеков); int n1, n2(Переменные для хранения размеров стека); int next_el (переменная для хранения элемента, который нужно записать в массив стека); int m1,m2 (Переменные в которые будут извлекать элементы)

Возвращаемое значение: Целочисленное значение - код возврата

Алгоритм функции представлен в таблице 1.

Таблица 1. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация строковых переменных name1, name2	2	
2		Инициализация целочисленных переменных n1,n2	3	
3		Считывание с клавиатуры значений строковой переменной name1 и целочисленной переменной n1	4	
4		Объявление объекта s1 класса Stack c	5	

		помощью конструктора с параметрами name1, n1		
5		Считывание с клавиатуры значений строковой переменной name2 и целочисленной переменной n2	6	
6		Объявление объекта s2 класса Stack с помощью конструктора с параметрами name2, n2	7	
7		Инициализация целочисленной переменной next_el	8	
8		Вывод на экран значения строковой переменной name1, " ", значения целочисленной переменной n1	9	
9		Вывод на экран значения строковой переменной name2, " ", значения целочисленной переменной n2	10	
10		Считывание с клавиатуры значений целочисленной переменной next_el	11	
11	Метод addEl с параметром next_el	Считывание с клавиатуры	11	

	объекта s1 класса Stack = true и Метод addEl с параметром next_el объекта s2 класса Stack = true	значений целочисленной переменной next_el		
			12	
12		Инициализация целочисленных переменных m1,m2	13	
13		Вывод на экран значения строковой переменной name1, выделяя для переменной 15 позиция, сдвигая значение переменной влево	14	
14		Вывод на экран значения строковой переменной name2, выделяя для переменной 15 позиция, сдвигая значение переменной влево	15	
15		Переход на новую строку	16	
16		Инициализация целочисленной переменной i	17	
17		Присваивание i=0	18	
18	i < минимального из возвращаемых значений метода getLen объектов s1 и s2 класса Stack	Вызов метода getEl объекта s1 класса Stack с параметром m1	19	
			23	
19		Вывод на экран значения целочисленной	20	

		переменной m1, выделяя для переменной 15 позиция, сдвигая значение переменной вправо		
20		Вызов метода getEl объекта s2 класса Stack с параметром m2	21	
21		Вывод на экран значения целочисленной переменной m2, выделяя для переменной 15 позиция, сдвигая значение переменной вправо	22	
22	i != минимальному из возвращаемых значений метода getLen объектов s1 и s2 класса Stack -1	Переход на новую строку	18	
			18	
23	Возвращаемое значения метода getLen объекта s1 класса Stack > возвращаемого значения метода getLen объекта s2 класса Stack	Вызов метода getEl объекта s1 класса Stack с параметром m1	24	
			Ø	
24		Переход на новую строку	25	
25		Вывод на экран	Ø	

		значения целочисленной переменной m1, выделяя для переменной 15 позиция, сдвигая значение переменной вправо		
--	--	--	--	--

Класс объекта: Stack

Модификатор доступа: public

Метод: Stack

Функционал: Параметризированный конструктор с параметрами наименования и размера

Параметры: string name (Наименование), int n (Размер)

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 2.

Таблица 2. Алгоритм метода Stack класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1		Присваивание полю name значения переданной переменной name	2	
2		Присваивание полю n значения переданной переменной n	3	
3		Присваивание полю size_n значения переданной переменной 0	4	
4		Присваивание полю a адресасозданного массива размера n	Ø	

Класс объекта: Stack

Модификатор доступа: public

Метод: addEl

Функционал: Добавление элемента и возвращение логического признака успеха

Параметры: int ele

Возвращаемое значение: bool

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода addEl класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1	Возвращаемое значение метода getLen() = возвращемому значению метода getCurLen()Добавление элемента и возвращение логического признака успеха	Возвращение false	Ø	
		Присваивание элементу массива a под номером равным возвращаемого значения метода getCurLen() значения переменной ele	2	
2		Инкрементирование поля cur_n	3	
3		Возвращение true	Ø	

