

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Постановка задачи.....	5
2 Метод решения.....	8
3 Описание алгоритма.....	11
4 Блок-схема алгоритма.....	12
5 Код программы.....	14
6 Тестирование.....	18
ЗАКЛЮЧЕНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	21

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:

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

которому в качестве аргументов передаются введенный идентификатор и натуральное число.

- 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

Объекты потока ввода/вывода cin, cout;

Объект класса class4;

Класс class1:

- Свойства/поля:
 - поле хранения названия объекта:
 - Наименование - name;
 - Тип - string;
 - Модификатор доступа - private;
 - поле хранения значения объекта:
 - Наименование - n;
 - Тип - int;
 - Модификатор доступа - private;
- Функционал:
 - Конструктор class1(string name, int n) устанавливает имя и целочисленное значение объекту;
 - Метод Print() выводит имя объекта и его значение.

Класс class2:

- Свойства/поля:
 - поле хранения названия объекта:
 - Наименование - name;
 - Тип - string;
 - Модификатор доступа - private;
 - поле хранения значения объекта:
 - Наименование - n;
 - Тип - int;

- Модификатор доступа - private;
- Функционал:
 - Конструктор `class2(string name, int n)` устанавливает имя и целочисленное значение объекту;
 - Метод `Print()` выводит имя объекта и его значение.

Класс `class3`:

- Свойства/поля:
 - поле хранения названия объекта:
 - Наименование - `name`;
 - Тип - `string`;
 - Модификатор доступа - `private`;
 - поле хранения значения объекта:
 - Наименование - `n`;
 - Тип - `int`;
 - Модификатор доступа - `private`;
- Функционал:
 - Конструктор `class3(string name, int n)` устанавливает имя и целочисленное значение объекту;
 - Метод `Print()` выводит имя объекта и его значение.

Класс `class4`:

- Свойства/поля:
 - поле хранения названия объекта:
 - Наименование - `name`;
 - Тип - `string`;
 - Модификатор доступа - `private`;
 - поле хранения значения объекта:
 - Наименование - `n`;

- Тип - int;
- Модификатор доступа - private;
- Функционал:
 - Конструктор class4(string name, int n) устанавливает имя и целочисленное значение объекту;
 - Метод Print() выводит имя объекта и его значение.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер	Комментарий
1	class1			Первый класс		
		class2	private		2	
2	class2			Второй класс		
		class3	private		3	
3	class3			Третий класс		
		class4	private		4	
4	class4			Четвертый класс		

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 class1

Функционал: определение значений закрытых свойств.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса class1

№	Предикат	Действия	№ перехода
1		Присвоить скрытому свойству name значение параметра name и дописать "_1"	2
2		Присвоить скрытому свойству n значение параметра n	∅

3.2 Алгоритм метода Print класса class1

Функционал: выводит имя объекта и его значение.

Параметры: .

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Print класса class1

№	Предикат	Действия	№ перехода
1		Вывод значений скрытых свойств name и n через пробел	∅

3.3 Алгоритм конструктора класса class2

Функционал: определение значений закрытых свойств.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса class2

№	Предикат	Действия	№ перехода
1		Присвоить скрытому свойству name значение параметра name и дописать "_2"	2
2		Присвоить скрытому свойству n значение параметра n*n	∅

3.4 Алгоритм метода Print класса class2

Функционал: выводит имя объекта и его значение.

Параметры: .

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Print класса class2

№	Предикат	Действия	№ перехода
1		Вывод значений скрытых свойств name и n через пробел	∅

3.5 Алгоритм конструктора класса class3

Функционал: определение значений закрытых свойств.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса class3

№	Предикат	Действия	№ перехода
1		Присвоить скрытому свойству name значение параметра name и дописать "_3"	2
2		Присвоить скрытому свойству n значение параметра n*n*n	∅

3.6 Алгоритм метода Print класса class3

Функционал: выводит имя объекта и его значение.

Параметры: .

