

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	6
1.1 Описание входных данных.....	7
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	8
3 ОПИСАНИЕ АЛГОРИТМОВ.....	11
3.1 Алгоритм конструктора класса cl1.....	11
3.2 Алгоритм метода Output класса cl1.....	11
3.3 Алгоритм конструктора класса cl2.....	12
3.4 Алгоритм метода Output класса cl2.....	12
3.5 Алгоритм конструктора класса cl3.....	13
3.6 Алгоритм метода Output класса cl3.....	13
3.7 Алгоритм конструктора класса cl4.....	13
3.8 Алгоритм метода Output класса cl4.....	14
3.9 Алгоритм функции main.....	14
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	16
5 КОД ПРОГРАММЫ.....	18
5.1 Файл cl1.cpp.....	18
5.2 Файл cl1.h.....	18
5.3 Файл cl2.cpp.....	19
5.4 Файл cl2.h.....	19
5.5 Файл cl3.cpp.....	20
5.6 Файл cl3.h.....	20
5.7 Файл cl4.cpp.....	21
5.8 Файл cl4.h.....	21
5.9 Файл main.cpp.....	22
6 ТЕСТИРОВАНИЕ.....	23

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	24
---------------------------------------	----

1 ПОСТАНОВКА ЗАДАЧИ

Иерархия наследования

Описать четыре класса которые последовательно наследуют друг друга, последовательными номерами классов 1,2,3,4.

Реализовать программу, в которой использовать единственный указатель на объект базового класса (номер класса 1).

Наследственность реализовать так, что можно было вызывать методы, принадлежащие объекту конкретного класса, только через объект данного класса.

В закрытом разделе каждого класса определены два свойства: строкового типа для наименования объекта и целого типа для значения определенного целочисленного выражения.

Описание каждого класса содержит один параметризованный конструктор с строковым и целочисленным параметром.

В реализации каждого конструктора объекта определяются значения закрытых свойств:

- Наименование объекта по шаблону: «значение строкового параметра»_«номер класса»;
- Целочисленного свойства значением выражения возведения в степень номера класса целочисленного значения параметра конструктора.

Еще в описании каждого класса определен метод с одинаковым наименованием для всех классов, реализующий вывод значений закрытых свойств класса.

В основной функции реализовать алгоритм:

1. Вводится идентификатор и натуральное число от 2 до 10.
2. Создать объект класса 4, используя параметризованный конструктор,

которому в качестве аргументов передаются введенный идентификатор и натуральное число.

3. Построчно, для всех объектов согласно наследственности, от объекта базового (класс 1) до производного объекта (класса 4) вывести наименование объекта класса и значение целочисленного свойства.

1.1 Описание входных данных

Первая строка:

«идентификатор» «натуральное число»

Пример ввода:

Object 2

1.2 Описание выходных данных

Построчно (четыре строки):

«идентификатор»_«номер класса» «значение целочисленного свойства»

Разделитель - 1 пробел.

Пример вывода:

Object_1 2
Object_2 4
Object_3 8
Object_4 16

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект Object класса cl4 предназначен для ;
- функция main для главная функция программы;
- Объект стандартного потока ввода cin с клавиатуры;
- Объект стандартного потока вывода cout на экран;
- библиотека iostream;
- Пространство имён std.

Класс cl1:

- свойства/поля:
 - поле Хранение названия объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
 - поле Хранение значения объекта:
 - наименование — val;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод cl1 — параметризованный конструктор. Установка имени и целочисленного значения объекта;
 - метод Output — вывод имени объекта и его значения.

Класс cl2:

- свойства/поля:
 - поле хранение названия объекта:
 - наименование — name;

- тип — string;
- модификатор доступа — private;
- поле хранение значения объекта:
 - наименование — val;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод cl2 — установка имени и целочисленного значения объекта;
 - метод Output — Вывод имени и значения объекта на экран.