Класс объекта: Stack

Модификатор доступа: public

Метод: getEl

Функционал: Извлечение элемента и возвращение логического признака успеха

Параметры: int& m

Возвращаемое значение: bool

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода getEl класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1	Возвращаемое значение getCurLen()>0	Присваивание m значения элемента массива a под номером равным getCurLen()-1	2	
		Возвращение false	Ø	
2		Уменьшение значения поля cur_n на 1	3	
3		Возвращение true	Ø	

Класс объекта: Stack

Модификатор доступа: public

Метод: getName

Функционал: Получение имени стека

Параметры: Отсутствуют

Возвращаемое значение: string - наименование

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода getName класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1		Возвращение значения поля name	Ø	

Класс объекта: Stack

Модификатор доступа: public

Метод: getLen

Функционал: Получение длины стека

Параметры: Отсутствуют

Возвращаемое значение: int - длина стека

Алгоритм метода представлен в таблице 6.

Таблица 6. Алгоритм метода getLen класса Stack

№	Предикат	Действия	№ перехода	Комментарий
1		Возвращение поля n	Ø	

Класс объекта: Stack

Модификатор доступа: public

Метод: geCurtLen

Функционал: Получение текущего количества элементов в стеке

Параметры: Отсутствуют

Возвращаемое значение: int - текущее количество элементов в стеке

Алгоритм метода представлен в таблице 7.

Таблица 7. Алгоритм метода geCurtLen класса Stack

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

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

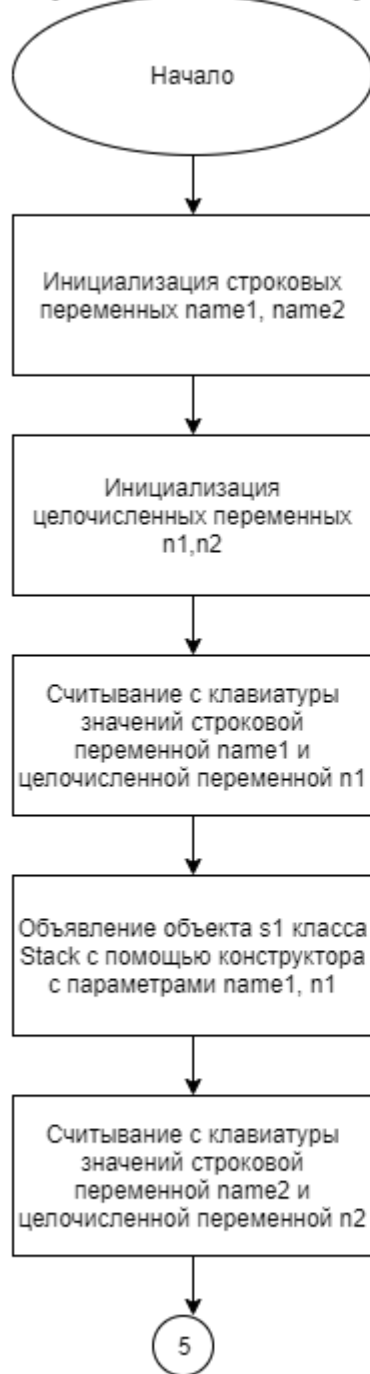


Рис. 1. Блок-схема алгоритма.



Рис. 2. Блок-схема алгоритма.

