

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 Print класса cl1.....	11
3.3 Алгоритм конструктора класса cl2.....	12
3.4 Алгоритм метода Print класса cl2.....	12
3.5 Алгоритм конструктора класса cl3.....	12
3.6 Алгоритм метода Print класса cl3.....	13
3.7 Алгоритм конструктора класса cl4.....	13
3.8 Алгоритм метода Print класса cl4.....	14
3.9 Алгоритм функции main.....	14
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	16
5 КОД ПРОГРАММЫ.....	20
5.1 Файл cl1.cpp.....	20
5.2 Файл cl1.h.....	20
5.3 Файл cl2.cpp.....	21
5.4 Файл cl2.h.....	21
5.5 Файл cl3.cpp.....	22
5.6 Файл cl3.h.....	22
5.7 Файл cl4.cpp.....	23
5.8 Файл cl4.h.....	23
5.9 Файл main.cpp.....	24
6 ТЕСТИРОВАНИЕ.....	25

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	26
---------------------------------------	----

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 `cin` класса `istream` предназначен для потока ввода;
- объект `cout` класса `ostream` предназначен для потока вывода;
- объект `MyClass` класса `Object` предназначен для демонстрации иерархии наследования;
- функция `main` для основного алгоритма функции.

Класс `cl1`:

- свойства/поля:
 - поле хранения названия объекта:
 - наименование — `obj1`;
 - тип — `string`;
 - модификатор доступа — `private`;
 - поле хранения значения объекта:
 - наименование — `obj2`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `cl1` — конструктор , устанавливает имя и целочисленное значение объекту;
 - метод `Print` — выводит имя объекта и его значение.

Класс `cl2`:

- свойства/поля:
 - поле хранения названия объекта:
 - наименование — `obj1`;
 - тип — `string`;

- модификатор доступа — private;
- о поле хранения названия объекта:
 - наименование — obj2;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - о метод cl2 — конструктор, устанавливает имя и целочисленное значение объекту;
 - о метод Print — выводит имя объекта и его значение.

Класс cl3:

- свойства/поля:
 - о поле хранения названия объекта:
 - наименование — obj1;
 - тип — string;
 - модификатор доступа — private;
 - о поле хранения названия объекта:
 - наименование — obj2;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - о метод cl3 — конструктор, устанавливает имя и целочисленное значение объекту;
 - о метод Print — выводит имя объекта и его значение.

Класс cl4:

- свойства/поля:
 - о поле хранения названия объекта:
 - наименование — obj1;

- тип — string;
- модификатор доступа — private;
- о поле хранения названия объекта:
 - наименование — obj2;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - о метод cl4 — конструктор, устанавливает имя и целочисленное значение объекту;
 - о метод Print — выводит имя объекта и его значение.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl1			родительский класс , содержит основные поля и методы	
2	cl2			дочерний класс класса cl1	
		cl1	private		1
3	cl3			дочерний класс класса class2	
		cl2	private		2
4	cl4			дочерний класс класса cl3	
		cl3	private		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 *cl1*

Функционал: конструктор , устанавливает имя и целочисленное значение объекту.

Параметры: *strubg obj1*, *int obj2*.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *cl1*

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству <i>obj1</i> значение параметра <i>obj1</i> и дописать "_1"	2
2		присвоить скрытому свойству <i>obj2</i> значение параметра <i>obj2</i>	Ø

3.2 Алгоритм метода *Print* класса *cl1*

Функционал: выводит имя объекта и его значение.

Параметры: нет.

Возвращаемое значение: *void*.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *Print* класса *cl1*

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств <i>obj1</i> и <i>obj2</i> через пробел	Ø

3.3 Алгоритм конструктора класса cl2

Функционал: конструктор, устанавливает имя и целочисленное значение объекту.

Параметры: string obj1, int obj2.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl2

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству obj1 значение параметра obj1 и дописать "_2"	2
2		присвоить скрытому свойству obj2 значение параметра obj2*obj2	Ø

3.4 Алгоритм метода Print класса cl2

Функционал: выводит имя объекта и его значение.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Print класса cl2

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств obj1 и obj2 через пробел	Ø

3.5 Алгоритм конструктора класса cl3

Функционал: конструктор, устанавливает имя и целочисленное значение объекту.