Класс cl3:

- свойства/поля:
 - поле хранение названия объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
 - поле хранение значения объекта:
 - наименование — val;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод cl3 — установка имени и целочисленного значения объекта;
 - метод Output — Вывод имени и значения объекта на экран.

Класс cl4:

- свойства/поля:
 - поле хранение названия объекта:
 - наименование — name;
 - тип — string;

- модификатор доступа — private;
- о поле хранения значения объекта:
 - наименование — val;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - о метод cl4 — установка имени и целочисленного значения объекта;
 - о метод Output — Вывод имени и значения объекта на экран.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl1				
		cl2	private		2
2	cl2				
		cl3	private		3
3	cl3				
		cl4	private		4
4	cl4				

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 cl1

Функционал: параметризованный конструктор. Установка имени и целочисленного значения объекта.

Параметры: Переменная строкового типа name, целочисленная переменная val.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		Присвоение скрытому полю name значения переданного аргумента name и добавления к его значению "_1"	2
2		Присвоение скрытому полю val значения переданного аргумента val	Ø

3.2 Алгоритм метода Output класса cl1

Функционал: вывод имени объекта и его значения.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Output класса cl1

№	Предикат	Действия	№ перехода
1		Вывод: <<значение скрытого поля name>>," ",<<значение скрытого поля val>>	Ø

3.3 Алгоритм конструктора класса cl2

Функционал: установка имени и целочисленного значения объекта.

Параметры: Переменная строкового типа name, целочисленная переменная val.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl2

№	Предикат	Действия	№ перехода
1		Присвоение скрытому полю name значения переданного аргумента name и добавления к его значению "_2"	2
2		Присвоение скрытому полю val значения произведения переданного аргумента самого на себя val * val	Ø

3.4 Алгоритм метода Output класса cl2

Функционал: Вывод имени и значения объекта на экран.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Output класса cl2

№	Предикат	Действия	№ перехода
1		Вывод: <<значение скрытого поля name>>," ",<<значение скрытого поля val>>	Ø

3.5 Алгоритм конструктора класса cl3

Функционал: установка имени и целочисленного значения объекта.

Параметры: Переменная строкового типа name, целочисленная переменная val.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса cl3

№	Предикат	Действия	№ перехода
1		Присвоение скрытому полю name значения переданного аргумента name и добавления к его значению "_2"	2
2		Присвоение скрытому полю val значения произведения переданного аргумента самого на себя val * val * val	Ø

3.6 Алгоритм метода Output класса cl3

Функционал: Вывод имени и значения объекта на экран.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода Output класса cl3

№	Предикат	Действия	№ перехода
1		Вывод: <<значение скрытого поля name>>," ",<<значение скрытого поля val>>	Ø

3.7 Алгоритм конструктора класса cl4

Функционал: установка имени и целочисленного значения объекта.

Параметры: Переменная строкового типа name, целочисленная переменная

val.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса *cl4*

№	Предикат	Действия	№ перехода
1		Присвоение скрытому полю name значения переданного аргумента name и добавления к его значению "_2"	2
2		Присвоение скрытому полю val значения произведения переданного аргумента самого на себя val * val * val * val	Ø

3.8 Алгоритм метода Output класса *cl4*

Функционал: Вывод имени и значения объекта на экран.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода Output класса *cl4*

№	Предикат	Действия	№ перехода
1		Вывод: <<значение скрытого поля name>>," ",<<значение скрытого поля val>>	Ø