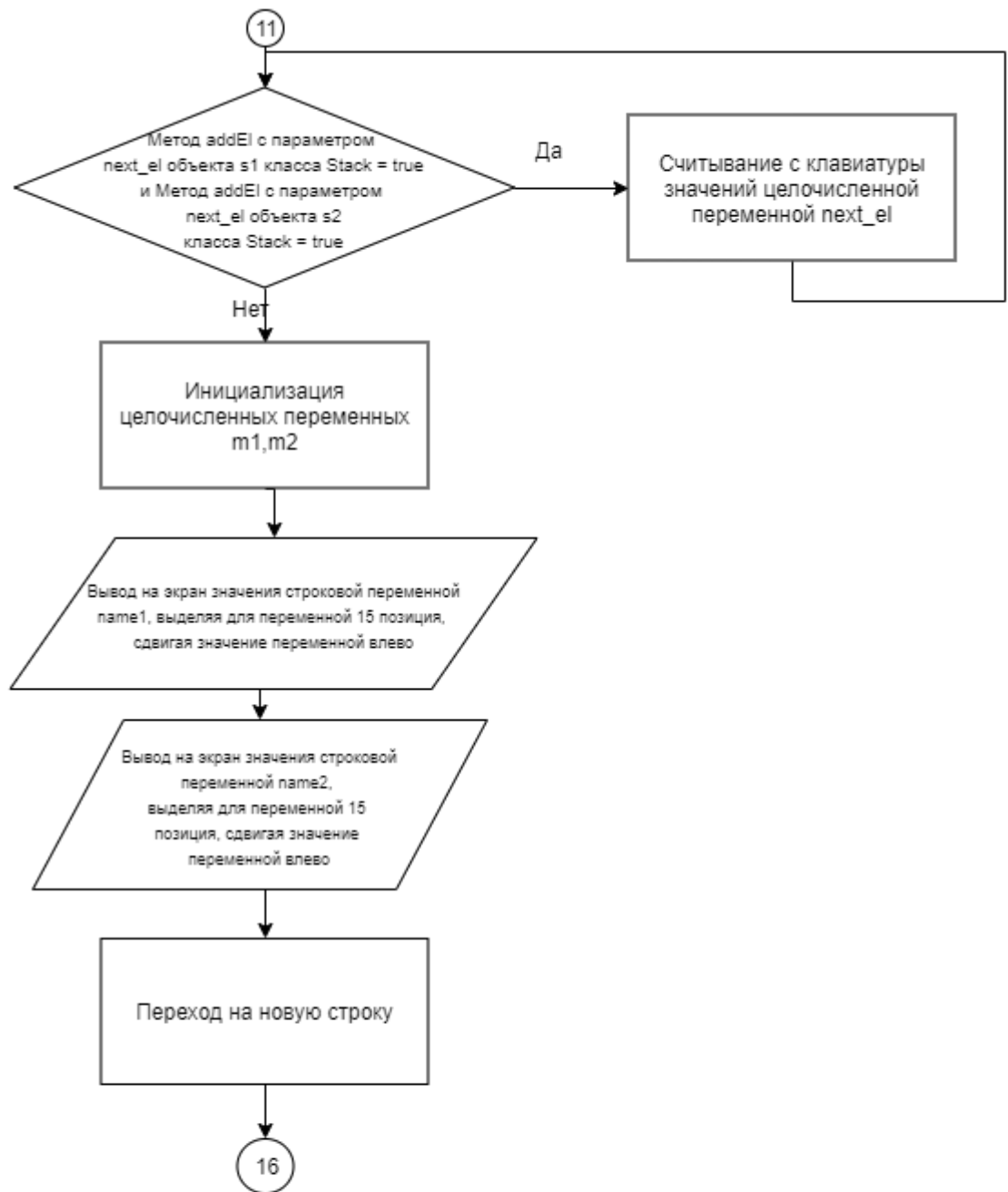


Рис. 3. Блок-схема алгоритма.

