Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Описать класс cl_parent объекта, в котором следующий состав элементов:

В закрытом разделе:

− одно свойство целого типа;

− метод, с одним целочисленным параметром. который меняет значение свойства в закрытом разделе на удвоенное значение параметра.

В открытом разделе:

− одно свойство целого типа;

− параметризированный конструктор, с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе. Значение закрытого свойства меняется посредством вызова метода из закрытого раздела;

− метод с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе. Значение закрытого свойства меняется посредством вызова метода из закрытого раздела;

− метод, который выводит на экран значение обоих свойств. Сперва значение закрытого свойства, потом значение открытого свойства.

Назовем объект данного класса родительским. Соответственно его класс родительским классом.

На базе родительского объекта сконструируем производный объект. Производный объект должен сохранить открытый доступ к открытым элементам родительского класса. Он должен иметь следующие собственные элементы:

В закрытом разделе:

− одно свойство целого типа, наименование которого совпадает с наименование закрытого свойства родительского объекта;

В открытом разделе:

− одно свойство целого типа, наименование которого совпадает с наименование открытого свойства родительского объекта;

− параметризированный конструктор, с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе;

− метод с двумя целочисленными параметрами, который устанавливает значения свойств в закрытом и открытом разделе. Наименование метода совпадает с наименованием аналогичного метода родительского объекта;

− метод, который выводит на экран значение обоих свойств. Сперва значение закрытого свойства, потом значение открытого свойства. Наименование метода совпадает с наименованием аналогичного метода родительского объекта.

Разработать производный класс используя класс cl_parent в качестве родительского.

В основной функции реализовать алгоритм:

1. Ввод значения двух целочисленных переменных.

2. Создать объект производного класса используя целочисленных переменных в конструкторе в качестве аргументов в последовательности, как им были присвоены значения. Первый аргумент содержит значение для свойства закрытого раздела/,/ второй для свойства открытого раздела.

3. Вывод значений свойств родительского объекта.

4. Вывод значений свойств производного объекта.

5. Если исходное значение закрытого свойства больше нуля, то:

5.1. Переопределить значения свойств производного объекта, увеличив на единицу введенные исходные значения.

5.2. Переопределить значения свойств родительского объекта, уменьшив на единицу введенные исходные значения.

5.3. Вывод значений свойств производного объекта.

5.4. Вывод значений свойств родительского объекта.

6. Иначе:

6.1. Переопределить значения свойств родительского объекта, увеличив на единицу введенные исходные значения.

6.2. Переопределить значения свойств производного объекта, уменьшив на единицу введенные исходные значения.

6.3. Вывод значений свойств родительского объекта.

6.4. Вывод значений свойств производного объекта.

1.1 Описание входных данных

В первой строке:

«Целое число» «Целое число»

Пример ввода:

```
8 5
```

1.2 Описание выходных данных

Начиная с первой строки:

«Целое число» «Целое число»
«Целое число» «Целое число»
«Целое число» «Целое число»
«Целое число» «Целое число»

