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 основной объект программы;
- функция main для основная функция программы;
- cin - объект стандартного потока ввода с клавиатуры;
- cout - объект стандартного потока вывода на экран.

Класс cl1:

- свойства/поля:
    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl1 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Класс cl2:

- свойства/поля:
    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl2 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Класс cl3:

- свойства/поля:

    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl3 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Класс cl4:

- свойства/поля:
    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl4 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Класс cl5:

- свойства/поля:
    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl5 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Класс cl6:

- свойства/поля:

    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl6 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Класс cl7:

- свойства/поля:
    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl7 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Класс cl8:

- свойства/поля:
    - поле наименование объекта:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
- функционал:
    - метод cl8 — параметрический конструктор;
    - метод print — возвращает значение поля name.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl1			Первый класс	
		cl2	public		2
		cl3	public		3
		cl4	virtual public		4
		cl5	virtual public		5
2	cl2			Второй класс	
		cl6	public		6
3	cl3			Третий класс	
		cl6	public		6
4	cl4			Четвёртый класс	
		cl7	public		7
5	cl5			Пятый класс	
		cl7	public		7
6	cl6			Шестой класс	
		cl8	public		8
7	cl7			Седьмой класс	
		cl8	public		8
8	cl8			Восьмой 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 cl1

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_1"	∅

3.2 Алгоритм метода print класса cl1

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода print класса cl1

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.3 Алгоритм конструктора класса cl2

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl2

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_2" с помощью конструктора класса cl1 с параметром name+"_2"	∅

3.4 Алгоритм метода print класса cl2

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода print класса cl2

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.5 Алгоритм конструктора класса cl3

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса cl3

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_3" с помощью конструктора класса cl1 с параметром name+"_3"	∅

3.6 Алгоритм метода print класса cl3

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода print класса cl3

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.7 Алгоритм конструктора класса cl4

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса cl4

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_4" с помощью конструктора класса cl1 с параметром name+"_4"	∅

3.8 Алгоритм метода print класса cl4

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода print класса cl4

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.9 Алгоритм конструктора класса cl5

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 10.

Таблица 10 – Алгоритм конструктора класса cl5

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_5" с помощью конструктора класса cl1 с параметром name+"_5"	∅

3.10 Алгоритм метода print класса cl5

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 11.

Таблица 11 – Алгоритм метода print класса cl5

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.11 Алгоритм конструктора класса cl6

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 12.

Таблица 12 – Алгоритм конструктора класса cl6

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_6" с помощью конструктора класса cl2 с параметром name+"_6" и конструктора класса cl3 с параметром name+"_6"	∅

3.12 Алгоритм метода print класса cl6

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 13.

Таблица 13 – Алгоритм метода print класса cl6

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.13 Алгоритм конструктора класса cl7

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 14.

Таблица 14 – Алгоритм конструктора класса cl7

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_7" с помощью конструктора класса cl4 с параметром name+"_7", конструктора класса cl5 с параметром name+"_7" и конструктора класса cl1 с	∅

№	Предикат	Действия	№ перехода
		параметром name+"_7"	

3.14 Алгоритм метода print класса cl7

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 15.

Таблица 15 – Алгоритм метода print класса cl7

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.15 Алгоритм конструктора класса cl8

Функционал: параметрический конструктор.

Параметры: string name.

Алгоритм конструктора представлен в таблице 16.

Таблица 16 – Алгоритм конструктора класса cl8

№	Предикат	Действия	№ перехода
1		присваивание полю name значение аргумента name и "_8" с помощью конструктора класса cl6 с параметром name+"_8", конструктора класса cl7 с параметром name+"_8" и конструктора класса cl1 с параметром name+"_8"	∅

3.16 Алгоритм метода print класса cl8

Функционал: возвращает значение поля name.

Параметры: нет.

Возвращаемое значение: string.

Алгоритм метода представлен в таблице 17.

Таблица 17 – Алгоритм метода print класса cl8

№	Предикат	Действия	№ перехода
1		возврат поля name	∅

3.17 Алгоритм функции main

Функционал: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 18.

