Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

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 вызвать 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 параметризированный 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 параметризированный 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 class_4 предназначен для демонстрация наследования
;
- cin - объект стандартного потока;
- cout - объект стандартного потока.

Класс class_1:

- свойства/поля:
    - поле имя объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле целочисленной значение:
        - наименование — value;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод class_1 — конструктор;
    - метод ShowValues — вывод значений полей.

Класс class_2:

- свойства/поля:
    - поле имя объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле целочисленное значение:

        - наименование — value;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод ShowValues — вывод значений полей;
    - метод class_2 — конструктор.

Класс class_3:

- свойства/поля:
    - поле имя объекта:
        - наименование — name;
        - тип — int;
        - модификатор доступа — private;
    - поле целочисленное значение:
        - наименование — value;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод ShowValues — вывод значений полей;
    - метод class_3 — конструктор.

Класс class_4:

- свойства/поля:
    - поле имя объекта:
        - наименование — name;
        - тип — int;
        - модификатор доступа — private;
    - поле целочисленное значение:
        - наименование — value;

        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод ShowValues — вывод значений полей;
    - метод class_4 — конструктор.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	class_1			базовый класс	
		class_2	private		2
2	class_2			наследованный от первого класса	
		class_3	private		3
3	class_3			наследует методы второго класса	
		class_4	private		4
4	class_4			наследует методы третьего класса	

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 class_1

Функционал: конструктор.

Параметры: string, name, имя объекта; int, value, целочисленное значение.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса class_1

№	Предикат	Действия	№ перехода
1		поле name = значение переменной name_номер класса	2
2		поле value = значение переменной value	∅

3.2 Алгоритм метода ShowValues класса class_1

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода ShowValues класса class_1

№	Предикат	Действия	№ перехода
1		вывод значения поля name, вывод значения поля value, переход на новую строчку	∅

3.3 Алгоритм метода ShowValues класса class_2

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода ShowValues класса class_2

№	Предикат	Действия	№ перехода
1		вывод значения поля name, вывод значения поля value, переход на новую строчку	∅

3.4 Алгоритм конструктора класса class_2

Функционал: конструктор.

Параметры: string, name, имя объекта; int, value, целочисленное значение.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса class_2

№	Предикат	Действия	№ перехода
1		поле name = значение переменной name_номер класса	2
2		поле value = значение переменной value в степени номера класса	∅

3.5 Алгоритм метода ShowValues класса class_3

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода ShowValues класса class_3

№	Предикат	Действия	№ перехода
1		вывод значения поля name, вывод значения поля value, переход на новую строчку	∅

3.6 Алгоритм конструктора класса class_3

Функционал: конструктор.

Параметры: string, name, имя объекта; int, value, целочисленное значение.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса class_3

№	Предикат	Действия	№ перехода
1		поле name = значение переменной name_номер класса	2
2		поле value = значение переменной value в степени номера класса	∅

3.7 Алгоритм метода ShowValues класса class_4

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода ShowValues класса class_4

№	Предикат	Действия	№ перехода
1		вывод значения поля name, вывод значения поля value, переход на новую строчку	∅

3.8 Алгоритм конструктора класса class_4

Функционал: конструктор.

Параметры: string, name, имя объекта; int, value, целочисленное значение.

Алгоритм конструктора представлен в таблице 9.

Таблица 9 – Алгоритм конструктора класса class_4

№	Предикат	Действия	№ перехода
1		поле name = значение переменной name_номер класса	2
2		поле value = значение переменной value в степени номера класса	∅

3.9 Алгоритм функции main

Функционал: решение поставленной задачи.

Параметры: нет.

