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 a класса cl_4 предназначен для указатель на объект производного класса cl_4;
- функция pow для возведение числа в степень;
- cin - стандартный объект потока ввода данных;
- cout - стандартный объект потока вывода данных.

Класс cl_1:

- свойства/поля:
  - поле хранение имени объекта:
    - наименование — name;
    - тип — string;
    - модификатор доступа — private;
  - поле закрытое поле:
    - наименование — x;
    - тип — int;
    - модификатор доступа — private;
- функционал:
  - метод cl_1 — конструктор;
  - метод out — вывод значений полей;
  - метод ~cl_1 — деструктор.

Класс cl_2:

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

        - модификатор доступа — private;
    - поле закрытое поле:
        - наименование — x;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод cl_2 — конструктор;
    - метод out — вывод значений полей.

Класс cl_3:

- свойства/поля:
    - поле хранение имени объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле закрытое поле:
        - наименование — x;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод cl_3 — конструктор;
    - метод out — вывод значений полей.

Класс cl_4:

- свойства/поля:
    - поле хранение имени объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;

- поле закрытое поле:
    - наименование — x;
    - тип — int;
    - модификатор доступа — private;
- функционал:
    - метод cl_4 — конструктор;
    - метод out — вывод значений полей.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl_1			родительский класс	
		cl_2	virtual public		2
2	cl_2			производный класс от cl_1	
		cl_3	virtual public		3
3	cl_3			производный класс от cl_2	
		cl_4	virtual public		4
4	cl_4			производный клсс от cl_3	

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 cl_1

Функционал: конструктор.

Параметры: string - name1 - название объекта; int - x1 - закрытое поле.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl_1

№	Предикат	Действия	№ перехода
1		присвоение полю name = name1 +"_1"	2
2		присвоение полю x значения x1 в степени 1	∅

3.2 Алгоритм метода out класса cl_1

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: void - не возвращает значений.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода out класса cl_1

№	Предикат	Действия	№ перехода
1		вывод с новой строки на экран полей через пробел	∅

3.3 Алгоритм деструктора класса cl_1

Функционал: деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса cl_1

№	Предикат	Действия	№ перехода
1		уничтожение объекта	∅

3.4 Алгоритм конструктора класса cl_2

Функционал: конструктор.

Параметры: string - name1 - название объекта; int - x1 - закрытое поле.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса cl_2

№	Предикат	Действия	№ перехода
1		присвоение полю name = name1 +"_2"	2
2		присвоение полю x значения x1 в степени 2	∅

3.5 Алгоритм метода out класса cl_2

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: void - не возвращает значений.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода out класса cl_2

№	Предикат	Действия	№ перехода
1		вывод с новой строки на экран полей через пробел	∅

3.6 Алгоритм конструктора класса cl_3

Функционал: конструктор.

Параметры: string - name1 - название объекта; int - x1 - закрытое поле.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса cl_3

№	Предикат	Действия	№ перехода
1		присвоение полю name = name1 +"_3"	2
2		присвоение полю x значения x1 в степени 3	∅

3.7 Алгоритм метода out класса cl_3

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: void - не возвращает значений.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода out класса cl_3

№	Предикат	Действия	№ перехода
1		вывод с новой строки на экран полей через пробел	∅

3.8 Алгоритм конструктора класса cl_4

Функционал: конструктор.

Параметры: string - name1 - название объекта; int - x1 - закрытое поле.

Алгоритм конструктора представлен в таблице 9.

Таблица 9 – Алгоритм конструктора класса cl_4

№	Предикат	Действия	№ перехода
1		присвоение полю name = name1 +"_4"	2
2		присвоение полю x значения x1 в степени 4	∅

3.9 Алгоритм метода out класса cl_4

Функционал: вывод значений полей.

Параметры: нет.

Возвращаемое значение: void - не возвращает значений.

Алгоритм метода представлен в таблице 10.

Таблица 10 – Алгоритм метода out класса cl_4

№	Предикат	Действия	№ перехода
1		вывод с новой строки на экран полей через пробел	∅

3.10 Алгоритм функции main

Функционал: основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: целочисленное - индикатор корректности выполнения алгоритма.

Алгоритм функции представлен в таблице 11.

