Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Множественное наследование

Даны 8 классов, которые нумеруются от 1 до 8. Классы 2, 3, 4 и 5 наследованы от первого класса. Шестой класс от второго и третьего. Седьмой от четвертого и пятого. Восьмой от шестого и седьмого.

У каждого класса есть параметризированный конструктор с одним параметром строкового типа и закрытое свойство строкового типа для хранения наименования объекта класса. Значение данного свойства определяется в параметризированном конструкторе согласно шаблону:

```
«значение строкового параметра»_«номер класса»
```

У каждого класса есть метод в открытом разделе с одинаковым наименованием, который возвращает наименование объекта класса.

В реализации конструкторов со второго по восьмой класс, вызвать конструктор или конструкторы родительских классов. При вызове передать в качестве параметра выражение:

```
«параметр производного класса + «_» + «номер производного класса»
```

Например, для конструктора второго класса

```
cl_2 :: cl_2 ( string s_name ) : cl_1 ( s_name + “_2” )
```

В основной функции реализовать алгоритм:

1. Объявить один указатель на объект класса x.
2. Объявить переменную строкового типа.
3. Ввести значение строковой переменной. Вводимое значение является идентификатором.
4. Создать объект класса 8 посредством параметризированного конструктора, передав в качестве аргумента строковую переменную.

5. Адрес созданного объекта присвоить указателю на объект класса x.

6. Используя только указатель на объект класса x вывести имена всех объектов в составе объекта класса 8 и имя самого объекта класса 8. Вывод выполнить построчно, упорядочивая согласно возрастанию номеров класса. Наименования объектов первого класса вывести последовательно для производных объектов 2,3.4 и 5 класса.

Наследственность реализовать так, чтобы всего объектов было 10 и обеспечить вывод по аналогии приведенному примеру вывода.

1.1 Описание входных данных

Первая строка:

«идентификатор»

Пример ввода

```
Object
```

1.2 Описание выходных данных

Построчно (одиннадцать строк):

«наименование объекта»

Пример вывода:

```
Object_8_6_2_1
Object_8_6_3_1
Object_8_1
Object_8_1
Object_8_6_2
Object_8_6_3
Object_8_7_4
Object_8_7_5
Object_8_6
```

Object_8_7
Object_8

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект Object класса cl8 предназначен для ;
- функция main для главная функция программы;
- Объект стандартного потока ввода cin с клавиатуры;
- Объект стандартного потока вывода cout на экран;
- Библиотека iostream;
- Библиотека string.

Класс cl1:

- свойства/поля:
    - поле Хранение имя объекта:
        - наименование — objname;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl1 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
    - метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Класс cl2:

- свойства/поля:
    - поле Хранение имя объекта:
        - наименование — objname;
        - тип — string;
        - модификатор доступа — private;
- функционал:

- метод cl2 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
- метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Класс cl3:

- свойства/поля:
  - поле Хранение имя объекта:
    - наименование — objname;
    - тип — string;
    - модификатор доступа — private;
- функционал:
  - метод cl3 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
  - метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Класс cl4:

- свойства/поля:
  - поле Хранение имя объекта:
    - наименование — objname;
    - тип — string;
    - модификатор доступа — private;
- функционал:
  - метод cl4 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
  - метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Класс cl5:

- свойства/поля:
    - поле Хранение имя объекта:
        - наименование — objname;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl5 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
    - метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Класс cl6:

- свойства/поля:
    - поле Хранение имя объекта:
        - наименование — objname;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl6 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
    - метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Класс cl7:

- свойства/поля:
    - поле Хранение имя объекта:
        - наименование — objname;
        - тип — string;
        - модификатор доступа — private;

- функционал:
    - метод cl7 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
    - метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Класс cl8:

- свойства/поля:
    - поле Хранение имя объекта:
        - наименование — objname;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl8 — Параметризированный конструктор. Задаёт значение скрытому полю objname;
    - метод OutputName — Возврат значения скрытого поля objname (имени объекта).