Возвращаемое значение: int - код успешности выполнения прораммы.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление целочисленной переменной value	2
2		объявление строковой переменной c_name	3
3		ввод c_name, value	4
4	value >= 2 и value <= 10	объявление указателя object на объект класса class_4	5
		вернуть 1	∅
5		использование метода ShowValues для object явно приведенного к class_1	6
6		использование метода ShowValues для object явно приведенного к class_2	7

№	Предикат	Действия	№ перехода
7		использование метода ShowValues для object явно приведенного к class_3	8
8		использование метода ShowValues для object явно приведенного к class_4	9
9		возврат 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

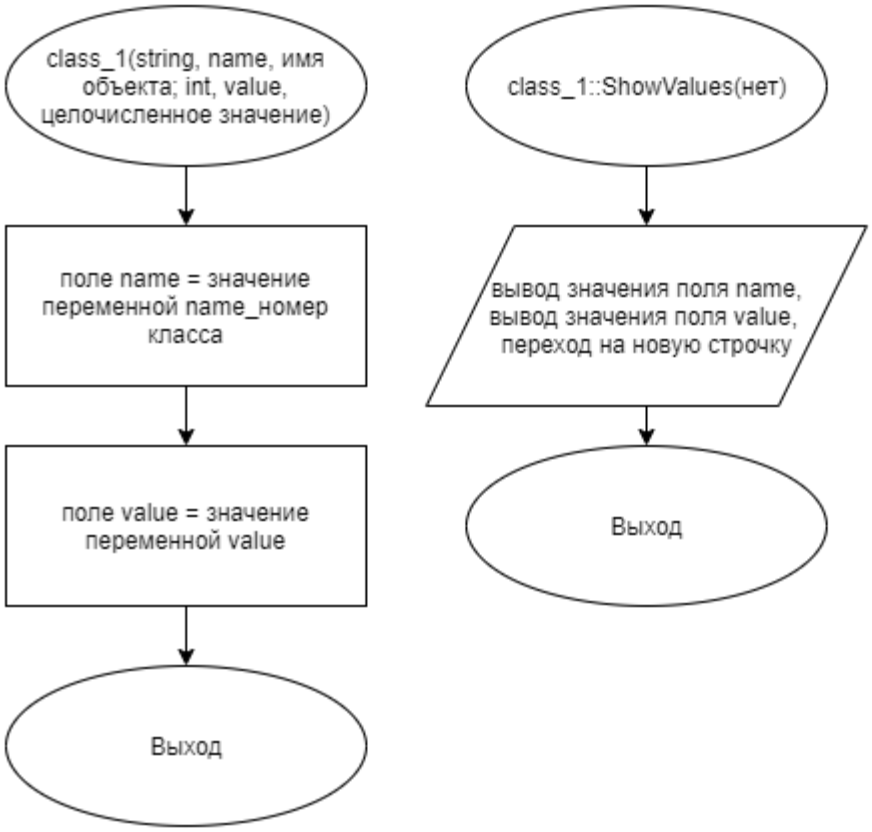

Рисунок 1 – Блок-схема алгоритма

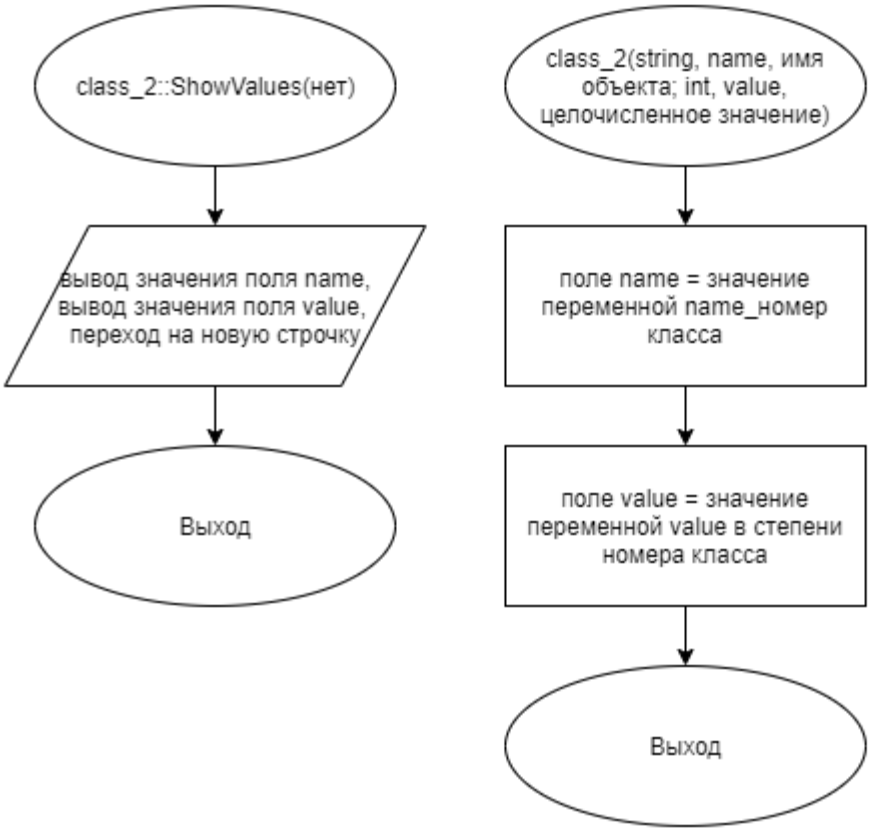

Рисунок 2 – Блок-схема алгоритма

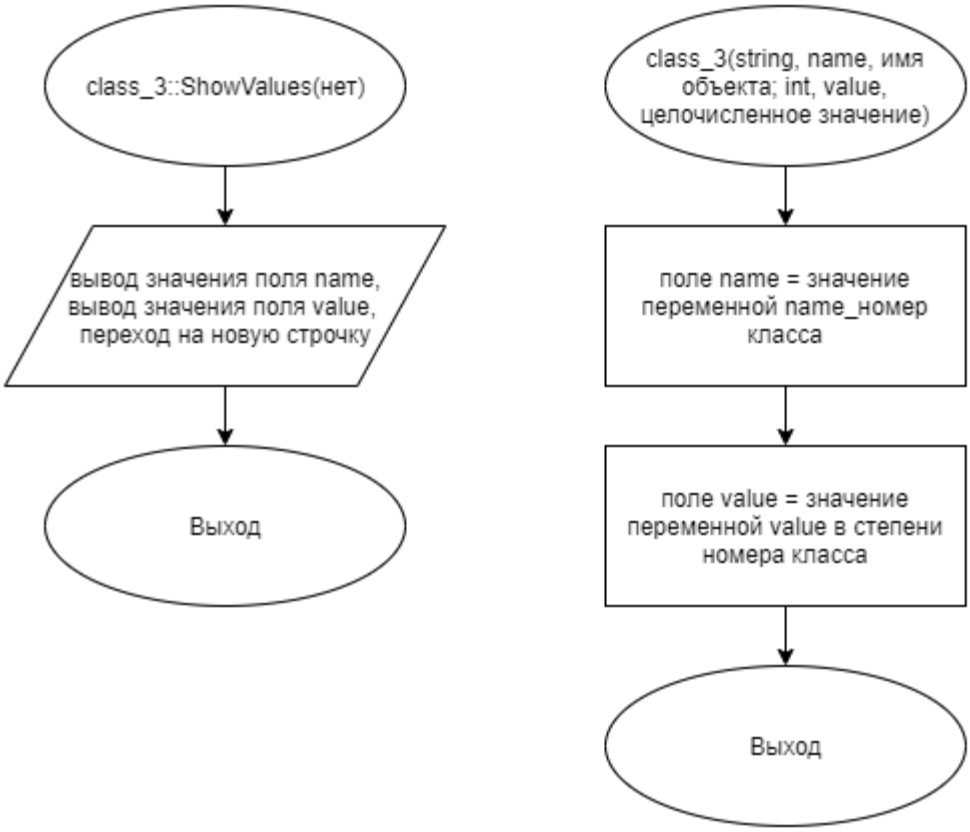

Рисунок 3 – Блок-схема алгоритма

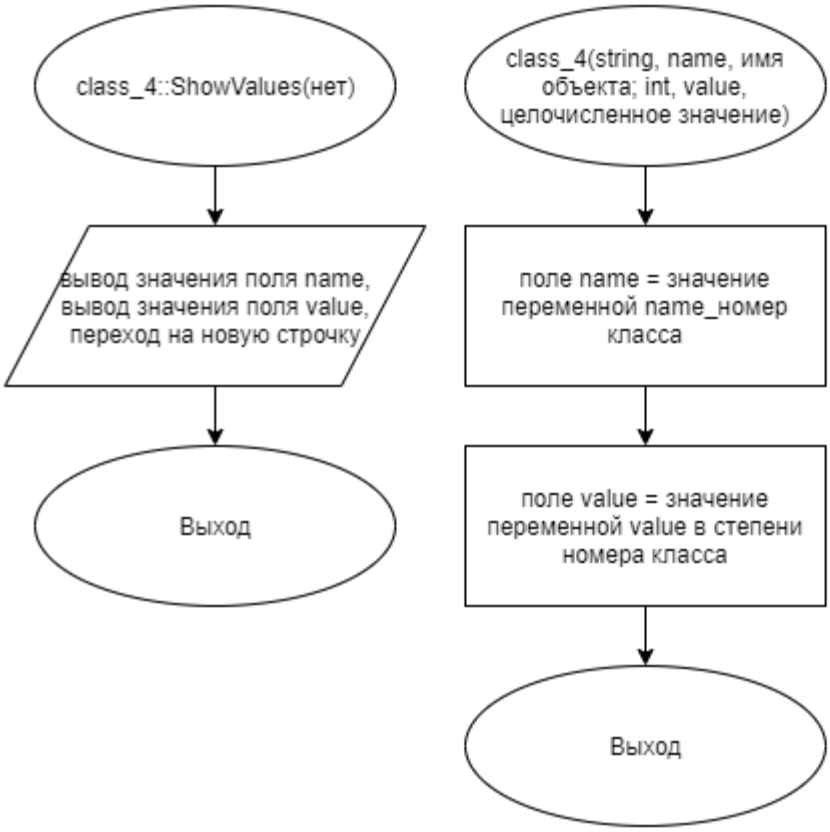