Таблица 11 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление строковой переменой name1	2
2		объявление целочисленной x1	3

№	Предикат	Действия	№ перехода
3		ввод name1 с клавиатуры	4
4		ввод x1 с клавиатуры	5
5		создание указателя на производный объект a класса cl_4 с параметрами name1, x1 от родительского класса cl_1	6
6		обращение к методу out родительского класса cl_1 для объекта a	7
7		обращение к методу out родительского класса cl_2 для объекта a	8
8		обращение к методу out родительского класса cl_3 для объекта a	9
9		обращение к методу out производного класса cl_4 для объекта a	10
10		освобождение памяти, выделенной для объекта a	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

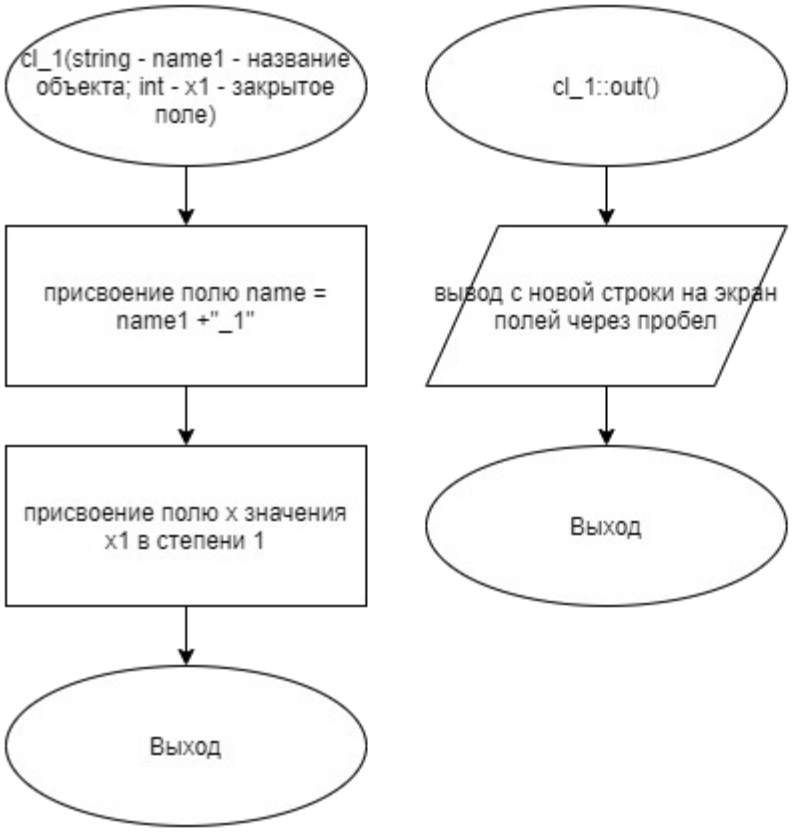

Рисунок 1 – Блок-схема алгоритма

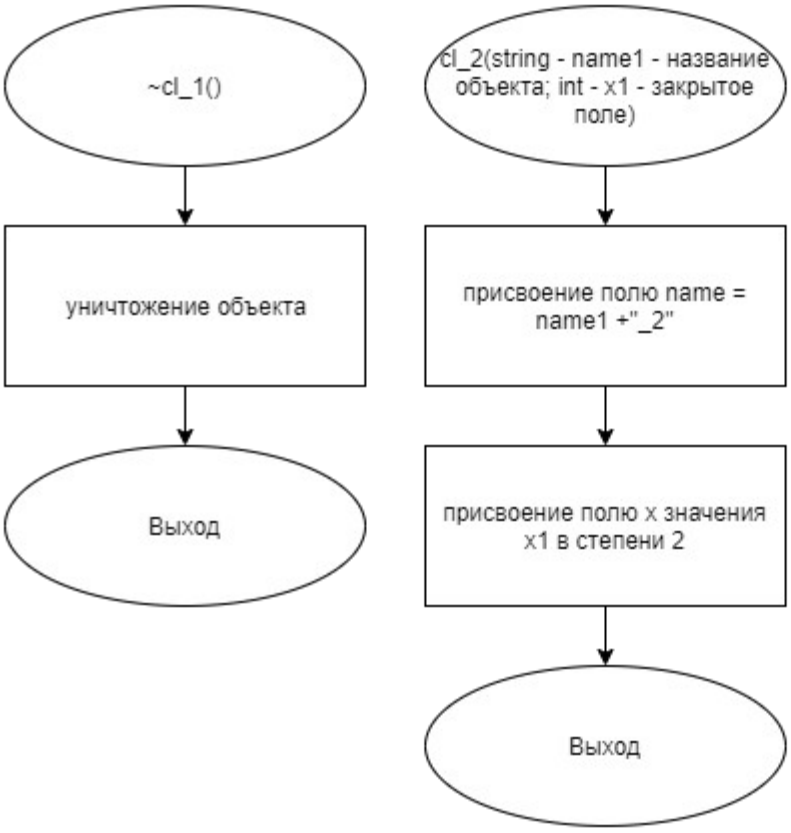

Рисунок 2 – Блок-схема алгоритма

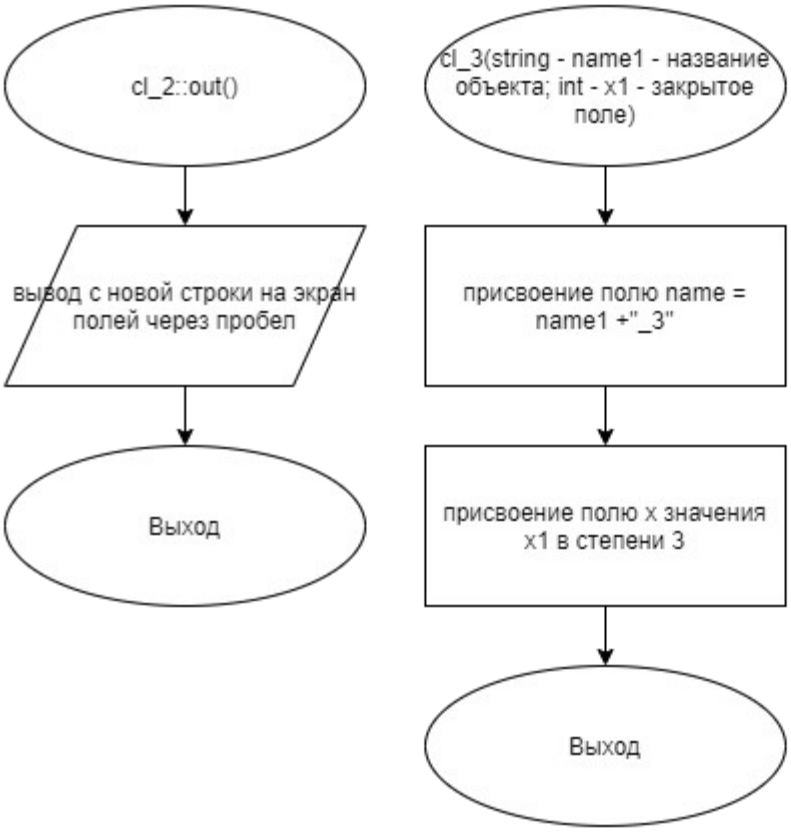

Рисунок 3 – Блок-схема алгоритма

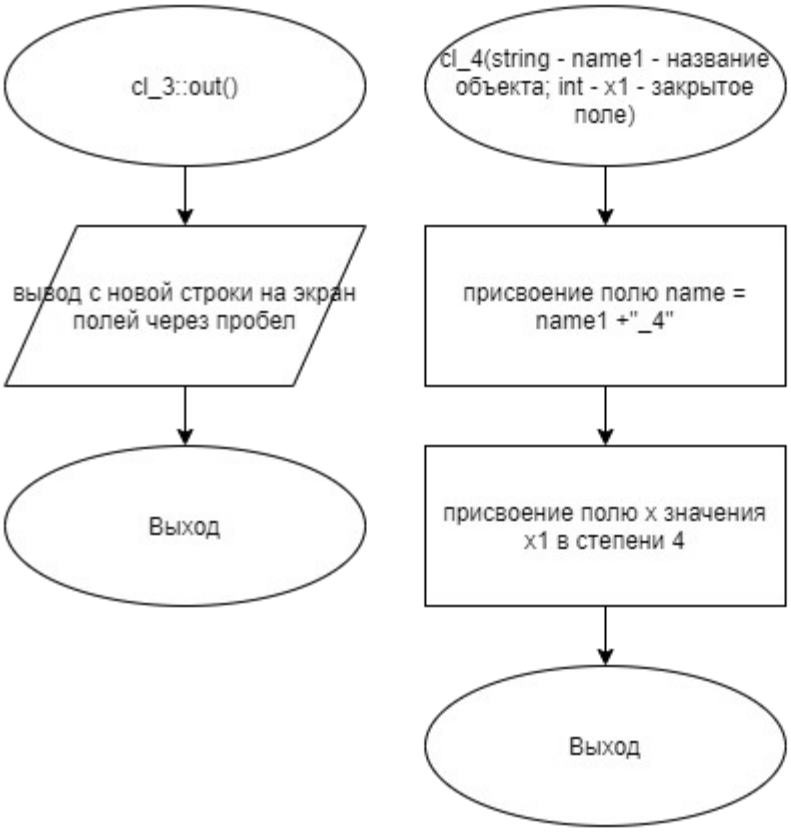