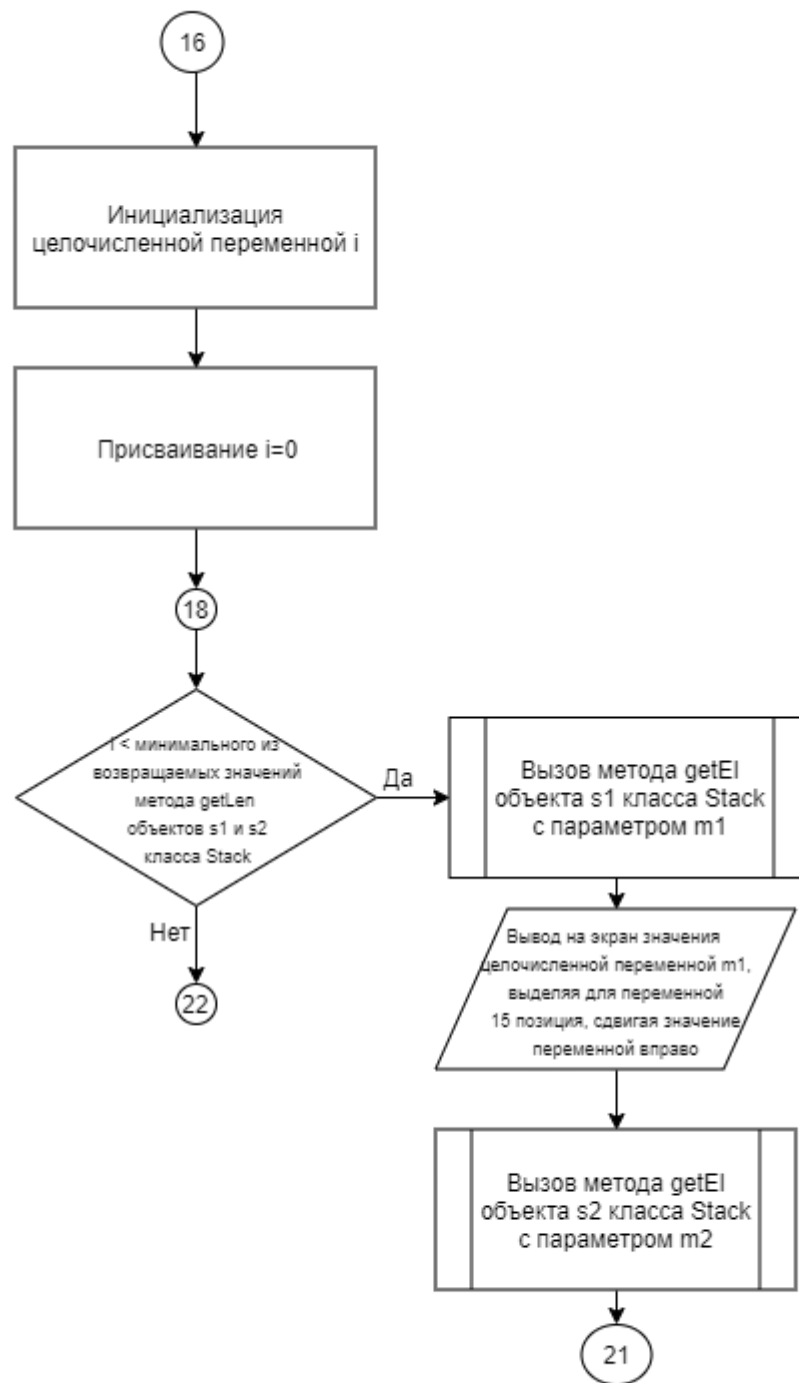


Рис. 4. Блок-схема алгоритма.