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода Print класса class3

№	Предикат	Действия	№ перехода
1		Вывод значений скрытых свойств name и n через пробел	∅

3.7 Алгоритм конструктора класса class4

Функционал: определение значений закрытых свойств.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса class4

№	Предикат	Действия	№ перехода
1		Присвоить скрытому свойству name значение параметра name и дописать "_4"	2
2		Присвоить скрытому свойству n значение параметра n*n*n*n	∅

3.8 Алгоритм метода Print класса class4

Функционал: выводит имя объекта и его значение.

Параметры: .

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода Print класса class4

№	Предикат	Действия	№ перехода
1		Вывод значений скрытых свойств name и n через пробел	∅

3.9 Алгоритм функции main

Функционал: Основная программа.

Параметры: .

Возвращаемое значение: int - код возврата.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление строковой переменной name	2
2		Объявление целочисленной переменной n	3
3		Ввод значения переменной name	4
4		Ввод значения переменной n	5
5		Инициализация указателя obj на объект класса class1 адресом нового объекта класса class4 с помощью оператора функции new, с передачей конструктору параметров name, n, с приведением к указателю на объект класса class1	6
6		Вызов метода Print() объекта obj	7
7		Вывод переноса на новую строку	8

№	Предикат	Действия	№ перехода
8		Вызов метода Print() объекта obj, приведенный к указателю на объект класса class2	9
9		Вывод переноса на новую строку	10
10		Вызов метода Print() объекта obj, приведенный к указателю на объект класса class3	11
11		Вывод переноса на новую строку	12
12		Вызов метода Print() объекта obj, приведенный к указателю на объект класса class4	13
13		Освобождение памяти, выделенной под объект obj с помощью оператора функции delete	14
14		Возвратить 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