Рисунок 4 – Блок-схема алгоритма

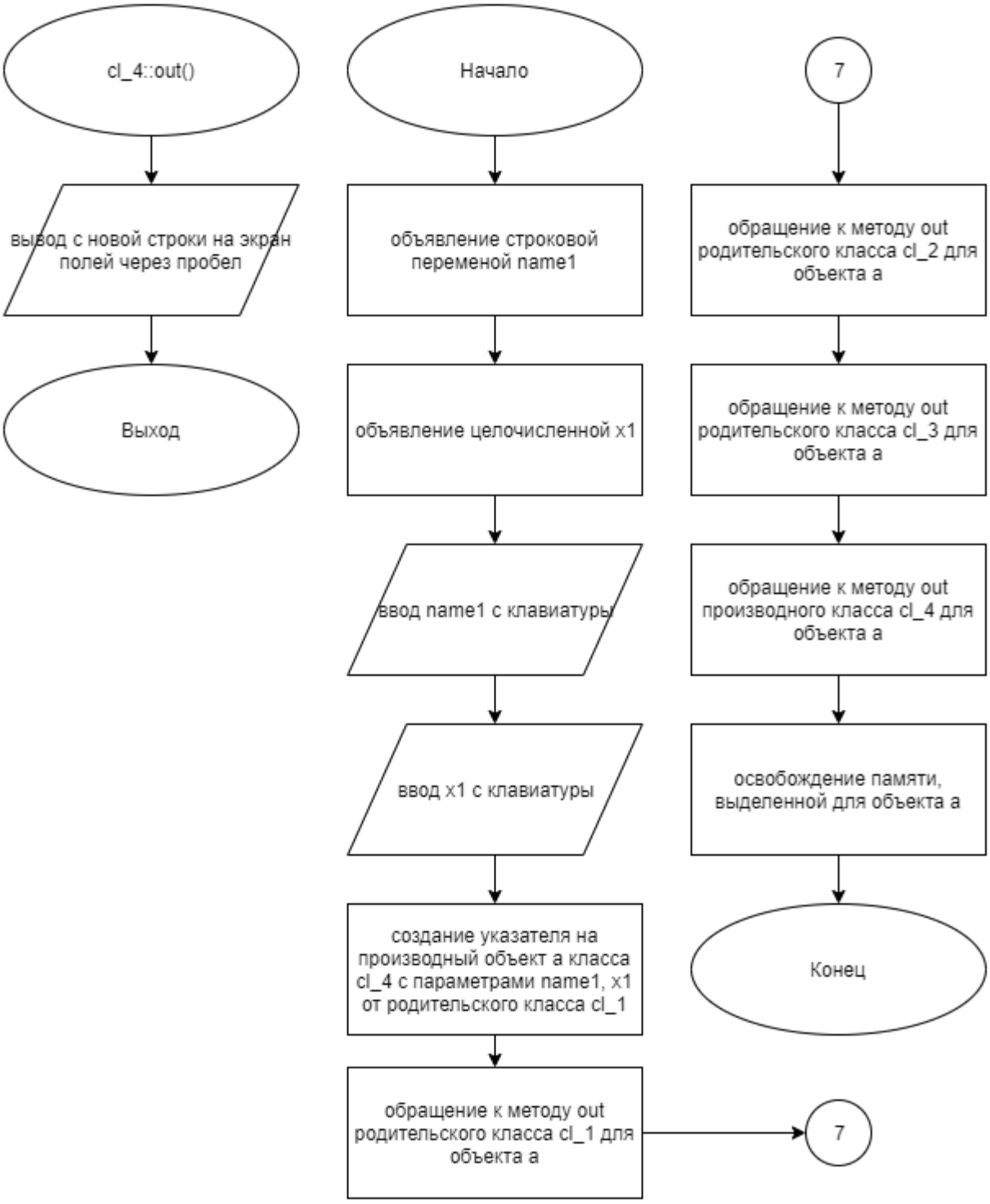

Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl_1.cpp

Листинг 1 – cl_1.cpp

```
#include "cl_1.h"
#include <iostream>
#include <string>
#include <cmath>
using namespace std;
cl_1::cl_1(string name1, int x1){
   name=name1+"_1";
   x = pow(x1,1);
}
void cl_1::out(){
   cout<<name<<" "<<x<<endl;
}
cl_1::~cl_1(){}
```

5.2 Файл cl_1.h

Листинг 2 – cl_1.h

```
#ifndef __CL_1__H
#define __CL_1__H
#include <iostream>
#include <string>
using namespace std;
class cl_1{
   private:
   string name;
   int x;
   public:
   cl_1(string name1, int x1);
   void out();
   ~cl_1();

};
#endif
```

5.3 Файл cl_2.cpp

Листинг 3 – cl_2.cpp

```
#include "cl_2.h"
#include <cmath>
using namespace std;
cl_2::cl_2(string name1, int x1):cl_1(name1,x1){
   name=name1+"_2";
   x = pow(x1,2);
}
void cl_2::out(){
   cout<<name<<" "<<x<<endl;
}
```

5.4 Файл cl_2.h

Листинг 4 – cl_2.h

```
#ifndef __CL_2__H
#define __CL_2__H