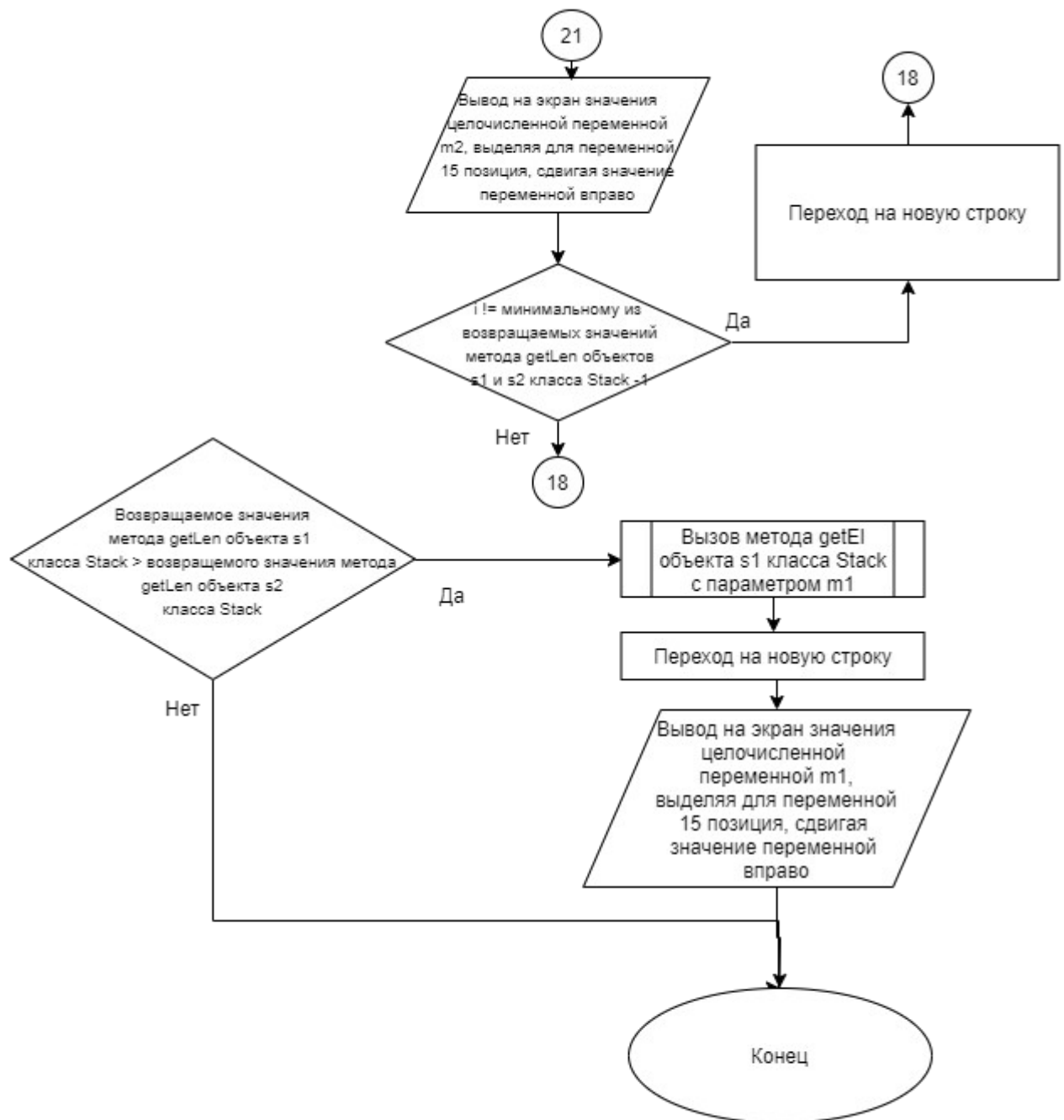


Рис. 5. Блок-схема алгоритма.