3.9 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление строковой переменной <i>name</i>	2
2		объявление целочисленной переменной <i>val</i>	3
3		Ввод значения переменной <i>name</i>	4
4		Ввод значения переменной <i>val</i>	5
5		Инициализация указателя <i>Object</i> на объект класса <i>cl1</i> адресом нового объекта класса <i>cl4</i> посредством оператора <i>new</i> , с передачей конструктору аргументов <i>name</i> и <i>val</i> в качестве параметров, с приведением к указателю на объект класса <i>cl1</i>	6
6		Вызов метода <i>Output()</i> объекта <i>Object</i>	7
7		Переход на новую строку	8
8		Вызов метода <i>Output()</i> объекта <i>Object</i> , приведённого к указателю на объект класса <i>cl2</i>	9
9		Переход на новую строку	10
10		Вызов метода <i>Output()</i> объекта <i>Object</i> , приведённого к указателю на объект класса <i>cl3</i>	11
11		Переход на новую строку	12
12		Вызов метода <i>Output()</i> объекта <i>Object</i> , приведённого к указателю на объект класса <i>cl4</i>	13
13		Освобождение памяти, выделенной под объект <i>Object</i> , посредством оператора <i>delete</i>	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

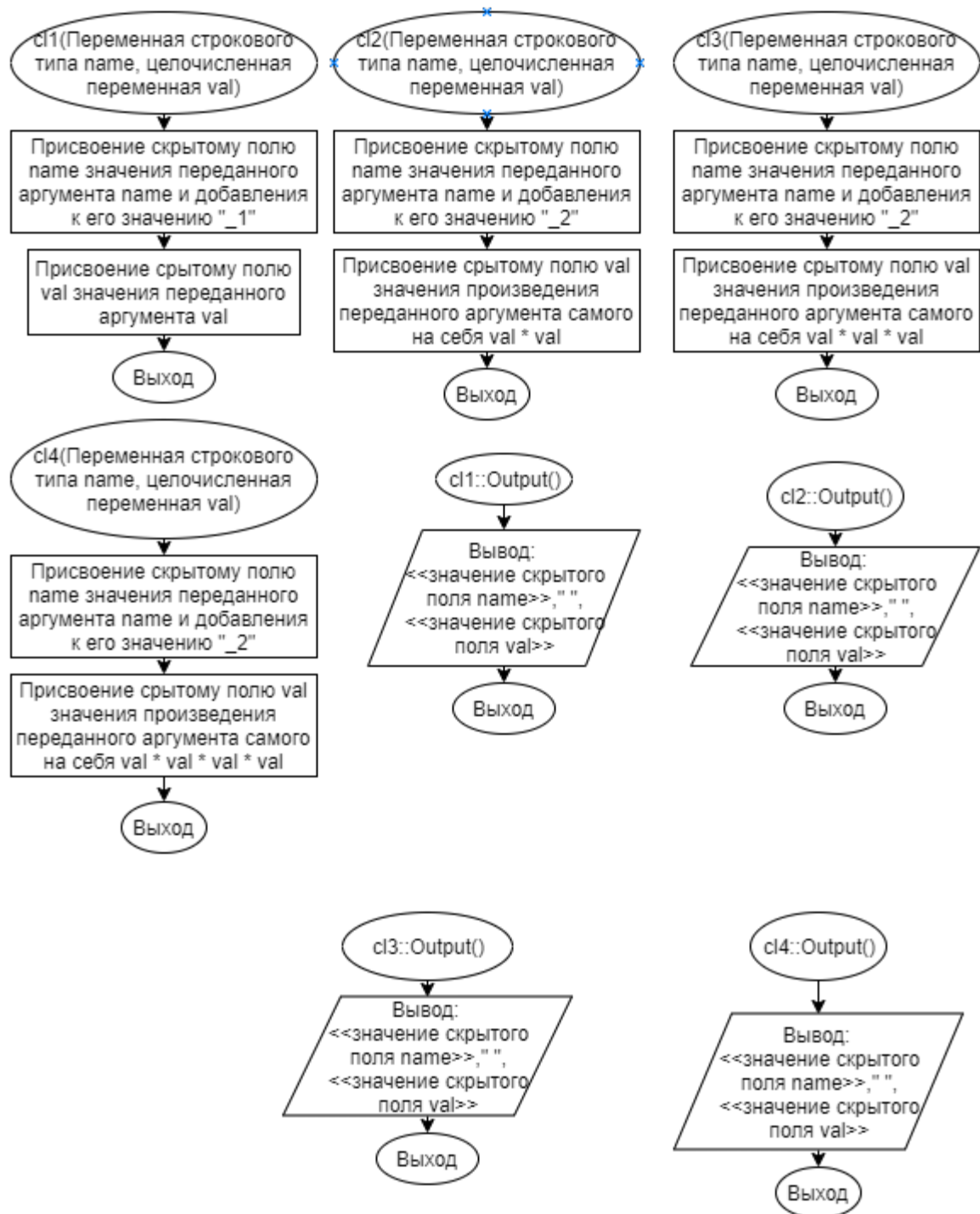


Рисунок 1 – Блок-схема алгоритма

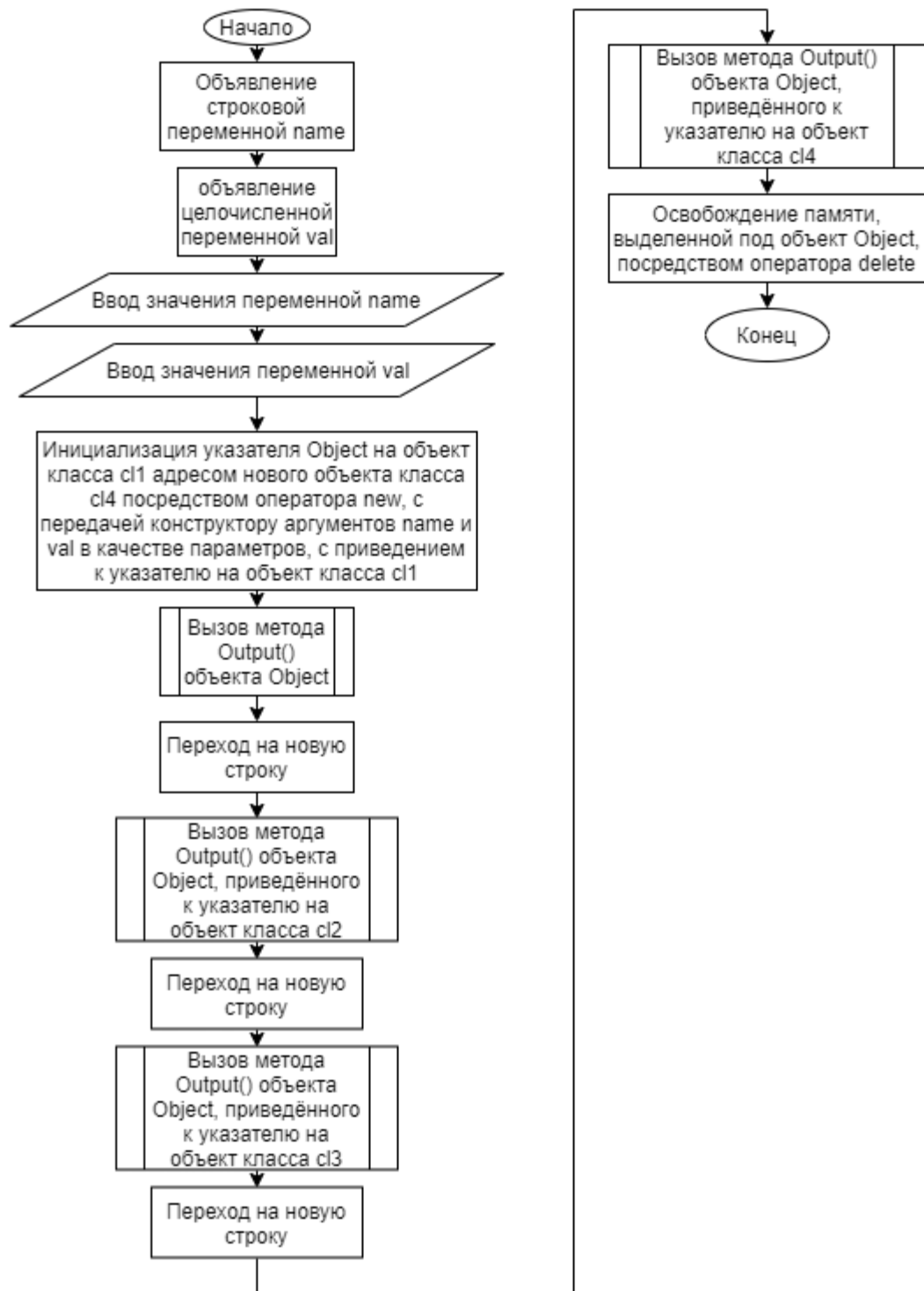


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl1.cpp

Листинг 1 – cl1.cpp

```
#include "cl1.h"
#include <iostream>
using namespace std;

cl1 :: cl1(string name, int val)
{
    this -> name = name + "_1";
    this -> val = val;
}
void cl1 :: Output()
{
    cout << name << " ";
