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:

- Объявить один указатель на объект класса x.
- Объявить переменную строкового типа.
- Ввести значение строковой переменной. Вводимое значение является идентификатором.
- Создать объект класса 8 посредством параметризированного конструктора, передав в качестве аргумента строковую переменную.
- Адрес созданного объекта присвоить указателю на объект класса x.
- 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 понадобится:

- объект потока ввода cin и объект потока вывода cout
- библиотека string
- указатель this
- 8 классов

Класс cl1 базовый класс

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl1(string s_name) - конструктор класса; void print() - вывод;

Класс cl2 наследует класс cl1

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl2(string s_name) - конструктор класса;void print() - вывод;

Класс cl3 наследует класс cl1

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl3(string s_name) - конструктор класса;void print() - вывод;

Класс cl4 виртуальный базовый класс cl1

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl4(string s_name) - конструктор класса;void print() - вывод;

Класс cl5 виртуальный базовый класс cl1

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl5(string s_name) - конструктор класса;void print() - вывод;

Класс cl6 наследует класс cl2, cl3

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl6(string s_name) - конструктор класса;void print() - вывод;

Класс cl7 наследует класс cl4, cl5, виртуальный базовый класс cl1

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl7(string s_name) - конструктор класса;void print() - вывод;

Класс cl8 наследует класс cl6, cl7, виртуальный базовый класс cl1

Поля:

скрытые элементы:

string s_name;

Методы:

открытые:

cl8(string s_name) - конструктор класса;void print() - вывод;

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 функции main

Функционал: главный метод программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление один указатель на объект класса 8	2
2		объявление переменную строкового типа	3
3		ввод значение	4
4		создание объекта класса 8	5
5		адрес созданного объекта присвоение указателю на объект класса 8	6
6		вывод построчно номеров класса	∅

3.2 Алгоритм конструктора класса cl1

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_1"	∅

3.3 Алгоритм метода print класса cl1

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода print класса cl1

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

3.4 Алгоритм конструктора класса cl2

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl2

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_2"	∅

3.5 Алгоритм метода print класса cl2

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода print класса cl2

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

3.6 Алгоритм конструктора класса cl3

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса cl3

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_3"	∅

3.7 Алгоритм метода print класса cl3

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода print класса cl3

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

3.8 Алгоритм конструктора класса cl4

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса cl4

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_4"	∅

3.9 Алгоритм метода print класса cl4

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода print класса cl4

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

3.10 Алгоритм конструктора класса cl5

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 10.

Таблица 10 – Алгоритм конструктора класса cl5

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_5"	∅

3.11 Алгоритм метода print класса cl5

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 11.

Таблица 11 – Алгоритм метода print класса cl5

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

3.12 Алгоритм конструктора класса cl6

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 12.

Таблица 12 – Алгоритм конструктора класса cl6

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_6"	∅

3.13 Алгоритм метода print класса cl6

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 13.

Таблица 13 – Алгоритм метода print класса cl6

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

3.14 Алгоритм конструктора класса cl7

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 14.

Таблица 14 – Алгоритм конструктора класса cl7

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_7"	∅

3.15 Алгоритм метода print класса cl7

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 15.

Таблица 15 – Алгоритм метода print класса cl7

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

3.16 Алгоритм конструктора класса cl8

Функционал: присваивание строкового параметра.

Параметры: string s_name.

Алгоритм конструктора представлен в таблице 16.

Таблица 16 – Алгоритм конструктора класса cl8

№	Предикат	Действия	№ перехода
1		this->s_name = s_name + "_8"	∅

3.17 Алгоритм метода print класса cl8

Функционал: вывод строки.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 17.

Таблица 17 – Алгоритм метода print класса cl8

№	Предикат	Действия	№ перехода
1		вывод s_name	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-6.