Пример вывода:

```
16   5
8   5
9   6
14   4
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj класса cl_obj предназначен для Для решения задачи;
- cin/cout - объекты стандартного потока ввода/вывода;
- if..else - условный оператор.

Класс cl_parent:

- свойства/поля:
    - поле Закрытое свойство:
        - наименование — cl;
        - тип — int;
        - модификатор доступа — private;
    - поле Открытое свойство:
        - наименование — op;
        - тип — int;
        - модификатор доступа — public;
- функционал:
    - метод m1 — Закрытый метод изменения свойств;
    - метод m2 — Открытый метод изменения свойств;
    - метод cl_parent — Параметризированный конструктор;
    - метод output — Метод вывода свойств на экран.

Класс cl_obj:

- свойства/поля:
    - поле Закрытое свойство:
        - наименование — cl;
        - тип — ;
        - модификатор доступа — private;

    - поле Открытое свойство:
        - наименование — op;
        - тип — ;
        - модификатор доступа — private;
- функционал:
    - метод m2 — Открытый метод изменения свойств;
    - метод cl_obj — Параметризированный конструктор;
    - метод output — Метод вывода свойств на экран.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl_parent			Родительский класс для решения задачи	
		cl_obj	public		2
2	cl_obj			Наследный класс для решения задачи	

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 метода m1 класса cl_parent

Функционал: Закрытый метод изменения свойств.

Параметры: int a - число для определения.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода m1 класса cl_parent

