Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....5
1.1 Описание входных данных.....6
1.2 Описание выходных данных.....6
2 МЕТОД РЕШЕНИЯ.....7
3 ОПИСАНИЕ АЛГОРИТМОВ.....8
3.1 Алгоритм конструктора класса steck.....8
3.2 Алгоритм деструктора класса steck.....8
3.3 Алгоритм метода back класса steck.....9
3.4 Алгоритм метода bod класса steck.....9
3.5 Алгоритм функции main.....10
3.6 Алгоритм метода getname класса steck.....10
3.7 Алгоритм метода getrazmer класса steck.....11
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....12
5 КОД ПРОГРАММЫ.....17
5.1 Файл main.cpp.....17
5.2 Файл steck.cpp.....18
5.3 Файл steck.h.....19
6 ТЕСТИРОВАНИЕ.....20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....21

1 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.

1.1 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.

1.2 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.

2 МЕТОД РЕШЕНИЯ

3 ОПИСАНИЕ АЛГОРИТМОВ

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

3.1 Алгоритм конструктора класса steck

Функционал: конструктор.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса steck

№	Предикат	Действия	№ перехода
1	this->razmer	this->razmer=razmer	2
			2
2		this->name=name	3
3		удаление steck	4
4		новый размер присваивается к steck	∅

3.2 Алгоритм деструктора класса steck

Функционал: деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса steck

№	Предикат	Действия	№ перехода
1		удаление steck	∅

3.3 Алгоритм метода back класса steck

Функционал: добавляем элемент.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода back класса steck

№	Предикат	Действия	№ перехода
1	a>razmer	steeck_one[a]=n	2
		return 0	∅
2		a++	3
3		return 1	∅

3.4 Алгоритм метода bod класса steck

Функционал: извлечение элемента.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода bod класса steck

№	Предикат	Действия	№ перехода
1	a<0	n=steck_one[a]	2
		return 0	∅
2		a--	3
3		return 1	∅

3.5 Алгоритм функции main

Функционал: основная функция.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		оъявление razmer_one and name_one ввод razmer_one, name_one объявление steck_one класса steck объявлеие razmer_two, name_two ввод razmer_two, name_two объявление steck2 класса steck вывод имя и размер двух стэков объявление b	2
2	steck_one.back(b) steck2.(b)	ввод b	3
			∅
3		вывод b	∅

3.6 Алгоритм метода getname класса steck

Функционал: получение названия.

Параметры: нет.

Возвращаемое значение: название.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода getname класса steck

№	Предикат	Действия	№ перехода
1		return name	∅

3.7 Алгоритм метода getrazmer класса steck

Функционал: получение размера.

Параметры: нет.

Возвращаемое значение: размер.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода getrazmer класса steck

№	Предикат	Действия	№ перехода
1		return razmer	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