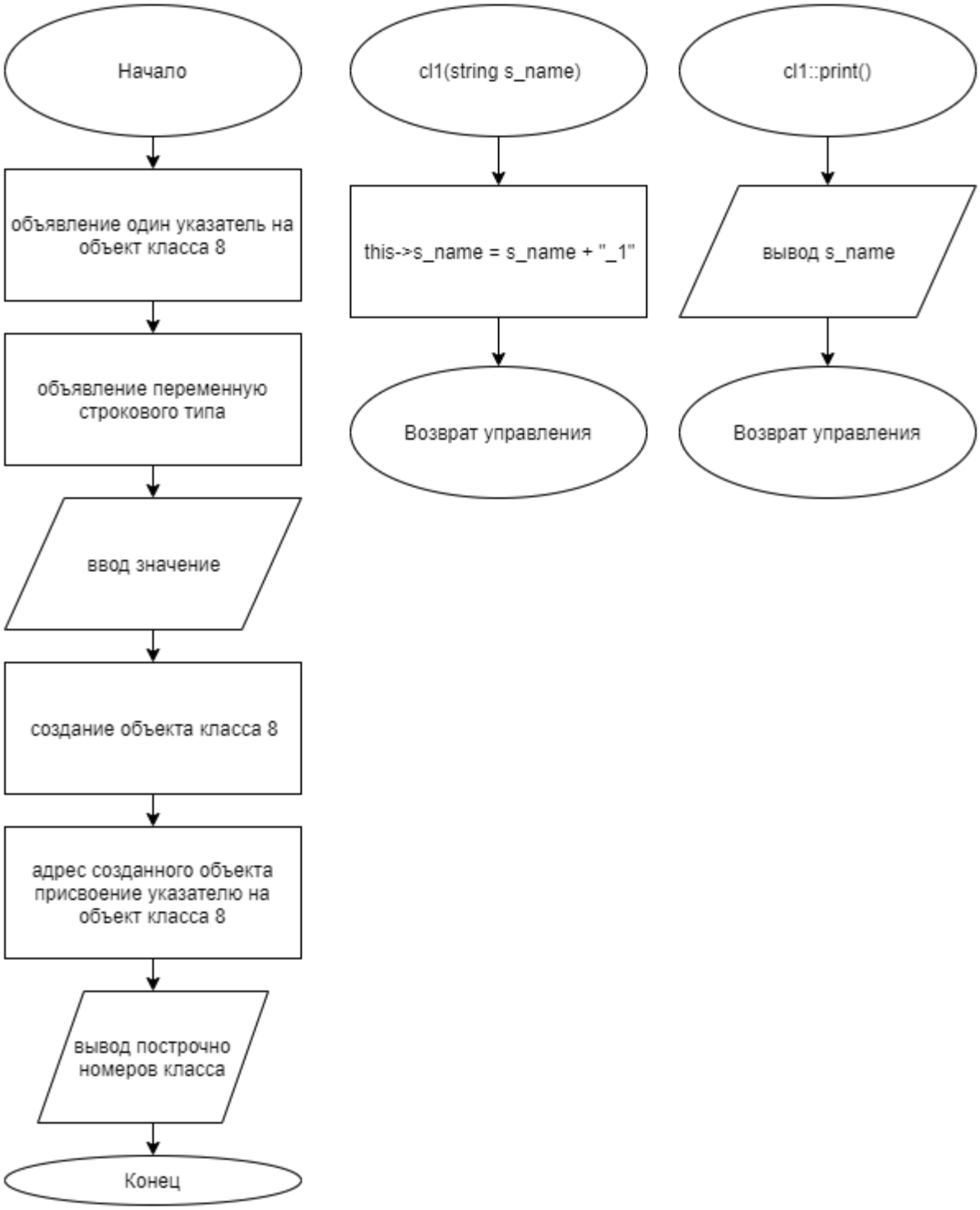

Рисунок 1 – Блок-схема алгоритма

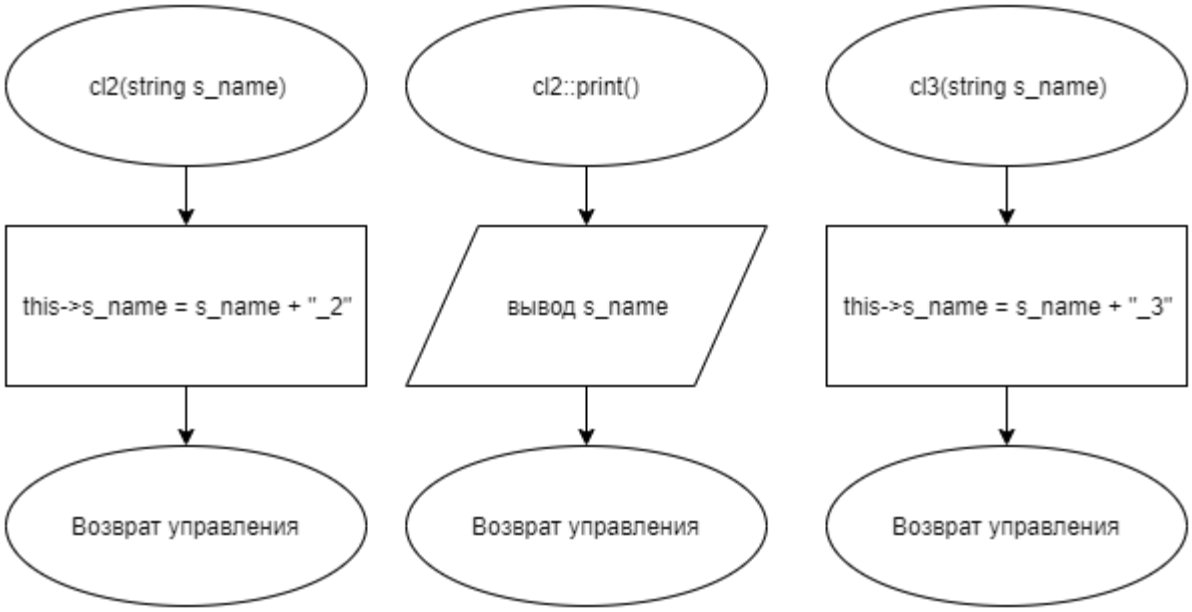

Рисунок 2 – Блок-схема алгоритма

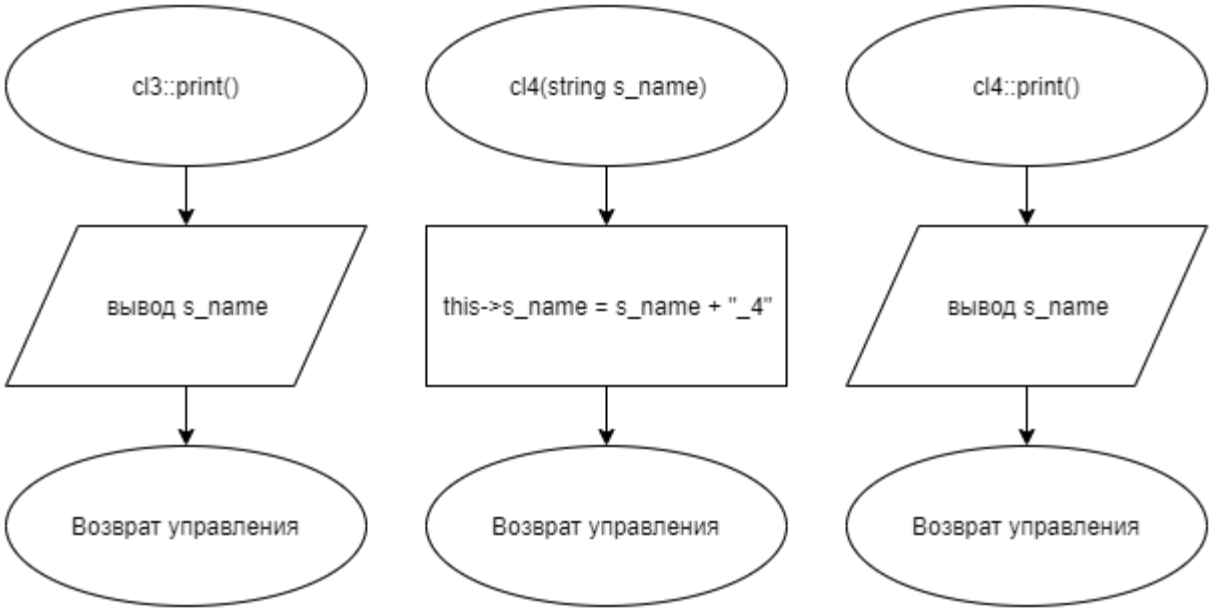

Рисунок 3 – Блок-схема алгоритма

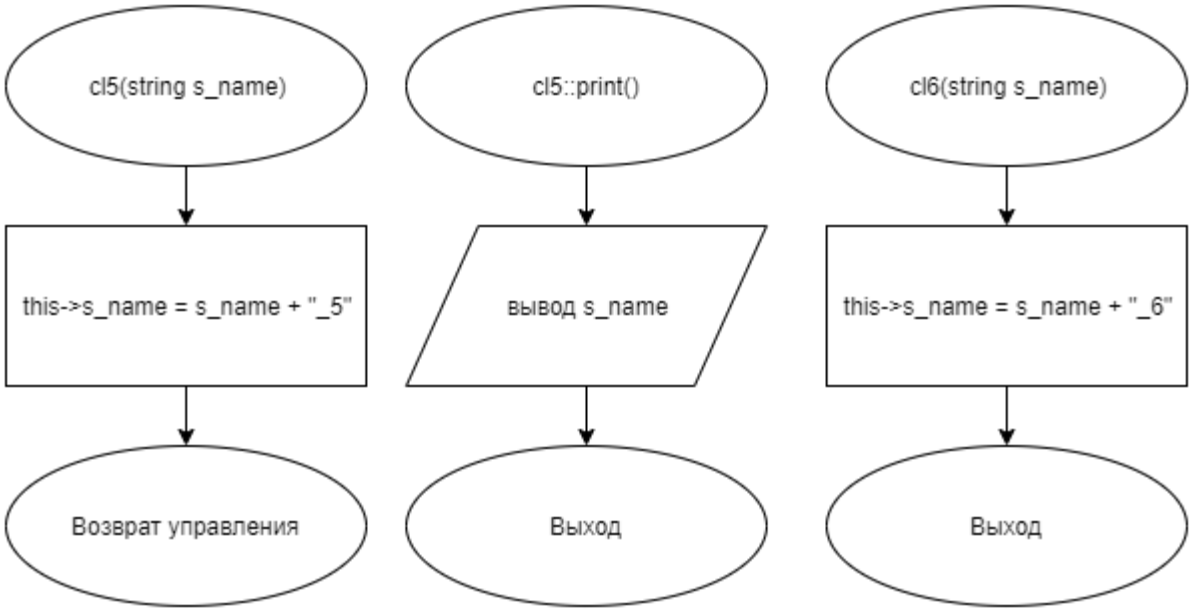

Рисунок 4 – Блок-схема алгоритма

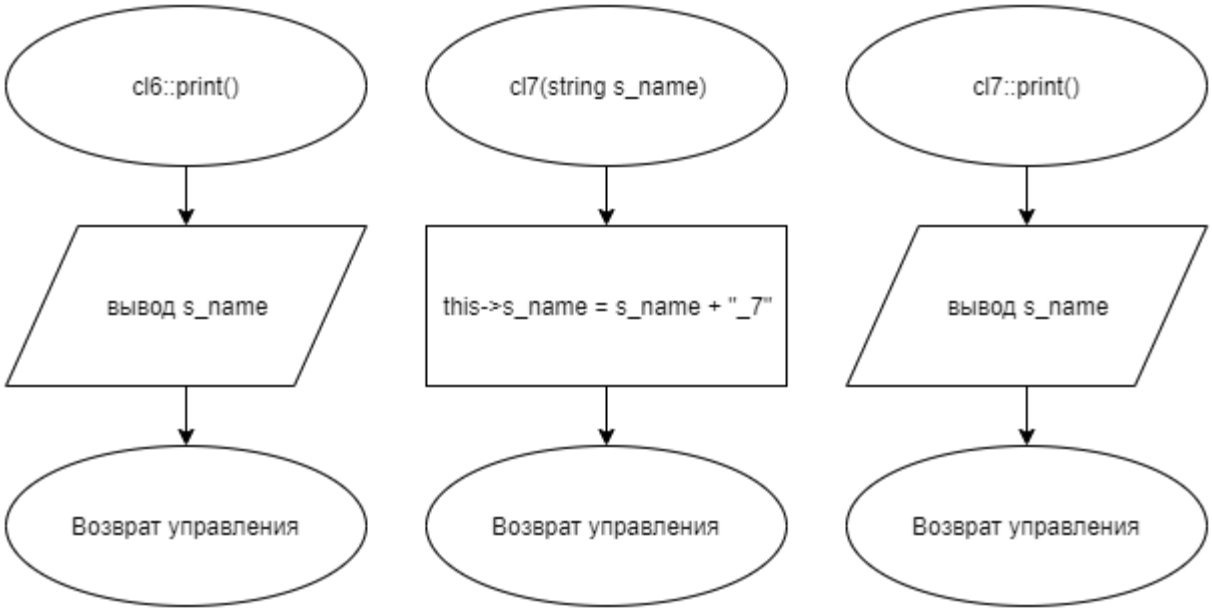

Рисунок 5 – Блок-схема алгоритма

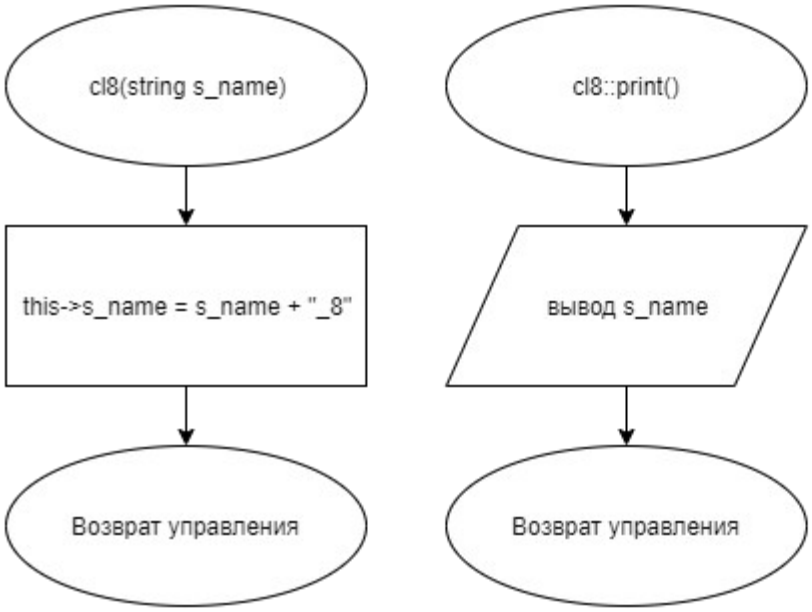

Рисунок 6 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl1.cpp

Листинг 1 – cl1.cpp

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

using namespace std;

cl1::cl1(string s_name)
{
	this->s_name = s_name + "_1";
}

void cl1::print()
{
	cout << s_name;
}
```

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 s_name;
public:
	cl1(string s_name);
	void print();
};

#endif
```

5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

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

using namespace std;

cl2::cl2(string s_name) : cl1(s_name + "_2")
{
	this->s_name = s_name + "_2";
}

void cl2::print()
{
	cout << s_name;
}
```

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 s_name;
public:
	cl2(string s_name);
	void print();
};

#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

```
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"

using namespace std;

cl3::cl3(string s_name) : cl1(s_name + "_3")
{
	this->s_name = s_name + "_3";
}

void cl3::print()
{
	cout << s_name;
}
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3__H
#define __CL3__H
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"

using namespace std;

class cl3 : public cl1
{
private:
	string s_name;
public:
	cl3(string s_name);
	void print();
};