№	Предикат	Действия	№ перехода
1		cl = a * 2	∅

3.2 Алгоритм метода m2 класса cl_parent

Функционал: Открытый метод изменения свойств.

Параметры: int cl, int op - параметры для установки в свойства.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода m2 класса cl_parent

№	Предикат	Действия	№ перехода
1		Вызов метода m1 с передачей cl как параметра	2
2		Установка свойства op равному параметру op	∅

3.3 Алгоритм конструктора класса cl_parent

Функционал: Параметризированный конструктор.

Параметры: int cl, int op - параметры для установки в свойства.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl_parent

№	Предикат	Действия	№ перехода
1		Вызов метода m1 с передачей cl как параметра	2
2		Установка свойства op равному параметру op	∅

3.4 Алгоритм метода output класса cl_parent

Функционал: Метод вывода свойств на экран.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода output класса cl_parent

№	Предикат	Действия	№ перехода
1		Вывод cl и op	∅

3.5 Алгоритм метода m2 класса cl_obj

Функционал: Открытый метод изменения свойств.

Параметры: int cl, int op - параметры для установки в свойства.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода m2 класса cl_obj

№	Предикат	Действия	№ перехода
1		Установка свойства cl равному параметру cl	2
2		Установка свойства op равному параметру op	∅

3.6 Алгоритм конструктора класса cl_obj

Функционал: Параметризированный конструктор.

Параметры: int cl, int op - параметры для установки в свойства.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса cl_obj