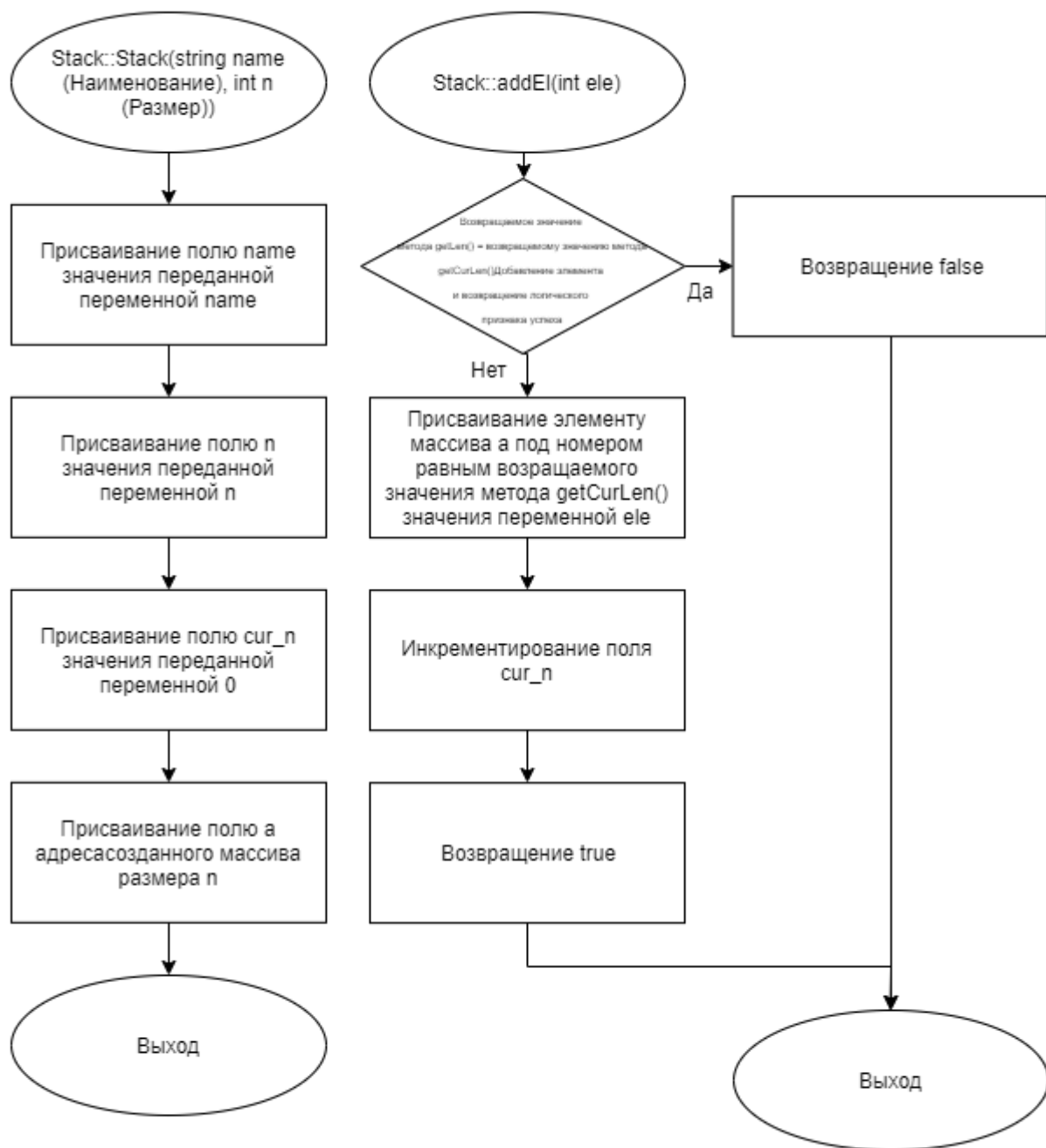


Рис. 6. Блок-схема алгоритма.

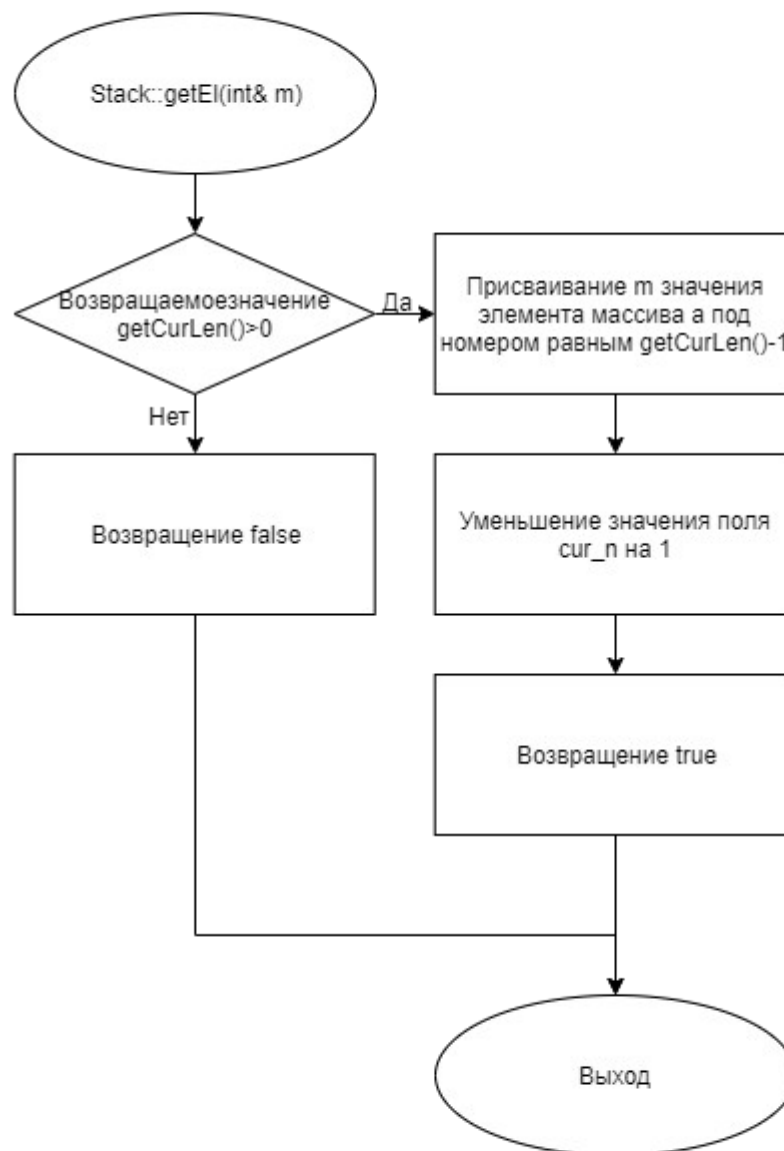


Рис. 7. Блок-схема алгоритма.