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

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

using namespace std;

cl4::cl4(string s_name) : cl1(s_name + "_4")
{
	this->s_name = s_name + "_4";
}

void cl4::print()
{
	cout << s_name;
}
```

5.8 Файл cl4.h

Листинг 8 – cl4.h

```
#ifndef __CL4__H
#define __CL4__H
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"

using namespace std;

class cl4 : virtual public cl1
{
private:
	string s_name;
public:
	cl4(string s_name);
	void print();
};

#endif
```

5.9 Файл cl5.cpp

Листинг 9 – cl5.cpp

```
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"

using namespace std;

cl5::cl5(string s_name) : cl1(s_name + "_5")
{
	this->s_name = s_name + "_5";
}

void cl5::print()
{
	cout << s_name;
}
```

5.10 Файл cl5.h

Листинг 10 – cl5.h

```
#ifndef __CL5__H
#define __CL5__H
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"

using namespace std;

class cl5 : virtual public cl1
{
private:
	string s_name;
public:
	cl5(string s_name);
	void print();
};

#endif
```

5.11 Файл cl6.cpp

Листинг 11 – cl6.cpp

```
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"
#include "cl6.h"

using namespace std;

cl6::cl6(string s_name) : cl2(s_name + "_6"), cl3(s_name + "_6")
{
	this->s_name = s_name + "_6";
}

void cl6::print()
{
	cout << s_name;
}
```

5.12 Файл cl6.h

Листинг 12 – cl6.h

```
#ifndef __CL6__H
#define __CL6__H
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"

using namespace std;

class cl6 : public cl2, public cl3
{
private:
	string s_name;
public:
	cl6(string s_name);
	void print();
};

#endif
```

5.13 Файл cl7.cpp

Листинг 13 – cl7.cpp

```
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"
#include "cl6.h"
#include "cl7.h"

using namespace std;

cl7::cl7(string s_name) : cl4(s_name + "_7"), cl5(s_name + "_7"), cl1(s_name +
"_7")
{
	this->s_name = s_name + "_7";
}

void cl7::print()
{
	cout << s_name;
}
```

5.14 Файл cl7.h

Листинг 14 – cl7.h

```
#ifndef __CL7__H
#define __CL7__H
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"
#include "cl6.h"

using namespace std;

class cl7 : public cl4, public cl5, virtual public cl1
{
private:
	string s_name;
```

```
public:
      cl7(string s_name);
      void print();
};

#endif
```

5.15 Файл cl8.cpp

Листинг 15 – cl8.cpp

```
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"
#include "cl6.h"
#include "cl7.h"
#include "cl8.h"

using namespace std;

cl8::cl8(string s_name) : cl6(s_name + "_8"), cl7(s_name + "_8"), cl1(s_name +
"_8")
{
      this->s_name = s_name + "_8";
}

void cl8::print()
{
      cout << s_name;
}
```

5.16 Файл cl8.h

Листинг 16 – cl8.h

```
#ifndef __CL8__H
#define __CL8__H
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"
#include "cl6.h"
```

```
#include "cl7.h"

using namespace std;

class cl8 : public cl6, public cl7, virtual public cl1
{
private:
	string s_name;
public:
	cl8(string s_name);
	void print();
};

#endif
```

5.17 Файл main.cpp

Листинг 17 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <string>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"
#include "cl6.h"
#include "cl7.h"
#include "cl8.h"

using namespace std;

int main()
{
	cl8* x;
	string s;
	cin >> s;
	cl8 obj(s);
	x = &obj;
	((cl1*)(cl2*)x)->print();
	cout << endl;
	((cl1*)(cl3*)x)->print();
	cout << endl;
	((cl1*)(cl4*)x)->print();
	cout << endl;
	((cl1*)(cl5*)x)->print();
	cout << endl;
	((cl2*)x)->print();
	cout << endl;
	((cl3*)x)->print();
	cout << endl;
```

```
      ((cl4*)x)->print();
      cout << endl;
      ((cl5*)x)->print();
      cout << endl;
      ((cl6*)x)->print();
      cout << endl;
      ((cl7*)x)->print();
      cout << endl;
      x->print();
      return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 18.

Таблица 18 – 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

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. C++: 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/methodichescoe_posobie_dlya_laboratorny h_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).