#include "cl_1.h"
#include <iostream>
#include <string>
using namespace std;
class cl_2:private cl_1{
   private:
   string name;
   int x;
   public:
   cl_2(string name1, int x1);
   void out();
};

#endif
```

5.5 Файл cl_3.cpp

Листинг 5 – cl_3.cpp

```
#include "cl_3.h"
```

```
#include <cmath>
using namespace std;
cl_3::cl_3(string name1, int x1):cl_2(name1,x1){
   name=name1+"_3";
   x = pow(x1,3);
}
void cl_3::out(){
   cout<<name<<" "<<x<<endl;
}
```

5.6 Файл cl_3.h

Листинг 6 – cl_3.h

```
#ifndef __CL_3__H
#define __CL_3__H
#include "cl_2.h"
#include <iostream>
#include <string>
using namespace std;
class cl_3:private cl_2{
   private:
   string name;
   int x;
   public:
   cl_3(string name1, int x1);
   void out();
};

#endif
```

5.7 Файл cl_4.cpp

Листинг 7 – cl_4.cpp

```
#include "cl_4.h"
#include <cmath>
using namespace std;
cl_4::cl_4(string name1, int x1):cl_3(name1,x1){
   name=name1+"_4";
   x = pow(x1,4);
}
void cl_4::out(){
   cout<<name<<" "<<x<<endl;
```

```
}
```

5.8 Файл cl_4.h

Листинг 8 – cl_4.h

```
#ifndef __CL_4__H
#define __CL_4__H
#include "cl_3.h"
#include <iostream>
#include <string>
using namespace std;
class cl_4:private cl_3{
   private:
   string name;
   int x;
   public:
   cl_4(string name1, int x1);
   void out();
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <string>
#include "cl_1.h"
#include "cl_2.h"
#include "cl_3.h"
#include "cl_4.h"
using namespace std;
int main()
{
   string name1;
   int x1;
   cin>>name1;
   cin>>x1;
   cl_1* a = (cl_1*) new cl_4(name1,x1);
   a->out();
   ((cl_2*)a) -> out();
```

```
        ((cl_3*)a) -> out();
        ((cl_4*)a) -> out();
        delete a;
        return(0);
    }
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 12.

Таблица 12 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**

`Ob 2`	`Ob_1 2`
	`Ob_2 4`
	`Ob_3 8`
	`Ob_4 16`
		`Ob_1 2`
		`Ob_2 4`
		`Ob_3 8`
		`Ob_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).