№	Предикат	Действия	№ перехода
1		Установка свойства cl равному параметру cl	2
2		Установка свойства op равному параметру op	∅

3.7 Алгоритм метода output класса cl_obj

Функционал: Метод вывода свойств на экран.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода output класса cl_obj

№	Предикат	Действия	№ перехода
1		Вывод cl и op	∅

3.8 Алгоритм функции main

Функционал: Главная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление переменных a и b типа int	2
2		Ввод значений a и b	3
3		Инициализация объекта obj класса cl_obj с передачей параметров a и b конструктору	4
4		Вызов метода output у родительского объекта obj	5
5		Вызов метода output у объекта obj	6
6	a > 0		7
			11
7		Вызов метода m2 у объекта obj с передачей параметров a + 1 и b + 1	8
8		Вызов метода m2 у родительского объекта obj с передачей параметров a - 1 и b - 1	9
9		Вызов метода output у объекта obj	10
10		Вызов метода output у родительского объекта obj	15
11		Вызов метода m2 у родительского объекта obj с передачей параметров a + 1 и b + 1	12
12		Вызов метода m2 у объекта obj с передачей параметров a - 1 и b - 1	13
13		Вызов метода output у родительского объекта obj	14
14		Вызов метода output у объекта obj	15
15		Возврат значения 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

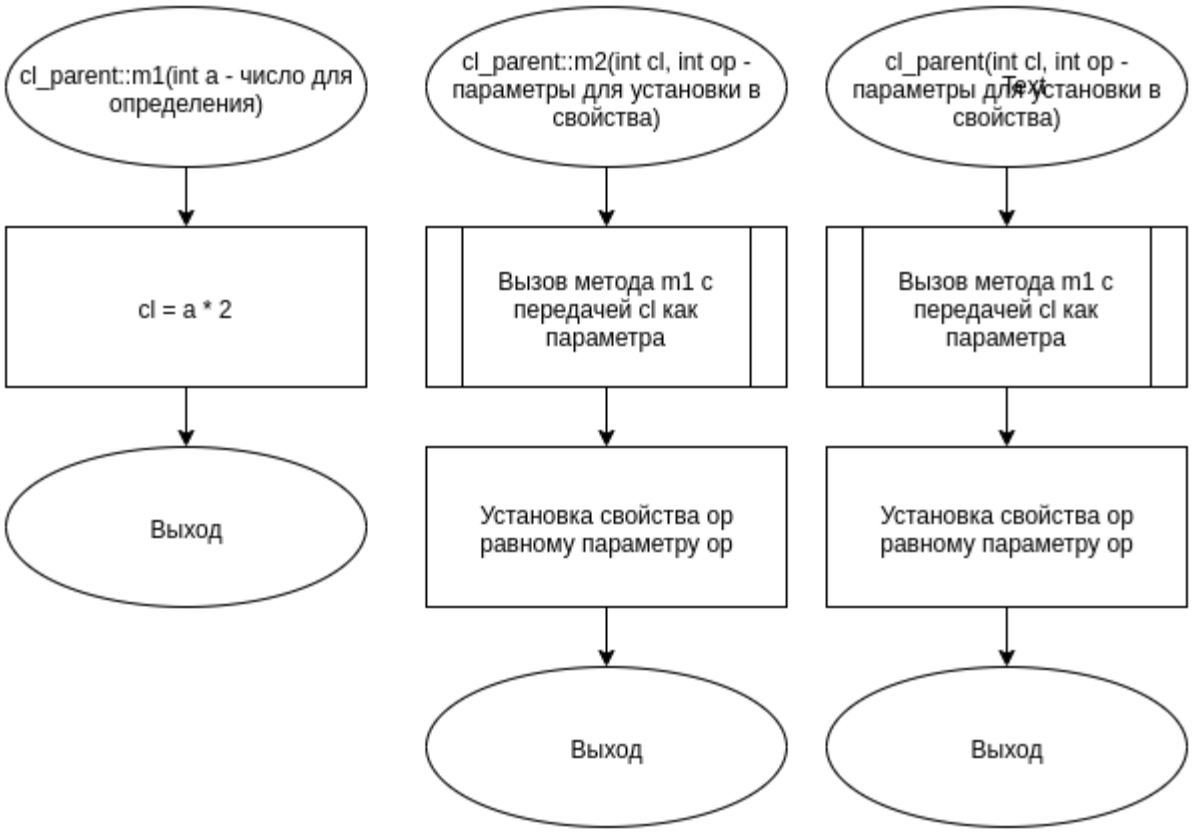

Рисунок 1 – Блок-схема алгоритма

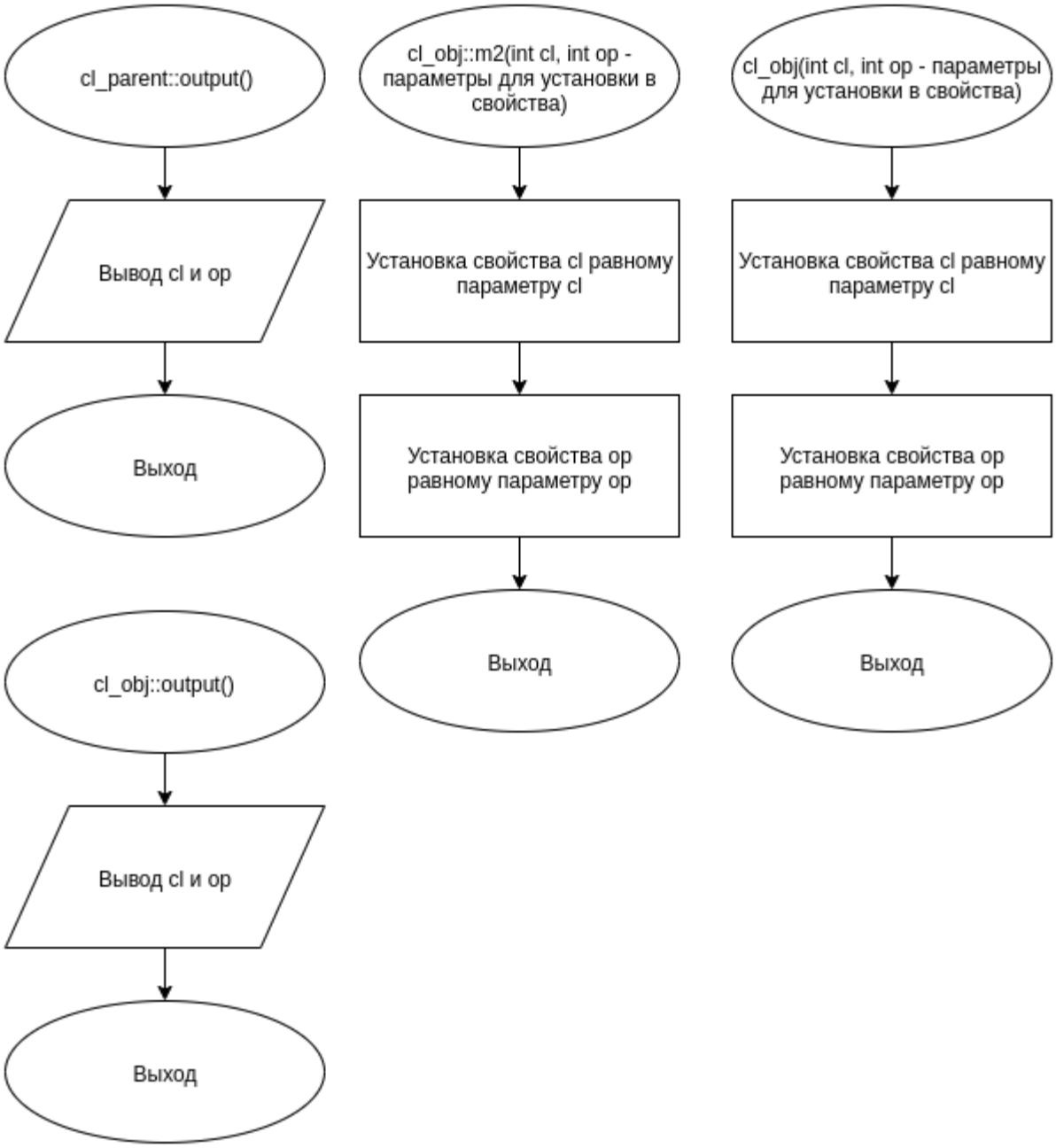

Рисунок 2 – Блок-схема алгоритма

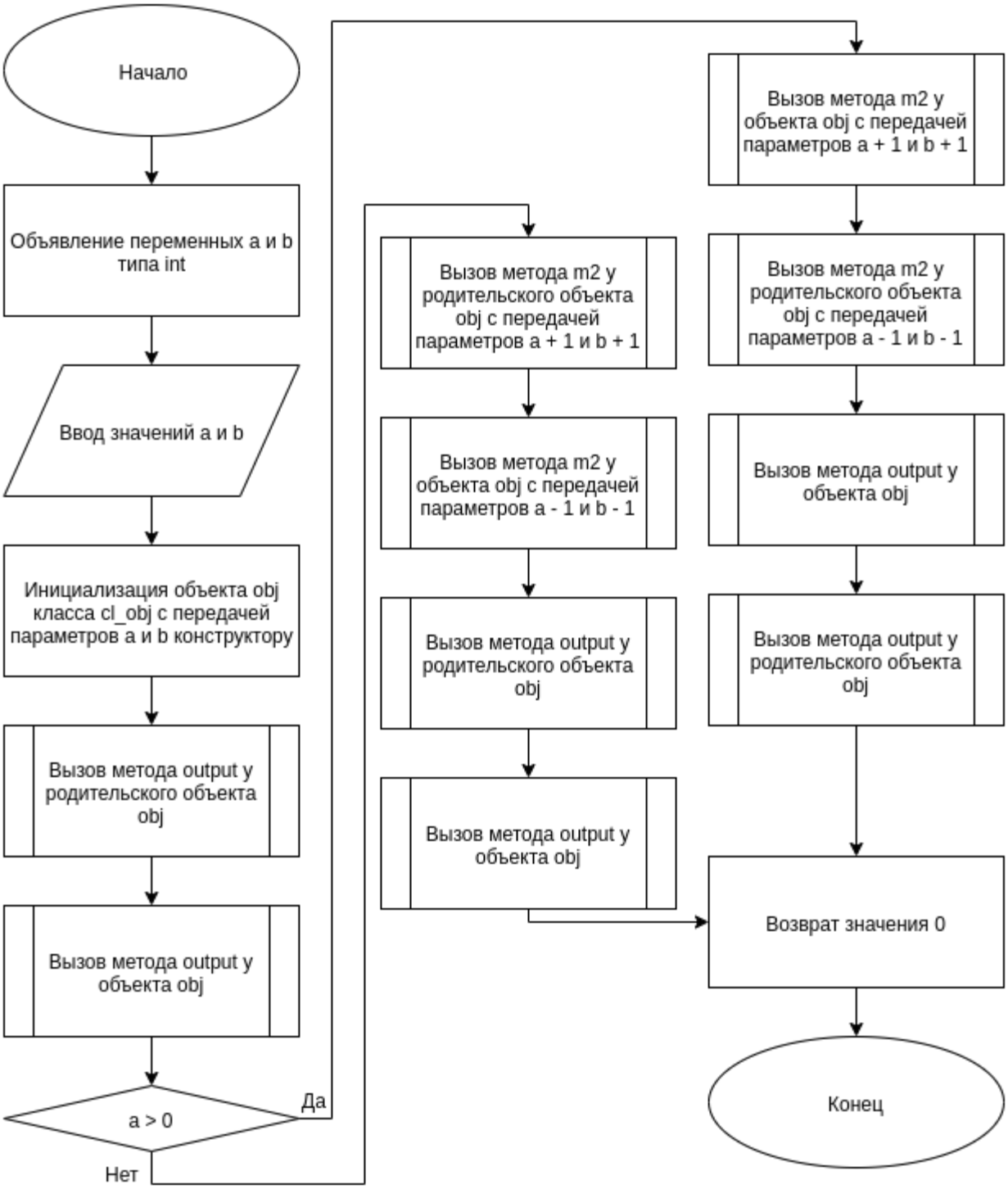

Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl_obj.cpp

Листинг 1 – cl_obj.cpp