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl1				
		cl2	public		2
		cl3	public		3
		cl4	public		4
		cl5	public		5
2	cl2				
		cl6	public		6
3	cl3				

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
		cl6	public		6
4	cl4				
		cl7	public		7
5	cl5				
		cl7	public		7
6	cl6				
		cl8	public		8
7	cl7				
		cl8	public		8
8	cl8				

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: Параметризированный конструктор. Задаёт значение скрытому полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного аргумента objname с добавлением к нему "_1"	∅

3.2 Алгоритм метода OutputName класса cl1

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода OutputName класса cl1

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.3 Алгоритм конструктора класса cl2

Функционал: Параметризированный конструктор. Задаёт значение скрытому полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl2

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного аргумента objname с добавлением к нему "_2"	∅

3.4 Алгоритм метода OutputName класса cl2

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода OutputName класса cl2

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.5 Алгоритм конструктора класса cl3

Функционал: Параметризированный конструктор. Задаёт значение скрытому полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса cl3

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного аргумента objname с добавлением к нему "_3"	∅

3.6 Алгоритм метода OutputName класса cl3

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода OutputName класса cl3

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.7 Алгоритм конструктора класса cl4

Функционал: Параметризированный конструктор. Задаёт значение скрытому полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса cl4

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного аргумента objname с добавлением к нему "_4"	∅

3.8 Алгоритм метода OutputName класса cl4

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода OutputName класса cl4

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.9 Алгоритм конструктора класса cl5

Функционал: Параметризированный конструктор. Задаёт значение скрытому полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 10.

Таблица 10 – Алгоритм конструктора класса cl5

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного аргумента objname с добавлением к нему "_5"	∅

3.10 Алгоритм метода OutputName класса cl5

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 11.

Таблица 11 – Алгоритм метода OutputName класса cl5

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.11 Алгоритм конструктора класса cl6

Функционал: Параметризированный конструктор. Задаёт значение скрытому полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 12.

Таблица 12 – Алгоритм конструктора класса cl6

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного аргумента objname с добавлением к нему "_6"	∅

3.12 Алгоритм метода OutputName класса cl6

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 13.

Таблица 13 – Алгоритм метода OutputName класса cl6

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.13 Алгоритм конструктора класса cl7

Функционал: Параметризированный конструктор. Задаёт значение скрытому

полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 14.

Таблица 14 – Алгоритм конструктора класса cl7

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного аргумента objname с добавлением к нему "_7"	∅

3.14 Алгоритм метода OutputName класса cl7

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 15.

Таблица 15 – Алгоритм метода OutputName класса cl7

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.15 Алгоритм конструктора класса cl8

Функционал: Параметризированный конструктор. Задаёт значение скрытому полю objname.

Параметры: нет.

Алгоритм конструктора представлен в таблице 16.

Таблица 16 – Алгоритм конструктора класса cl8

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю objname значения переданного	∅

№	Предикат	Действия	№ перехода
		аргумента objname с добавлением к нему "_8"	

3.16 Алгоритм метода OutputName класса cl8

Функционал: Возврат значения скрытого поля objname (имени объекта).

Параметры: нет.

Возвращаемое значение: переменная objname строкового типа.

Алгоритм метода представлен в таблице 17.

Таблица 17 – Алгоритм метода OutputName класса cl8

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля objname	∅

3.17 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 18.

Таблица 18 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление указателя ptrObject на cl8	2
2		Объявление переменной строкового типа objname	3
3		Ввод значения перемнной objname	4
4		Создание объекта Object класса cl8 посредством параметризированного конструктора	5
5		Присвоение указателю ptrObject ссылки на объект Object	6
6		Вывод: <<результат работы метода OutputName(), вызванного	7

№	Предикат	Действия	№ перехода
		посредством указателя ptrObject для объекта класса cl1, созданного объектом класса cl2>>	
7		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl1, созданного объектом класса cl3>>	8
8		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl1, созданного объектом класса cl4>>	9
9		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl1, созданного объектом класса cl5>>	10
10		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl2>>	11
11		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl3>>	12
12		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl4>>	13
13		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl5>>	14
14		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl6>>	15
15		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject для объекта класса cl7>>	16
16		Вывод: <<результат работы метода OutputName(), вызванного посредством указателя ptrObject>>	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