Рисунок 4 – Блок-схема алгоритма

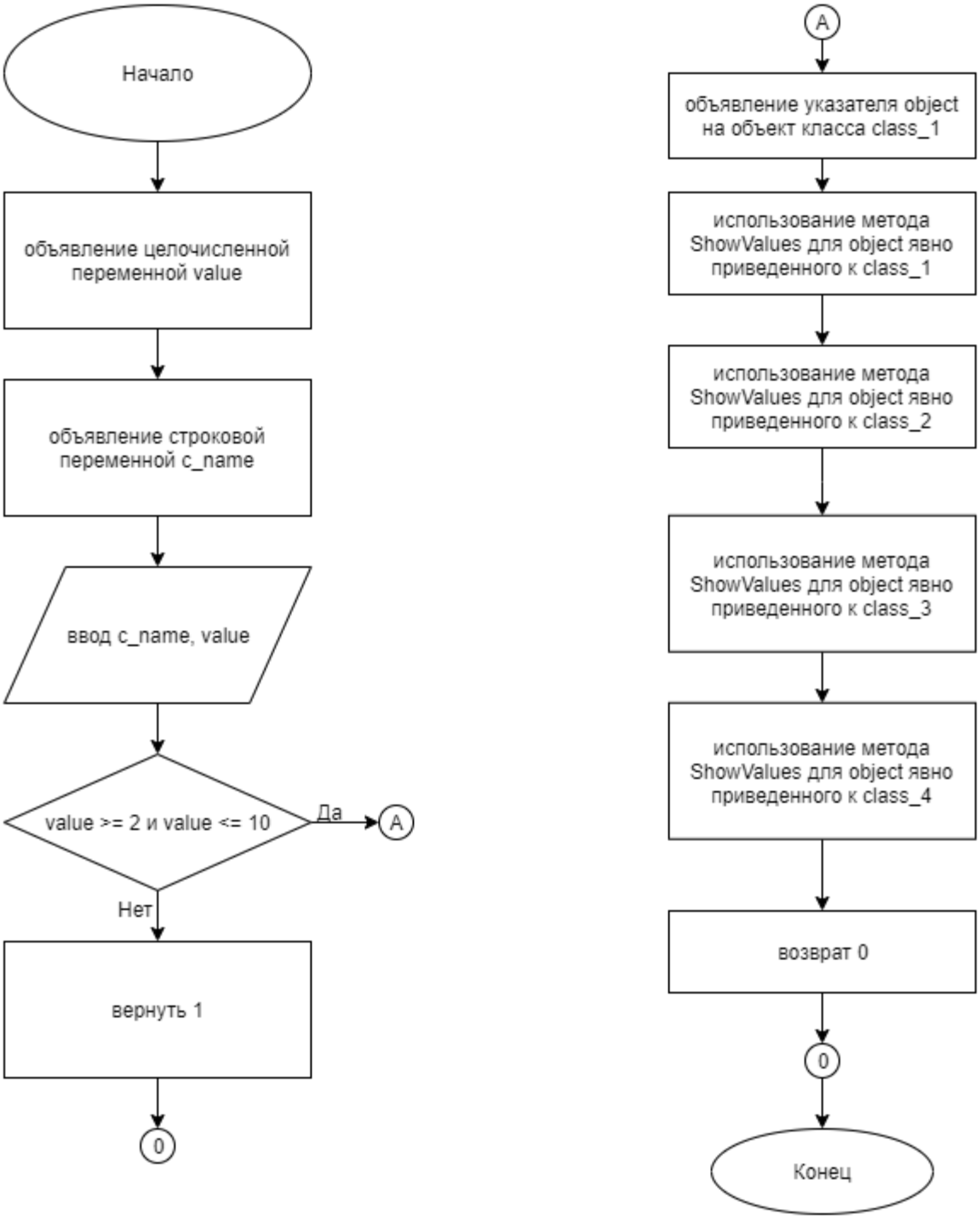

Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл class_1.cpp

Листинг 1 – class_1.cpp

```
#include "class_1.h"

class_1::class_1(std::string name, int value){
   this->name = name + "_1";
   this->value = value;
}
void class_1::ShowValues(){
   std::cout << this->name << " " << this->value << "\n";
}
```

5.2 Файл class_1.h

Листинг 2 – class_1.h

```
#ifndef __CLASS_1__H
#define __CLASS_1__H
#include <iostream>
#include <string>
#include <math.h>

class class_1{
public:
   class_1(std::string c_name, int c_value);
   void ShowValues();

private:
   int value;
   std::string name;
};

#endif
```

5.3 Файл class_2.cpp

Листинг 3 – class_2.cpp

```
#include "class_2.h"

class_2::class_2(std::string name, int value) : class_1(name, value){
   this->name = name + "_2";
   this->value = pow(value, 2);
}
void class_2::ShowValues(){
   std::cout << this->name << " " << this->value << "\n";
}
```