```
#include "cl_obj.h"


cl_obj::cl_obj(int cl, int op) : cl_parent(cl, op)
{
   this->cl = cl;
   this->op = op;
}

void cl_obj::m2(int cl, int op)
{
   this->cl = cl;
   this->op = op;
}

void cl_obj::output()
{
   cout << cl << "    " << op;
}
```

5.2 Файл cl_obj.h

Листинг 2 – cl_obj.h

```
#ifndef __CL_OBJ__H
#define __CL_OBJ__Hh
#include "cl_parent.h"

class cl_obj : public cl_parent
{
private:
   int cl;
public:
   int op;
```

```
   cl_obj(int cl, int op);
   void m2(int cl, int op);
   void output();
};

#endif
```

5.3 Файл cl_parent.cpp

Листинг 3 – cl_parent.cpp

```
#include "cl_parent.h"


void cl_parent::m1(int a)
{
   cl = a * 2;
}

cl_parent::cl_parent(int cl, int op)
{
   m1(cl);
   this->op = op;
}

void cl_parent::m2(int cl, int op)
{
   m1(cl);
   this->op = op;
}


void cl_parent::output()
{
   cout << cl << "    " << op;
}
```

5.4 Файл cl_parent.h

Листинг 4 – cl_parent.h

```
#ifndef __CL_PARENT__H
#define __CL_PARENT__H

#include <iostream>
```

```
using namespace std;

class cl_parent
{
private:
   int cl;
   void m1(int a);
public:
   int op;
   cl_parent(int cl, int op);
   void m2(int cl, int op);
   void output();
};

#endif
```

5.5 Файл main.cpp

Листинг 5 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "cl_obj.h"


int main()
{
   int a, b;
   cin >> a >> b;
   cl_obj obj(a, b);

   obj.cl_parent::output();
   cout << endl;
   obj.output();
   cout << endl;

   if(a > 0){
      obj.m2(a + 1, b + 1);
      obj.cl_parent::m2(a - 1, b - 1);
      obj.output();
      cout << endl;
      obj.cl_parent::output();

   }
   else
   {
      obj.cl_parent::m2(a + 1, b + 1);
      obj.m2(a - 1, b - 1);
      obj.cl_parent::output();
      cout << endl;
```

```
            obj.output();
        }

        return(0);
    }
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 10.

Таблица 10 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**
8 5	16 5 8 5 9 6 14 4	16 5 8 5 9 6 14 4

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).