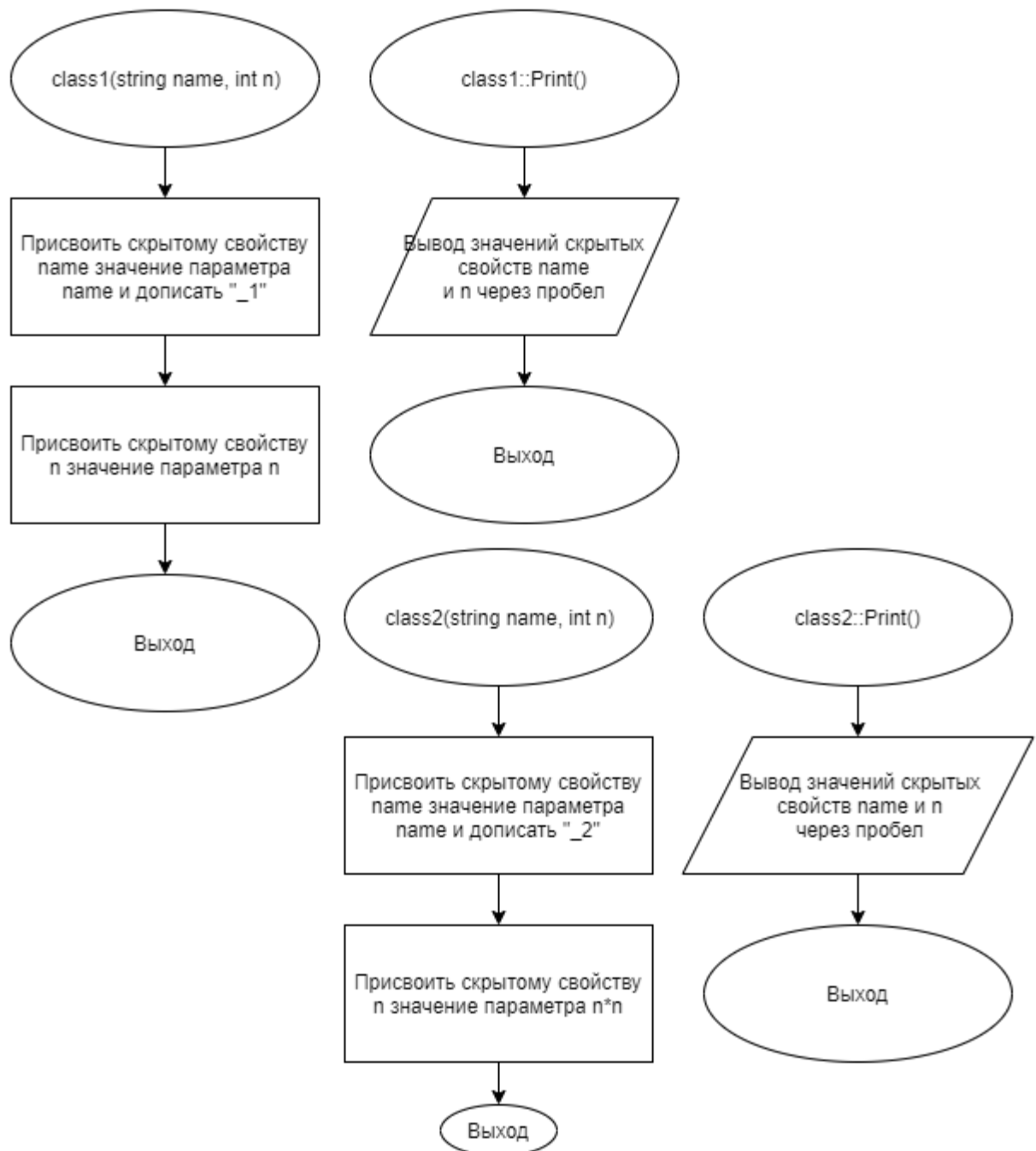


Рисунок 1 – Блок-схема алгоритма

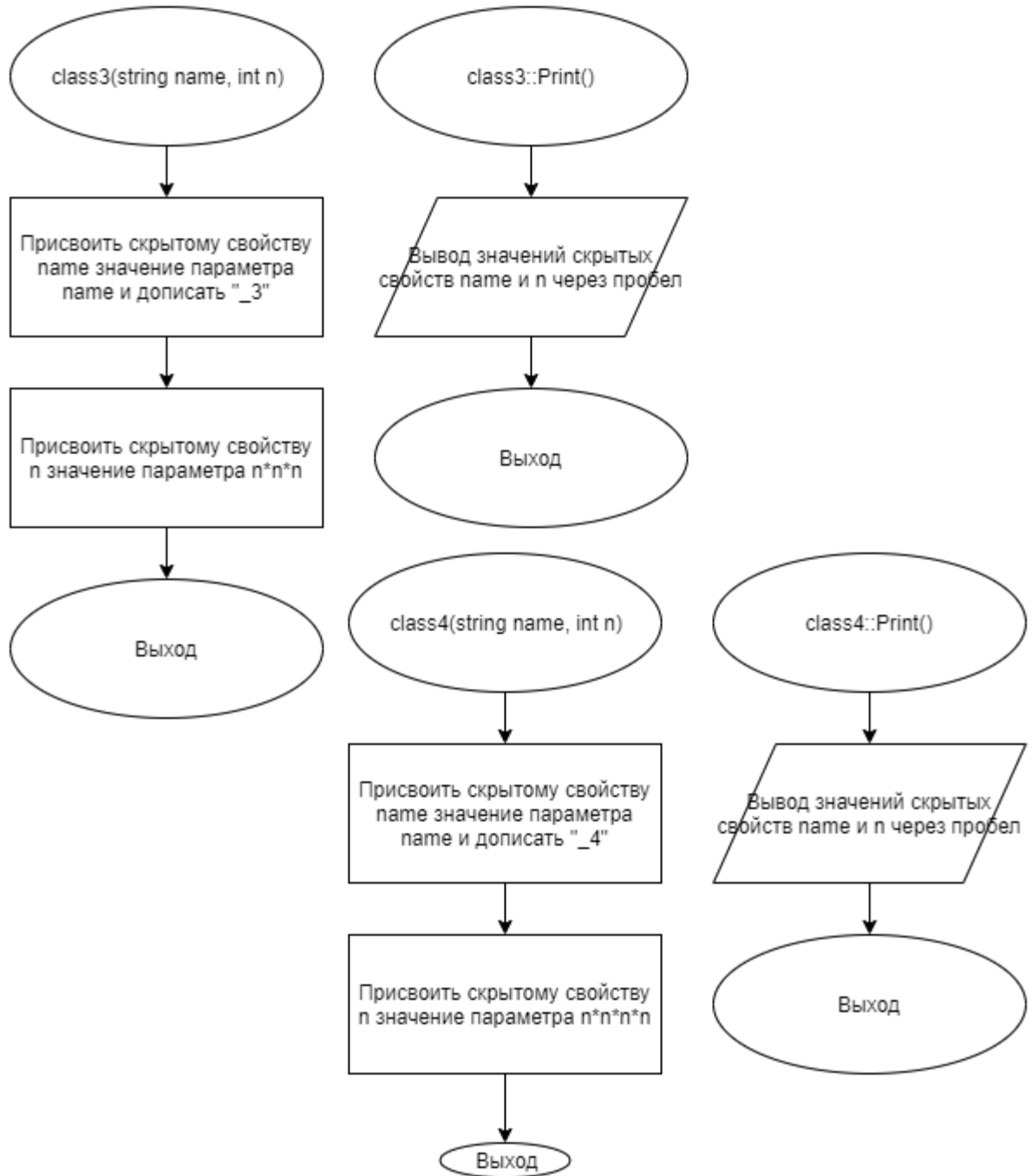


Рисунок 2 – Блок-схема алгоритма

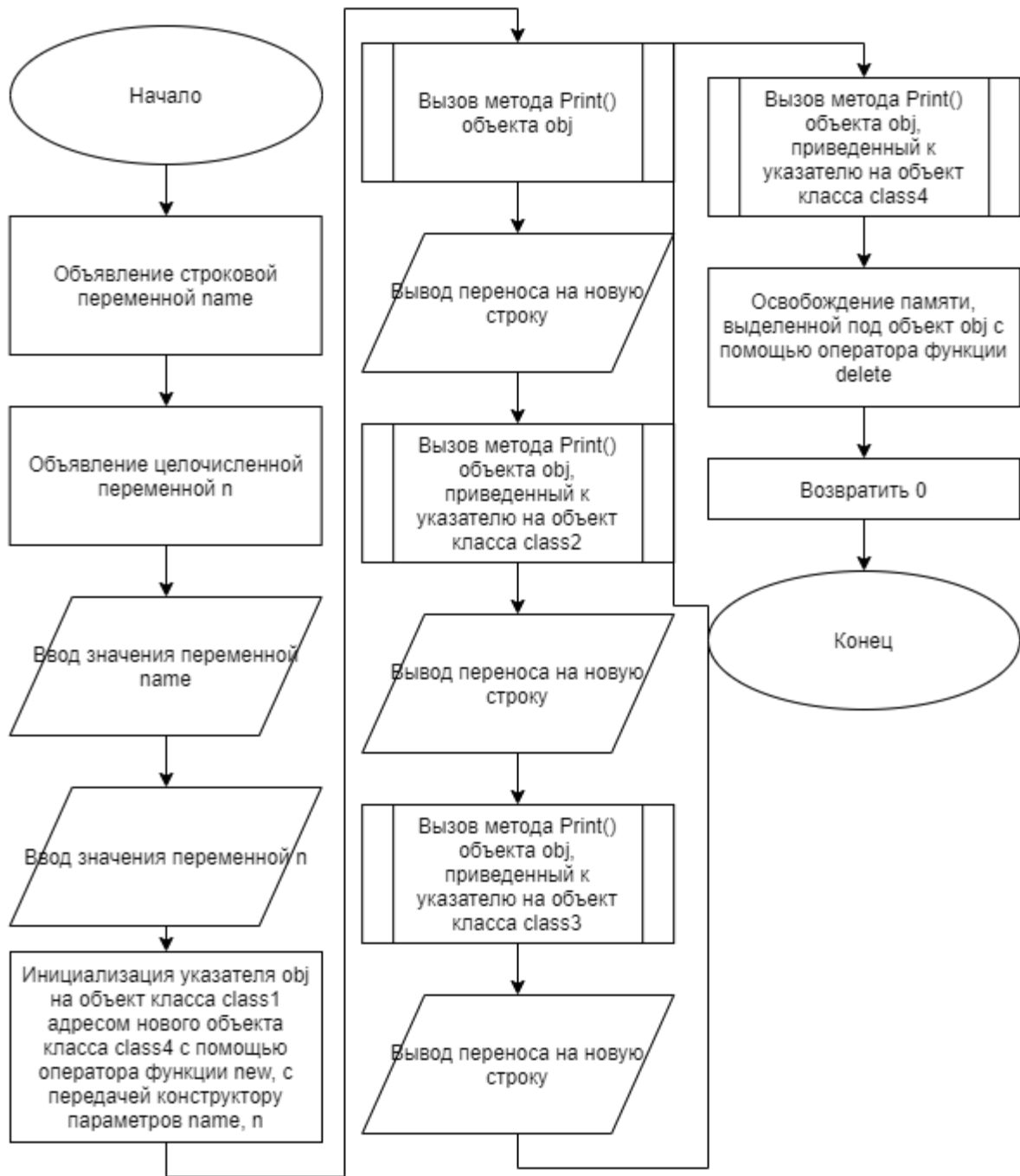


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл class1.cpp

Листинг 1 – class1.cpp

```
#include "class1.h"

class1::class1(string name, int n) {
    this->name = name + "_1";
    this->n = n;
}

void class1::Print() {
    cout << name << " ";
    cout << n;
}
```

5.2 Файл class1.h

Листинг 2 – class1.h

```
#ifndef __CLASS1__H
#define __CLASS1__H
#include <iostream>
using namespace std;

class class1 {
private:
    string name;
    int n;
public:
    class1(string name, int n);
    void Print();
};

#endif
```


5.3 Файл class2.cpp

Листинг 3 – class2.cpp

```
#include "class2.h"
#include "class1.h"

class2::class2(string name, int n):class1::class1(name, n) {
    this->name = name + "_2";
    this->n = n*n;
}
void class2::Print() {
    cout << name << " ";
    cout << n;
}
```

5.4 Файл class2.h

Листинг 4 – class2.h

```
#ifndef __CLASS2__H
#define __CLASS2__H