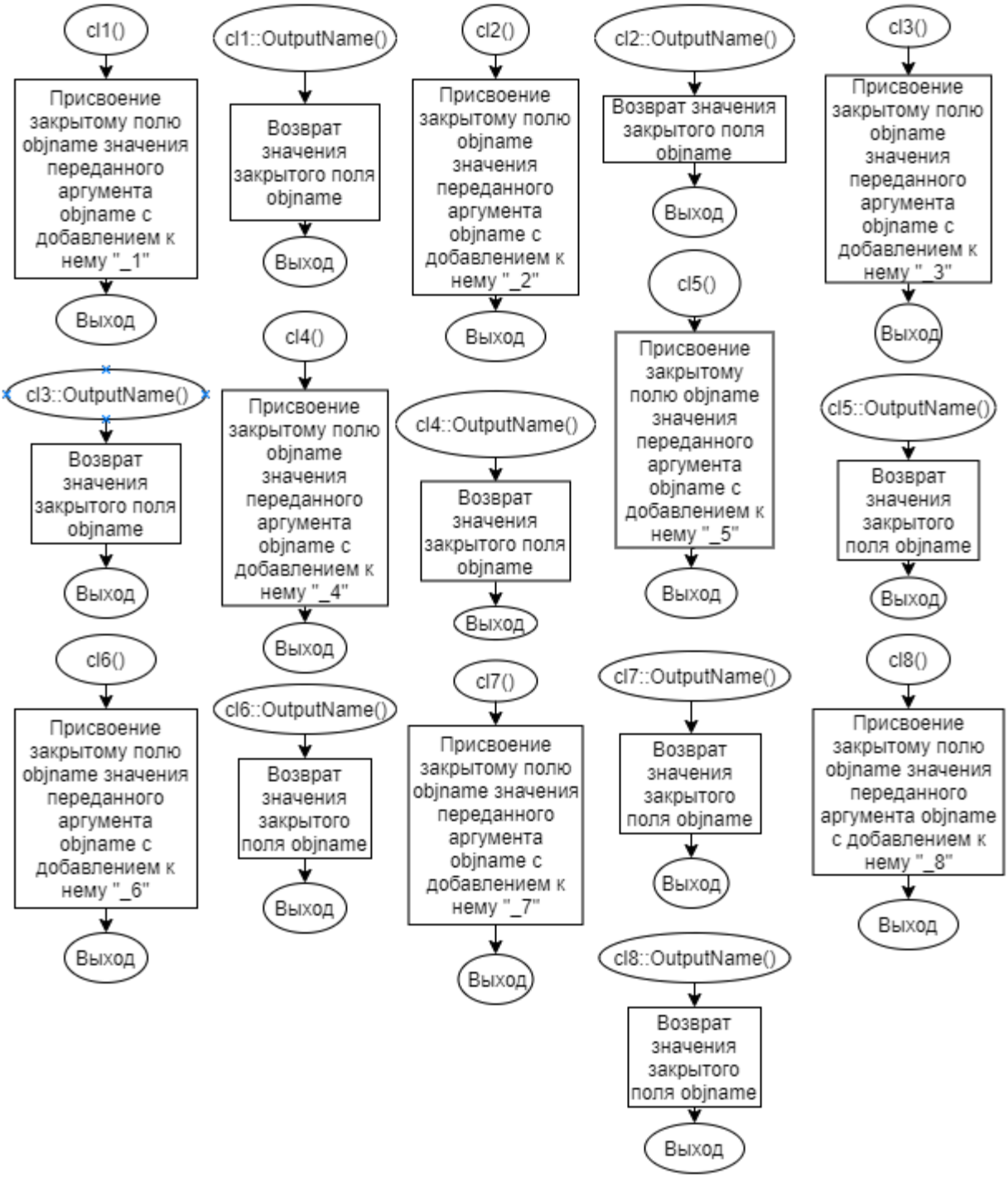

Рисунок 1 – Блок-схема алгоритма

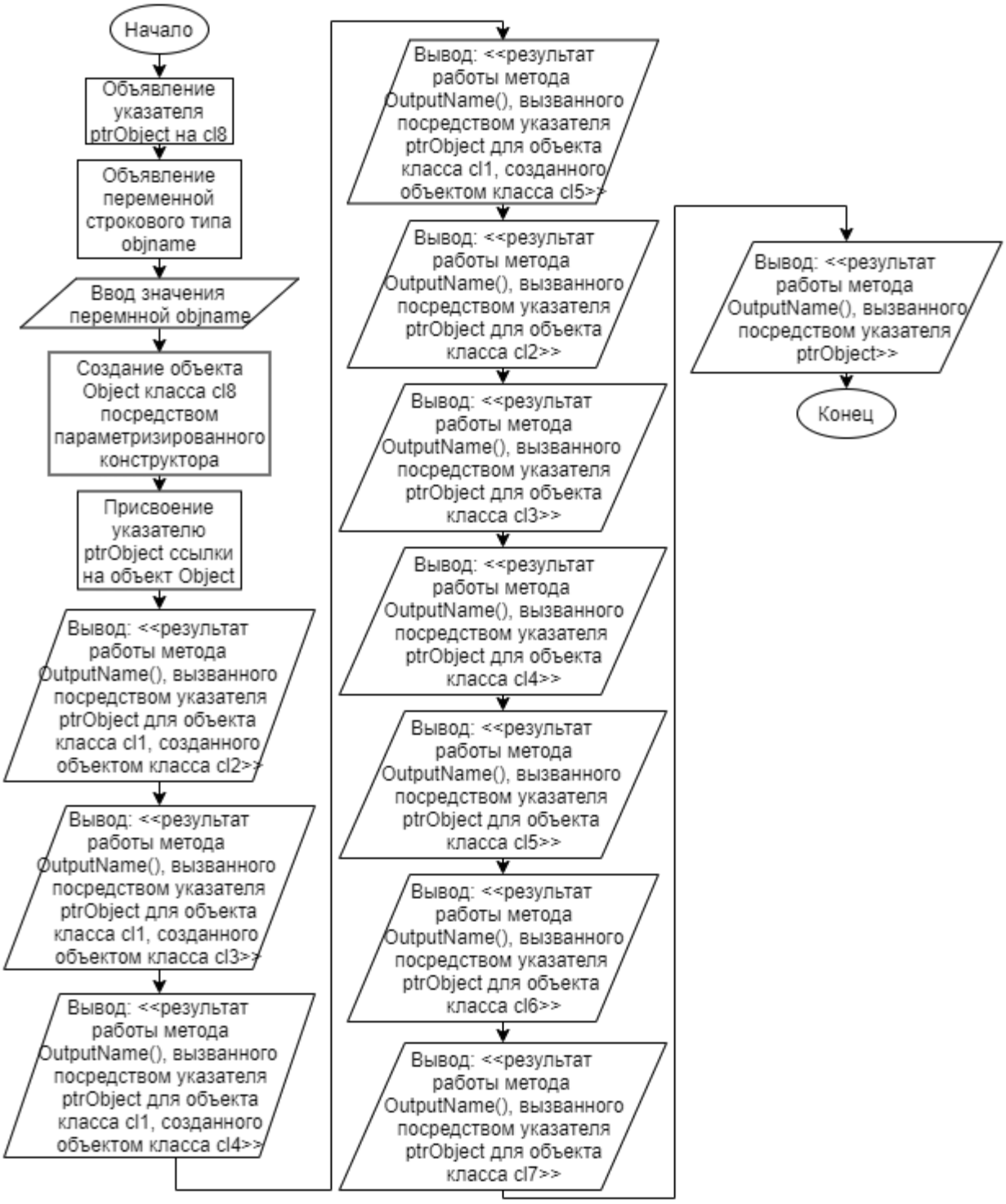

Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl1.cpp

Листинг 1 – cl1.cpp

```
#include "cl1.h"
cl1 :: cl1(string objname)
{
   this -> objname = objname + "_1";
}
string cl1 :: OutputName()
{
   return objname;
}
```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```
#ifndef __CL1__H
#define __CL1__H

#include <iostream>
#include <string>
using namespace std;

class cl1
{
   private:
   string objname;
   public:
   cl1(string objname);
   string OutputName();
};

#endif
```

5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

```
#include "cl2.h"
cl2 :: cl2(string objname) : cl1(objname + "_2")
{
   this -> objname = objname + "_2";
}
string cl2 :: OutputName()
{
   return objname;
}
```

5.4 Файл cl2.h

Листинг 4 – cl2.h

```
#ifndef __CL2__H
#define __CL2__H

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

class cl2 : public cl1
{
   private:
   string objname;
   public:
   cl2(string objname);
   string OutputName();
};


#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

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

```
cl3 :: cl3(string objname) : cl1(objname + "_3")
{
   this -> objname = objname + "_3";
}
string cl3 :: OutputName()
{
   return objname;
}
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3__H
#define __CL3__H

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

class cl3 : public cl1
{
   private:
   string objname;
   public:
   cl3(string objname);
   string OutputName();
};

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

```
#include "cl4.h"
cl4 :: cl4(string objname) : cl1(objname + "_4")
{
   this -> objname = objname + "_4";
}
string cl4 :: OutputName()
{
   return objname;
}
```

5.8 Файл cl4.h

Листинг 8 – cl4.h

```
#ifndef __CL4__H
#define __CL4__H

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

class cl4 : virtual public cl1
{
   private:
   string objname;
   public:
   cl4(string objname);
   string OutputName();
};

#endif
```

5.9 Файл cl5.cpp

Листинг 9 – cl5.cpp

```
#include "cl5.h"
cl5 :: cl5(string objname) : cl1(objname + "_5")
{
   this -> objname = objname + "_5";
}
string cl5 :: OutputName()
{
   return objname;
}
```

5.10 Файл cl5.h

Листинг 10 – cl5.h

```
#ifndef __CL5__H
#define __CL5__H

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

class cl5 : virtual public cl1
{
   private:
   string objname;
   public:
   cl5(string objname);
   string OutputName();
};