    cout << val;
}
```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```
#ifndef __CL1__H
#define __CL1__H
#include <iostream>
using namespace std;
class cl1
{
    private:
        string name;
        int val;
    public:
        cl1(string name, int val);
        void Output();
};

#endif
```


5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

```
#include "cl2.h"
#include "cl1.h"
#include <iostream>
using namespace std;

cl2 :: cl2(string name, int val) : cl1 :: cl1(name, val)
{
    this -> name = name + "_2";
    this -> val = val * val;
}
void cl2 :: Output()
{
    cout << name << " ";
    cout << val;
}
```

5.4 Файл cl2.h

Листинг 4 – cl2.h

```
#ifndef __CL2__H
#define __CL2__H
#include "cl1.h"
#include <iostream>
using namespace std;
class cl2 : private cl1
{
    private:
        string name;
        int val;
    public:
        cl2(string name, int val);
        void Output();
};

#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

```
#include "cl3.h"
#include "cl2.h"
#include <iostream>
using namespace std;

cl3 :: cl3(string name, int val) : cl2 :: cl2(name, val)
{
    this -> name = name + "_3";
    this -> val = val * val * val;
}
void cl3 :: Output()
{
    cout << name << " ";
    cout << val;
}
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3__H
#define __CL3__H

#include "cl2.h"
#include <iostream>
using namespace std;
class cl3 : private cl2
{
    private:
        string name;
        int val;
    public:
        cl3(string name, int val);
        void Output();
};

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