Параметры: string obj1, int obj2.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса *cl3*

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству obj1 значение параметра obj1 и дописать "_3"	2
2		присвоить скрытому свойству obj2 значение параметра obj2*obj2*obj2	Ø

3.6 Алгоритм метода Print класса *cl3*

Функционал: выводит имя объекта и его значение.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода *Print* класса *cl3*

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств obj1 и obj2 через пробел	Ø

3.7 Алгоритм конструктора класса *cl4*

Функционал: конструктор, устанавливает имя и целочисленное значение объекту.

Параметры: string obj1, int obj2.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса *cl4*

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству obj значение параметра obj1 и дописать "_4"	2
2		присвоить скрытому свойству obj2 значение параметра	Ø

№	Предикат	Действия	№ перехода
		obj2*obj2*obj2*obj2*	

3.8 Алгоритм метода Print класса cl4

Функционал: выводит имя объекта и его значение.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода Print класса cl4

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств obj1 и obj2 через пробел	Ø

3.9 Алгоритм функции main

Функционал: основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменной obj1 типа string	2
2		объявление переменной obj2 типа string	3
3		ввод значения переменной obj1	4
4		ввод значения переменной obj2	5
5		инициализация указателя object на объект класса cl1 адресом нового объекта класса cl4 с помощью оператора функции new с передачей конструктору параметров obj1,obj2 с приведением к указателю на	6

№	Предикат	Действия	№ перехода
		объект класса cl1	
6		вызов метода Print() объекта object	7
7		вывод переноса на новую строку	8
8		вызов метода Print() объекта object , риведенный к указателю на объект класса cl2	9
9		вывод переноса на новую строку	10
10		вызов метода Print() объектом object, приведенный к указателю на объект класса cl3	11
11		вывод переноса на новую строку	12
12		вызов метода Print() объекта object, приведенный к указателю на объект класса cl4	13
13		освобождение памяти , выделенной под объект object с помощью оператора функции delete	14
14		return 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