5.4 Файл class_2.h

Листинг 4 – class_2.h

```
#ifndef __CLASS_2__H
#define __CLASS_2__H
#include "class_1.h"

class class_2 : private class_1{
public:
   class_2(std::string c_name, int c_value);
   void ShowValues();
private:
   std::string name;
   int value;
};
#endif
```

5.5 Файл class_3.cpp

Листинг 5 – class_3.cpp

```
#include "class_3.h"

class_3::class_3(std::string name, int value) : class_2(name, value){
   this->name = name + "_3";
   this->value = pow(value, 3);
}
void class_3::ShowValues(){
```

```
   std::cout << this->name << " " << this->value << "\n";
}
```

5.6 Файл class_3.h

Листинг 6 – class_3.h

```
#ifndef __CLASS_3__H
#define __CLASS_3__H
#include "class_2.h"

class class_3 : private class_2{
public:
   class_3(std::string c_name, int c_value);
   void ShowValues();
private:
   std::string name;
   int value;
};

#endif
```

5.7 Файл class_4.cpp

Листинг 7 – class_4.cpp

```
#include "class_4.h"

class_4::class_4(std::string name, int value) : class_3(name, value){
   this->name = name + "_4";
   this->value = pow(value, 4);
}
void class_4::ShowValues(){
   std::cout << this->name << " " << this->value << "\n";
}
```

5.8 Файл class_4.h

Листинг 8 – class_4.h

```
#ifndef __CLASS_4__H
#define __CLASS_4__H
#include "class_3.h"

class class_4 : private class_3{
public:
   class_4(std::string c_name, int c_value);
   void ShowValues();
private:
   std::string name;
   int value;
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "class_4.h"

int main()
{
   int c_value;
   std::string c_name;
   std::cin >> c_name >> c_value;

   if (c_value >= 2 && c_value <= 10){
      class_1* object =(class_1*)(new class_4(c_name, c_value));
      ((class_1*)object)->ShowValues();
      ((class_2*)object)->ShowValues();
      ((class_3*)object)->ShowValues();
      ((class_4*)object)->ShowValues();

      return 0;
   }
   return 1;
}
```

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`

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).