```
#include "cl4.h"
#include "cl3.h"
#include <iostream>
using namespace std;

cl4 :: cl4(string name, int val) : cl3 :: cl3(name, val)
{
    this -> name = name + "_4";
    this -> val = val * val * val * val;
}
void cl4 :: Output()
{
    cout << name << " ";
    cout << val;
}
```

5.8 Файл cl4.h

Листинг 8 – cl4.h

```
#ifndef __CL4__H
#define __CL4__H

#include "cl3.h"
#include <iostream>
using namespace std;
class cl4 : private cl3
{
    private:
        string name;
        int val;
    public:
        cl4(string name, int val);
        void Output();
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include <iostream>
using namespace std;

int main()
{
    // program here
    string name;
    int val;
    cin >> name;
    cin >> val;
    cl1* Object = (cl1*) new cl4(name, val);
    Object -> Output();
    cout << "\n";
    ((cl2*)Object) -> Output();
    cout << "\n";
    ((cl3*)Object) -> Output();
    cout << "\n";
    ((cl4*)Object) -> Output();
    delete Object;
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
Object 2	Object_1 2 Object_2 4 Object_3 8 Object_4 16	Object_1 2 Object_2 4 Object_3 8 Object_4 16
ob 3	ob_1 3 ob_2 9 ob_3 27 ob_4 81	ob_1 3 ob_2 9 ob_3 27 ob_4 81
obj 4	obj_1 4 obj_2 16 obj_3 64 obj_4 256	obj_1 4 obj_2 16 obj_3 64 obj_4 256

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-avvora.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-avvora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
4. Шилдт Г. С++: 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).