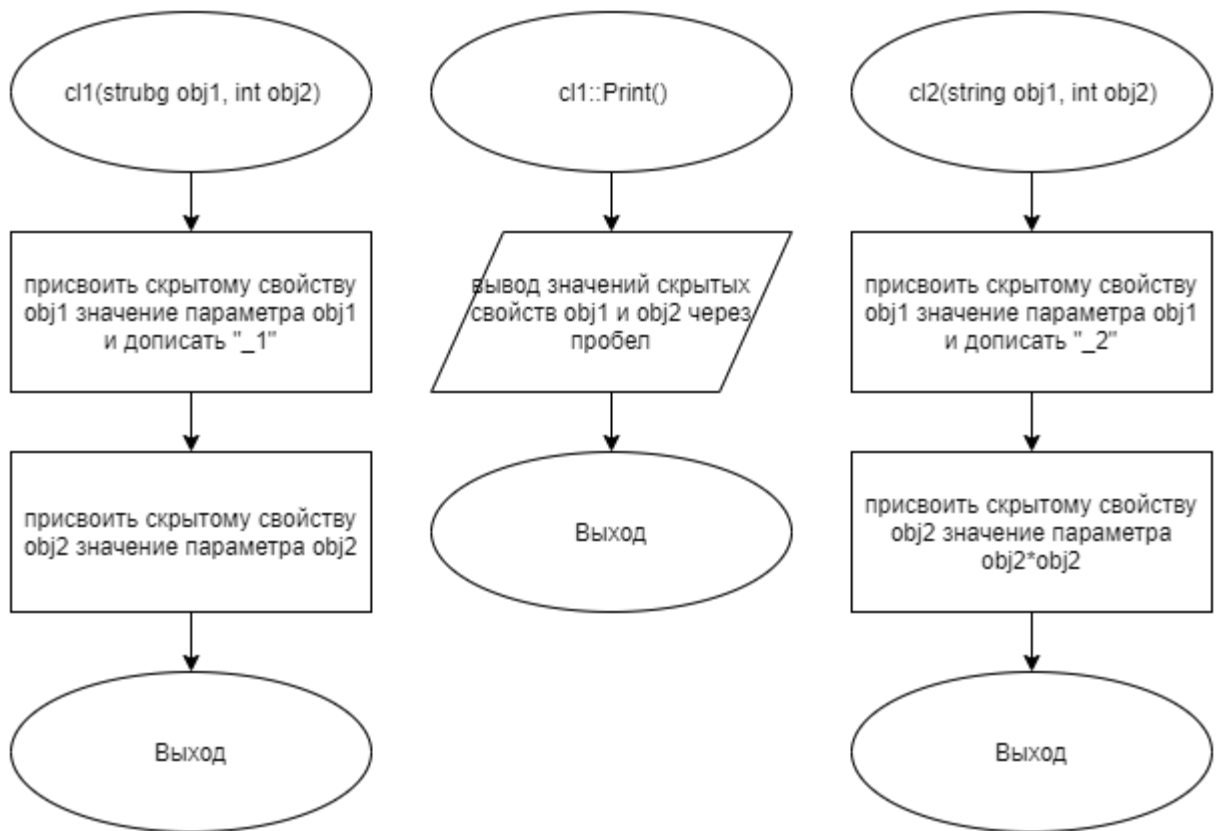


Рисунок 1 – Блок-схема алгоритма

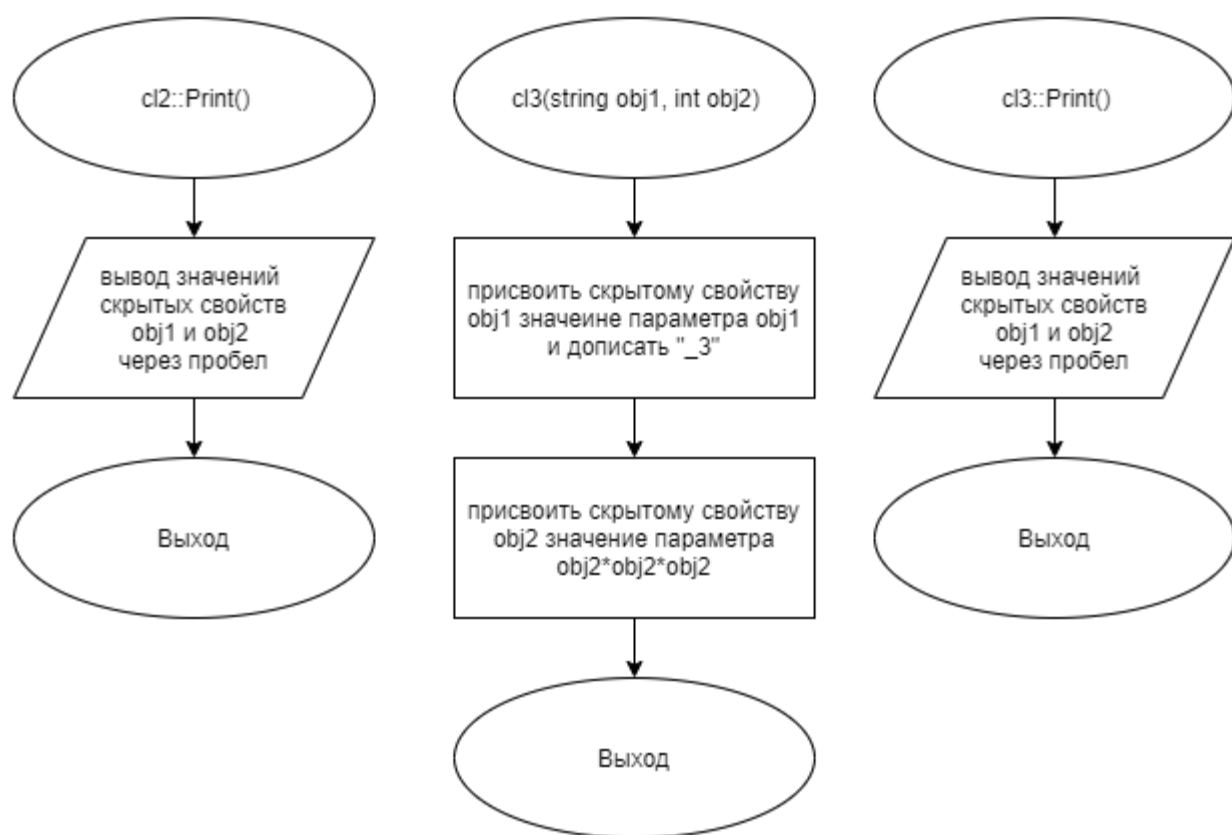


Рисунок 2 – Блок-схема алгоритма

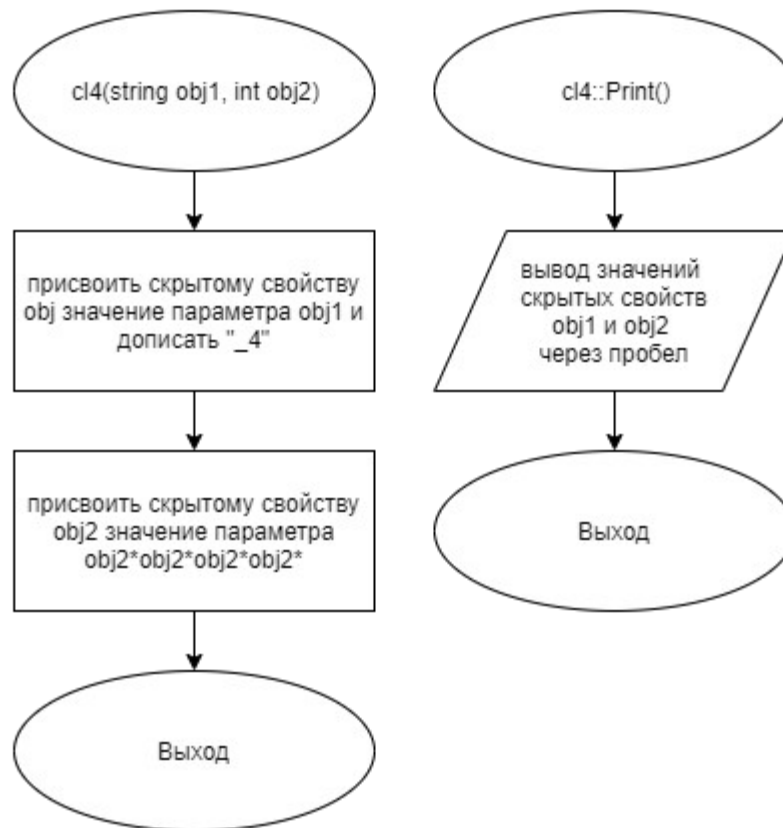


Рисунок 3 – Блок-схема алгоритма

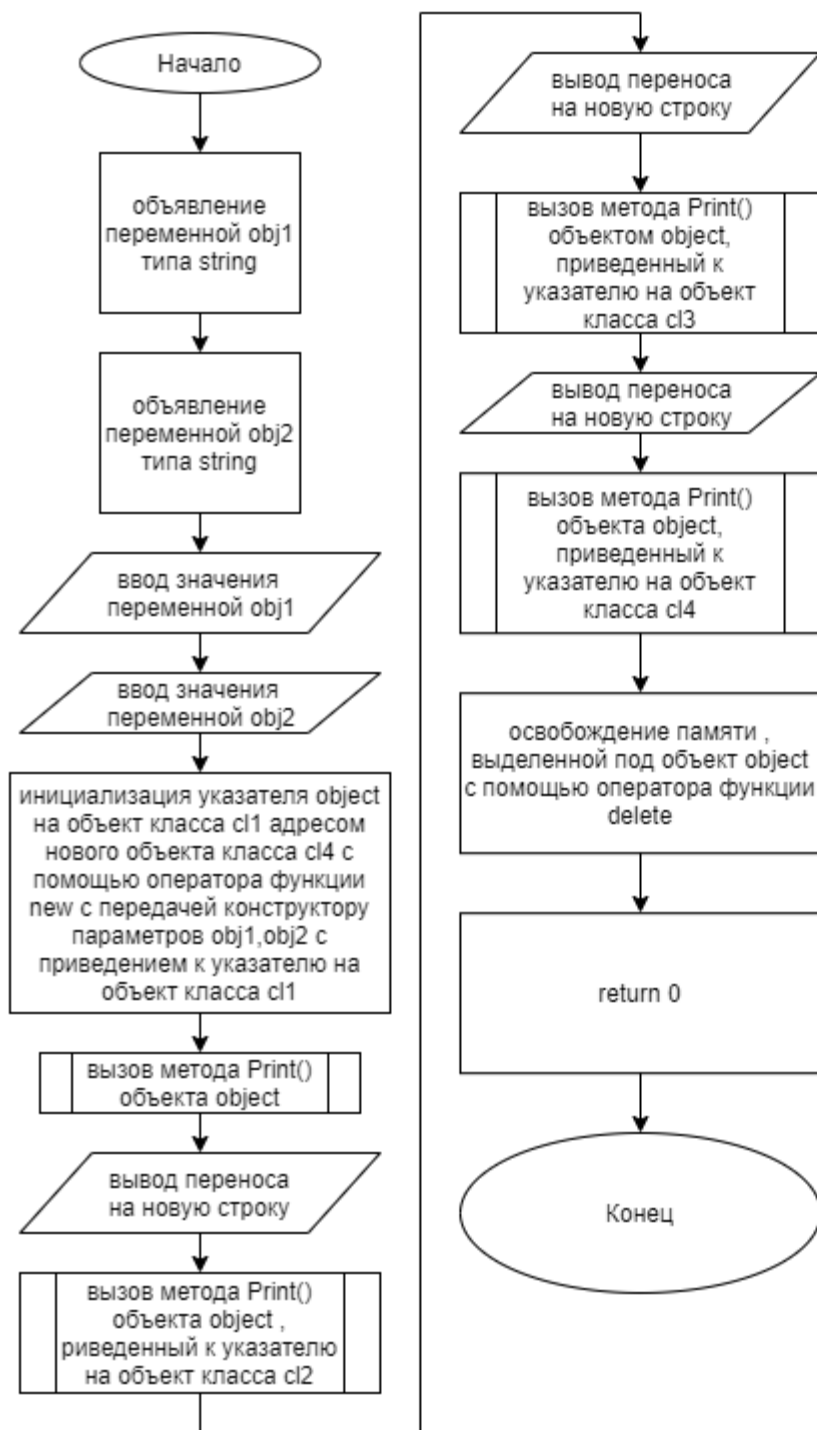


Рисунок 4 – Блок-схема алгоритма

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>

cl1::cl1(string obj1, int obj2)
{
    this ->obj1=obj1+"_1";
    this ->obj2=obj2;
}
void cl1::Print()
{
    cout<<obj1<<" ";
    cout<<obj2;
}
```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```
#ifndef __CL1__H
#define __CL1__H
#include <iostream>

using namespace std;

class cl1
{
private:
    string obj1;
    int obj2;
public:
    cl1(string obj1, int obj2);
    void Print();
};
```

```
#endif
```

5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

```
#include <iostream>
#include "cl2.h"
#include "cl1.h"

cl2::cl2(string obj1, int obj2):cl1::cl1(obj1, obj2)
{
    this -> obj1=obj1+"_2";