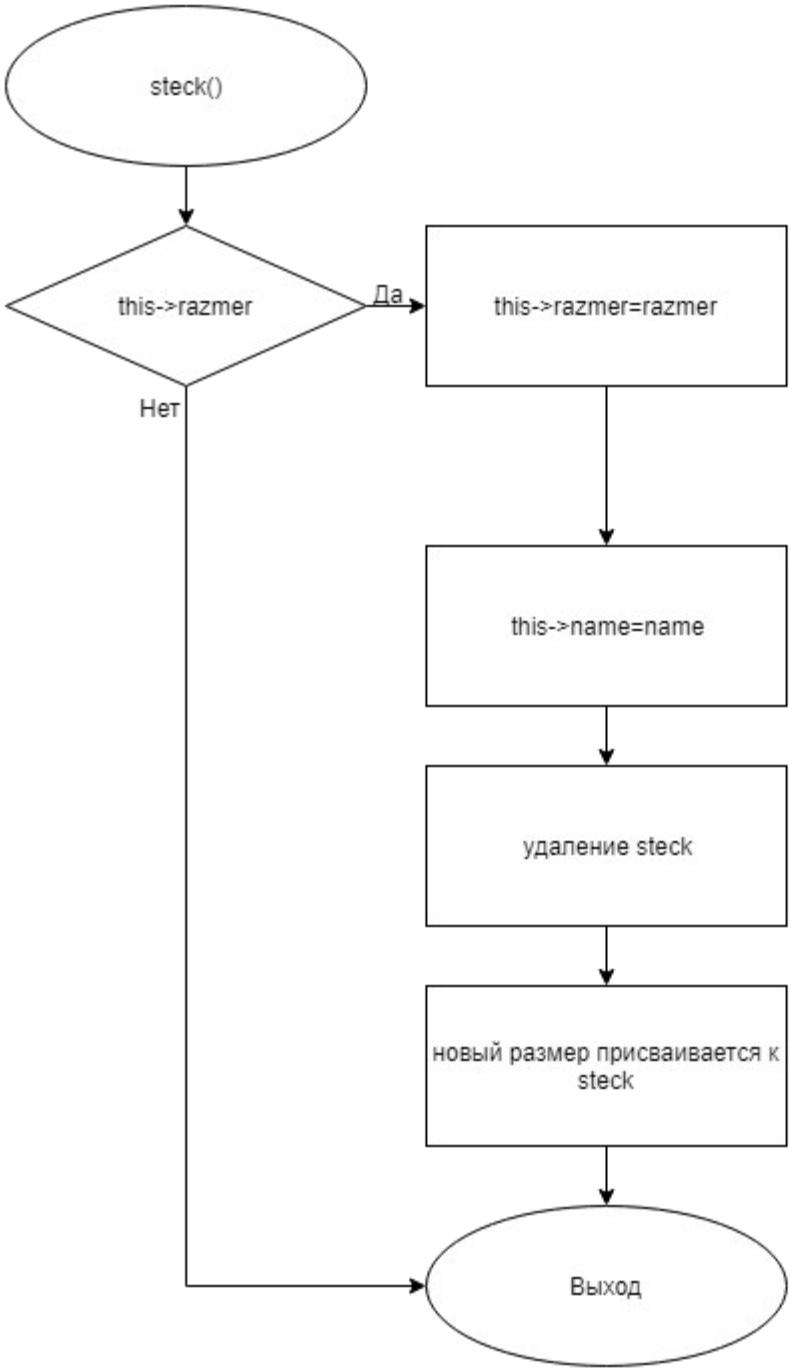

Рисунок 1 – Блок-схема алгоритма

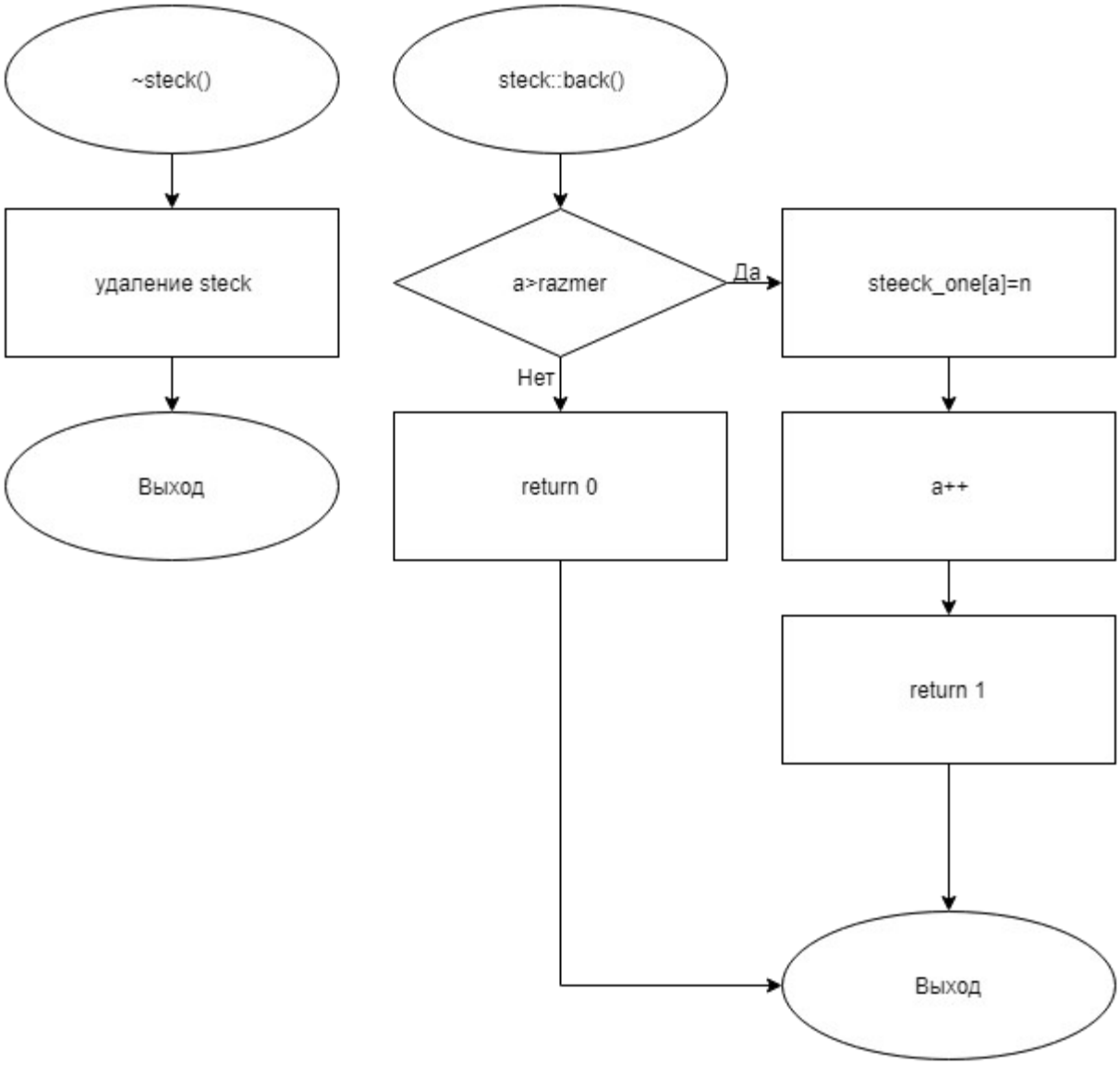

Рисунок 2 – Блок-схема алгоритма

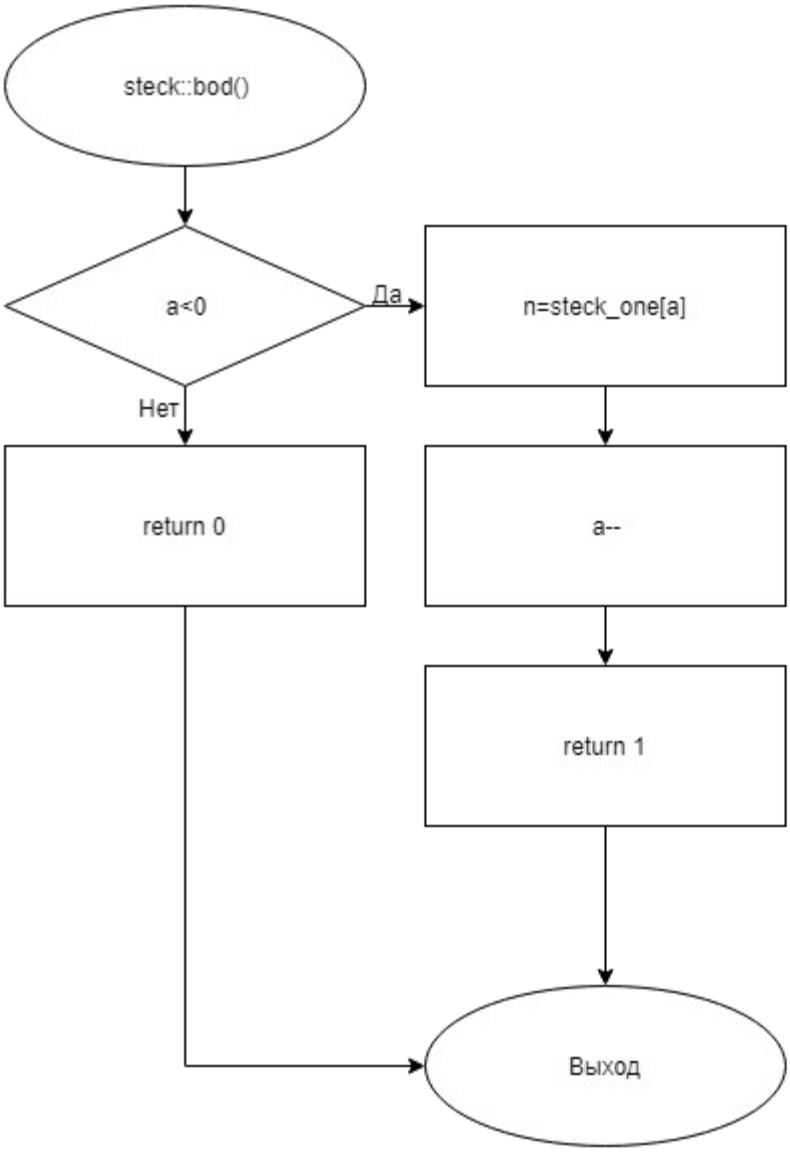

Рисунок 3 – Блок-схема алгоритма

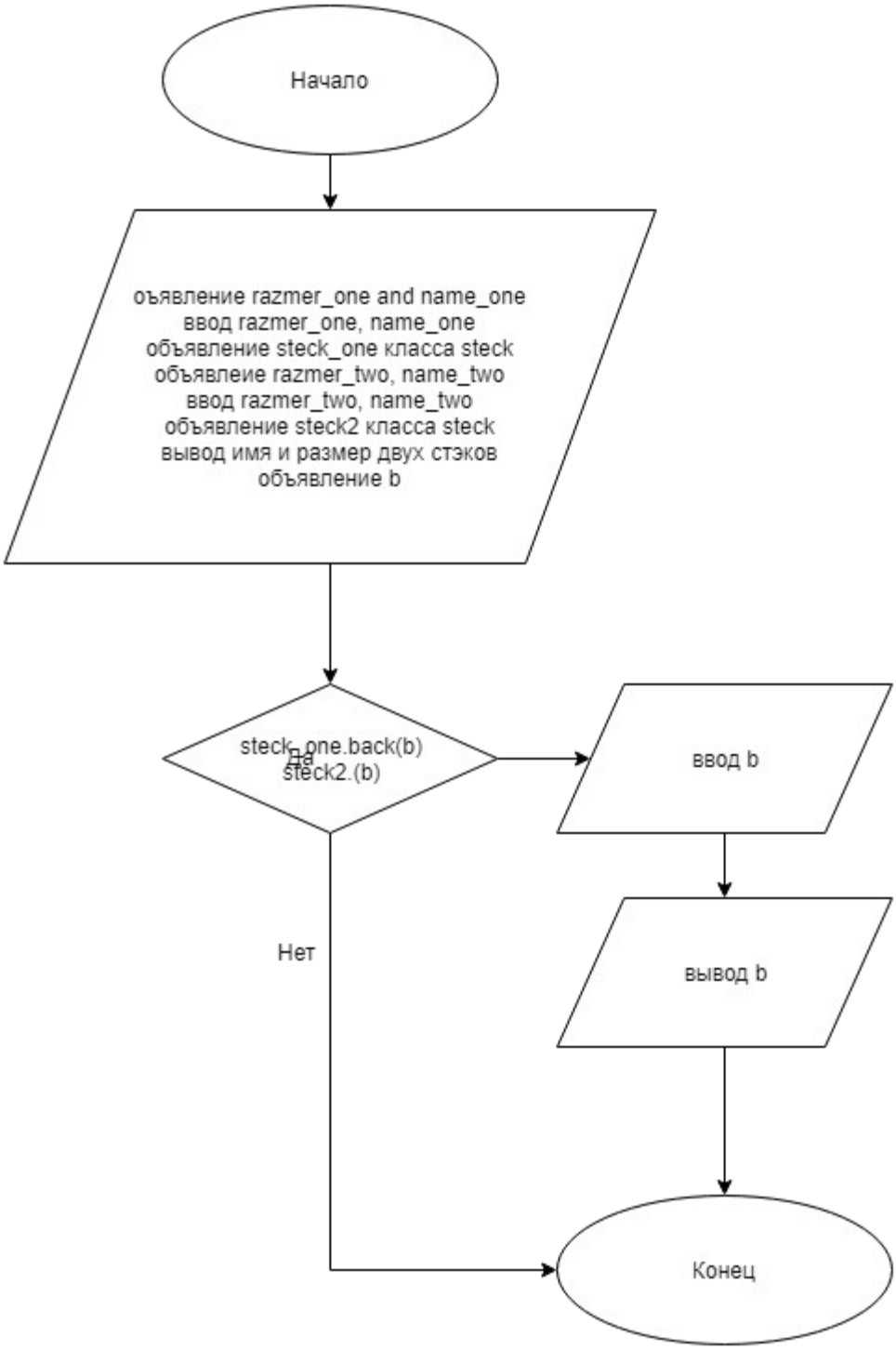

Рисунок 4 – Блок-схема алгоритма

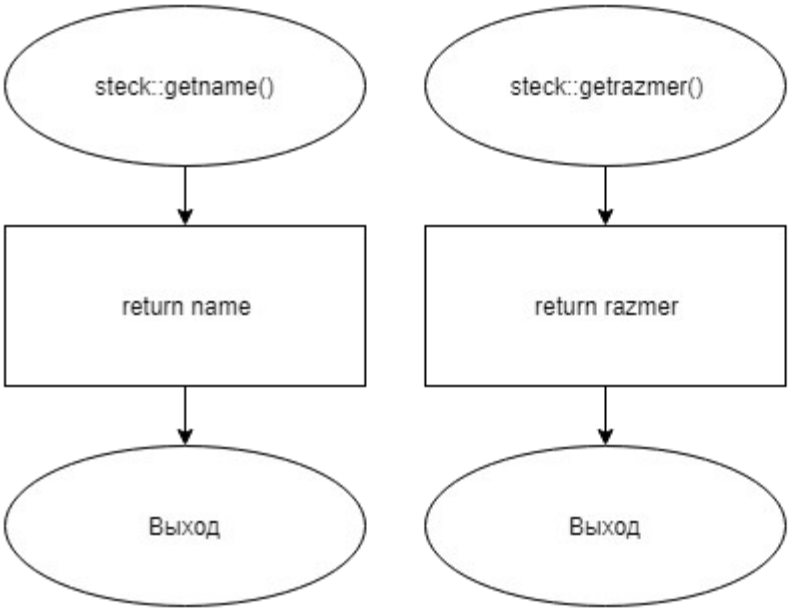

Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iomanip>
#include "steck.h"
#include <iostream>
#include <string>
using namespace std;

int main()
{
   int razmer_one;
   string name_one;
   cin >> name_one >> razmer_one;
   steck steck_one(name_one, razmer_one);

   int razmer_two;
   string name_two;
   cin >> name_two >> razmer_two;
   steck steck2(name_two, razmer_two);

   cout << steck_one.getname() << " " << steck_one.getrazmer()<<"\n";