#endif
```

5.11 Файл cl6.cpp

Листинг 11 – cl6.cpp

```
#include "cl6.h"
cl6 :: cl6(string objname) : cl2(objname + "_6"), cl3(objname + "_6")
{
   this -> objname = objname + "_6";
}
string cl6 :: OutputName()
{
   return objname;
}
```

5.12 Файл cl6.h

Листинг 12 – cl6.h

```
#ifndef __CL6__H
#define __CL6__H
```

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

class cl6 :  public cl2, public cl3
{
   private:
   string objname;
   public:
   cl6(string objname);
   string OutputName();
};

#endif
```

5.13 Файл cl7.cpp

Листинг 13 – cl7.cpp

```
#include "cl7.h"
cl7 :: cl7(string objname) : cl1(objname + "_7"), cl4(objname + "_7"),
cl5(objname + "_7")
{
   this -> objname = objname + "_7";
}
string cl7 :: OutputName()
{
   return objname;
}
```

5.14 Файл cl7.h

Листинг 14 – cl7.h

```
#ifndef __CL7__H
#define __CL7__H

#include <iostream>
#include <string>
#include "cl4.h"
#include "cl5.h"
using namespace std;
```

```
class cl7 :  public cl4, public cl5
{
   private:
   string objname;
   public:
   cl7(string objname);
   string OutputName();
};


#endif
```

5.15 Файл cl8.cpp

Листинг 15 – cl8.cpp

```
#include "cl8.h"
cl8 :: cl8(string objname) : cl7 :: cl1(objname + "_8"), cl6(objname +
"_8"), cl7(objname + "_8")
{
   this -> objname = objname + "_8";
}
string cl8 :: OutputName()
{
   return objname;
}
```

5.16 Файл cl8.h

Листинг 16 – cl8.h

```
#ifndef __CL8__H
#define __CL8__H

#include <iostream>
#include <string>
#include "cl6.h"
#include "cl7.h"
using namespace std;

class cl8 :  public cl6, public cl7
{
   private:
   string objname;
```

```
   public:
   cl8(string objname);
   string OutputName();
};

#endif
```

5.17 Файл main.cpp

Листинг 17 – main.cpp

```
#include "cl8.h"
using namespace std;

int main()
{
   // program here
   cl8* ptrObject;
   string objname;
   cin >> objname;
   cl8 Object(objname);
   ptrObject = &Object;
   cout << ((cl1*)(cl2*)ptrObject) -> OutputName() << "\n";
   cout << ((cl1*)(cl3*)ptrObject) -> OutputName() << "\n";
   cout << ((cl1*)(cl4*)ptrObject) -> OutputName() << "\n";
   cout << ((cl1*)(cl5*)ptrObject) -> OutputName() << "\n";
   cout << ((cl2*)ptrObject) -> OutputName() << "\n";
   cout << ((cl3*)ptrObject) -> OutputName() << "\n";
   cout << ((cl4*)ptrObject) -> OutputName() << "\n";
   cout << ((cl5*)ptrObject) -> OutputName() << "\n";
   cout << ((cl6*)ptrObject) -> OutputName() << "\n";
   cout << ((cl7*)ptrObject) -> OutputName() << "\n";
   cout << ptrObject -> OutputName();
   return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 19.

Таблица 19 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные

`Object`	`Object_8_6_2_1`
	`Object_8_6_3_1`
	`Object_8_1`
	`Object_8_1`
	`Object_8_6_2`
	`Object_8_6_3`
	`Object_8_7_4`
	`Object_8_7_5`
	`Object_8_6`
	`Object_8_7`
	`Object_8`
		`Object_8_6_2_1`
		`Object_8_6_3_1`
		`Object_8_1`
		`Object_8_1`
		`Object_8_6_2`
		`Object_8_6_3`
		`Object_8_7_4`
		`Object_8_7_5`
		`Object_8_6`
		`Object_8_7`
		`Object_8`

`obj`	`obj_8_6_2_1`
	`obj_8_6_3_1`
	`obj_8_1`
	`obj_8_1`
	`obj_8_6_2`
	`obj_8_6_3`
	`obj_8_7_4`
	`obj_8_7_5`
	`obj_8_6`
	`obj_8_7`
	`obj_8`
		`obj_8_6_2_1`
		`obj_8_6_3_1`
		`obj_8_1`
		`obj_8_1`
		`obj_8_6_2`
		`obj_8_6_3`
		`obj_8_7_4`
		`obj_8_7_5`
		`obj_8_6`
		`obj_8_7`
		`obj_8`

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).