    this -> obj2=obj2*obj2;
}
void cl2::Print()
{
    cout<<obj1<<" ";
    cout<<obj2;
}
```

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 obj1;
        int obj2;
    public:
        cl2(string obj1, int obj2);
        void Print();
};

#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

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

cl3::cl3(string obj1, int obj2):cl2::cl2(obj1 , obj2)
{
    this -> obj1=obj1+"_3";
    this ->obj2=obj2*obj2*obj2;
}
void cl3::Print()
{
    cout<<obj1<<" ";
    cout<<obj2;
}
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3__H
#define __CL3__H
#include <iostream>
#include "cl2.h"
using namespace std;

class cl3:private cl2
{
    private:
        string obj1;
        int obj2;
    public:
        cl3(string obj1, int obj2);
        void Print();
};

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

```
#include "cl4.h"
#include "cl3.h"

using namespace std;

cl4::cl4(string obj1, int obj2):cl3::cl3(obj1, obj2)
{
    this -> obj1=obj1+"_4";
    this ->obj2=obj2*obj2*obj2*obj2;
}
void cl4::Print()
{
    cout<<obj1<<" ";
    cout<<obj2;
}
```

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 obj1;
        int obj2;
    public:
        cl4(string obj1, int obj2);
        void Print();
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"

int main(){
    string obj1;
    int obj2;
    cin>>obj1;
    cin>>obj2;
    cl1* object=(cl1*)new cl4(obj1, obj2);
    object->Print();
    cout<<endl;
    ((cl2*)object)->Print();
    cout<<endl;
    ((cl3*)object)->Print();
    cout<<endl;
    ((cl4*)object)->Print();
    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
Object 4	Object_1 4 Object_2 16 Object_3 64 Object_4 256	Object_1 4 Object_2 16 Object_3 64 Object_4 256
Object 17	Object_1 17 Object_2 289 Object_3 4913 Object_4 83521	Object_1 17 Object_2 289 Object_3 4913 Object_4 83521

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).