   cout << steck2.getname() << " " << steck2.getrazmer()<< "\n";
   cout << setw(15) << left << steck_one.getname();
   cout << setw(15) << left << steck2.getname() << endl;
   int b = 0;
   cin >> b;

   while (steck_one.back(b) && steck2.back(b))
   {
      cin >> b;
   }

   while (1)
   {
      if (!steck_one.bod(b))
      {
         break;
      }
      else
```

```
            {
               cout << setw(15) << right << b;
               if (steck2.bod(b))
               {
                  cout << setw(15) << right << b << endl;
               }
            }
         }
         return(0);
      }
```

5.2 Файл steck.cpp

Листинг 2 – steck.cpp

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

      steck::steck(string name, int razmer)
      {
         this->razmer = razmer;
         this->name = name;
         this->steck_one = new int[razmer];
         a = 0;
      }

      steck::~steck()
      {
         delete[]steck_one;
      }

      bool steck::back(int n)
      {
         if (a >= razmer)
         {
            return 0;
         }
         steck_one[a] = n;
         a++;
         return 1;
      }
      bool steck::bod(int& n)
      {
         if(a<=0)
         {
            return 0;
         }
         a--;
```

```
   n = steck_one[a];
   return 1;
}

string steck::getname()
{
   return name;
}

int steck::getrazmer()
{
   return razmer;
}
```

5.3 Файл steck.h

Листинг 3 – steck.h

```
#ifndef __STECK__H
#define __STECK__H
#include <iostream>
#include <string>
using namespace std;

class steck
{
private:
   int razmer;
   string name;
   int* steck_one;
   int a = 0;
public:
   steck(string name, int razmer);
   ~steck();
   bool back(int n);
   bool bod(int& n);
   string getname();
   int getrazmer();
};


#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 8.

Таблица 8 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**
A 2 B 3 12345	A 2 B 3 A B 12345 12345 12345 12345	A 2 B 3 A B 12345 12345 12345 12345

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.

2. Методическое пособие студента для выполнения 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).

3. 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).

4. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.

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).