#include "class1.h"
#include <iostream>
using namespace std;

class class2:private class1 {
private:
    string name;
    int n;
public:
    class2(string name, int n);
    void Print();
};

#endif
```

5.5 Файл class3.cpp

Листинг 5 – class3.cpp

```
#include "class3.h"
#include "class2.h"

class3::class3(string name, int n):class2::class2(name, n) {
    this->name = name + "_3";
}
```

```

        this->n = n*n*n;
    }
    void class3::Print() {
        cout << name << " ";
        cout << n;
    }
}

```

5.6 Файл class3.h

Листинг 6 – class3.h

```

#ifndef __CLASS3__H
#define __CLASS3__H

#include "class2.h"
#include <iostream>
using namespace std;

class class3:private class2 {
private:
    string name;
    int n;
public:
    class3(string name, int n);
    void Print();
};

#endif

```

5.7 Файл class4.cpp

Листинг 7 – class4.cpp

```

#include "class4.h"
#include "class3.h"

class4::class4(string name, int n):class3::class3(name, n) {
    this->name = name + "_4";
    this->n = n*n*n*n;
}
void class4::Print() {
    cout << name << " ";
    cout << n;
}

```

5.8 Файл class4.h

Листинг 8 – class4.h

```
#ifndef __CLASS4__H
#define __CLASS4__H

#include "class3.h"
#include <iostream>
using namespace std;

class class4:private class3 {
private:
    string name;
    int n;
public:
    class4(string name, int n);
    void Print();
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include <iostream>
#include "class1.h"
#include "class2.h"
#include "class3.h"
#include "class4.h"
using namespace std;
int main() {
    string name;
    int n;
    cin >> name;
    cin >> n;

    class1* obj = (class1*)new class4(name, n);

    obj->Print();
    cout << endl;
    ((class2*)obj)->Print();
    cout << endl;
    ((class3*)obj)->Print();
    cout << endl;
    ((class4*)obj)->Print();
    delete obj;
    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

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на С++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodicheskoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
4. 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).
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).