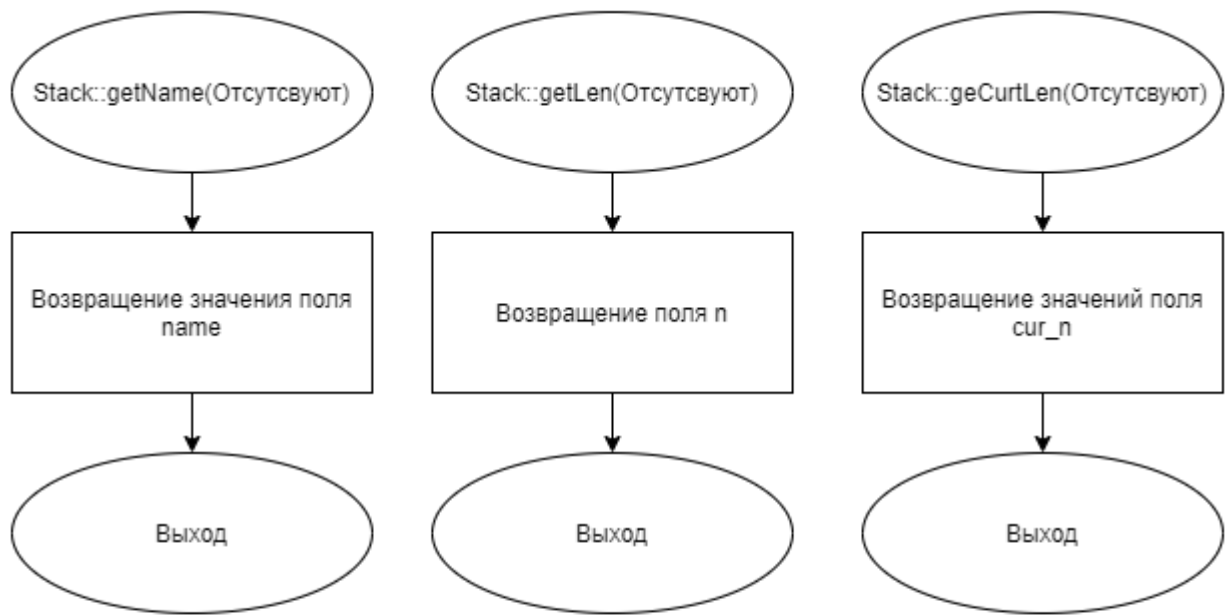


Рис. 8. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <iomanip>
#include "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) && 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;
    this->a = new int[n];
};
bool Stack::addEl(int el)
{
    if (getLen() == getCurLen()) {
        return false;
    }
    else
    {
        this->a[getCurLen()] = el;
        this->cur_n++;
        return true;
    }
};
bool Stack::getEl(int& m)
{
    if (getCurLen() > 0)
    {
        m = this->a[getCurLen() - 1];
        this->cur_n--;
        return true;
    }
    else return false;
};
int Stack::getCurLen()
{
    return this->cur_n;
};
string Stack::getName()
{
    return this->name;
};
int Stack::getLen() {
    return this->n;
};

```

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();
};
#endif
```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
qwe 5 dfgdv 6 789 1254 4546 0 745556 7854412 11	qwe 5 dfgdv 6 qwe dfgdv 745556 745556 0 0 4546 4546 1254 1254 789 789	qwe 5 dfgdv 6 qwe dfgdv 745556 745556 0 0 4546 4546 1254 1254 789 789
abc 3 abd 2 66 1337 29 0	abc 3 abd 2 abc abd 29 1337 1337 66 66	abc 3 abd 2 abc abd 29 1337 1337 66 66

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ)

1. Васильев А.Н. Объектно-ориентированное программирование на С++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. С++: 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-avrrora.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-avrrora.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).