Таблица 18 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление указателя pointer на объект класса x	2
2		объявление переменной name типа string	3
3		ввод значения переменной name с клавиатуры	4
4		создание объекта object класса cl8 с помощью параметрического конструктора с аргументом name	5
5		присваивание указателю pointer адрес на объект object	6
6		вывод на экран: результат метода print() указателя pointer, приведённого к типу cl1 и cl2, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl1 и cl3, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl1 и cl4, переход на новус строку и результат метода print() указателя pointer,	7

№	Предикат	Действия	№ перехода
		приведённого к типу cl1 и cl5, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl2, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl3, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl4, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl5, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl6, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl7, переход на новус строку и результат метода print() указателя pointer, приведённого к типу cl8	
7		возврат значения 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

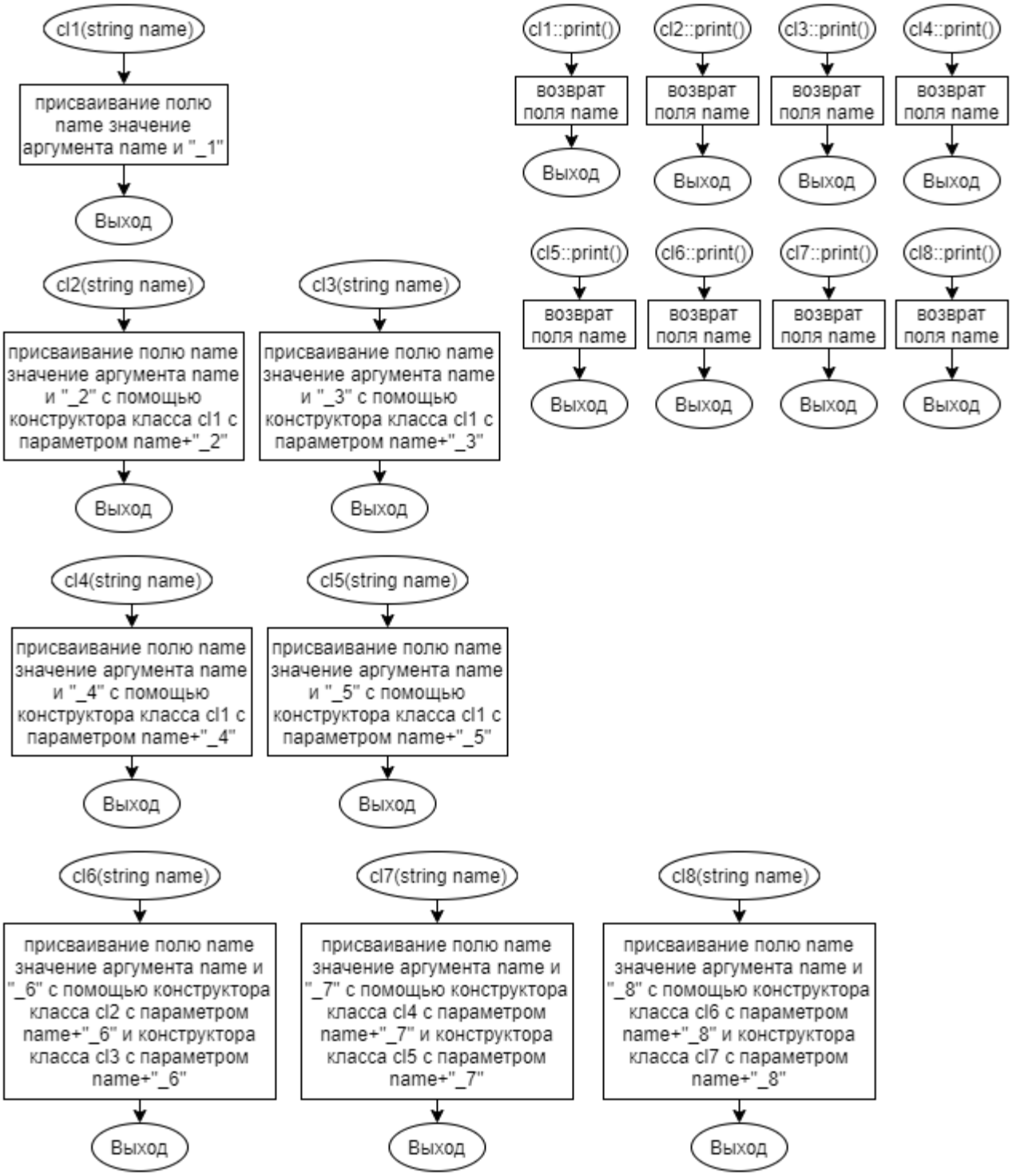

Рисунок 1 – Блок-схема алгоритма

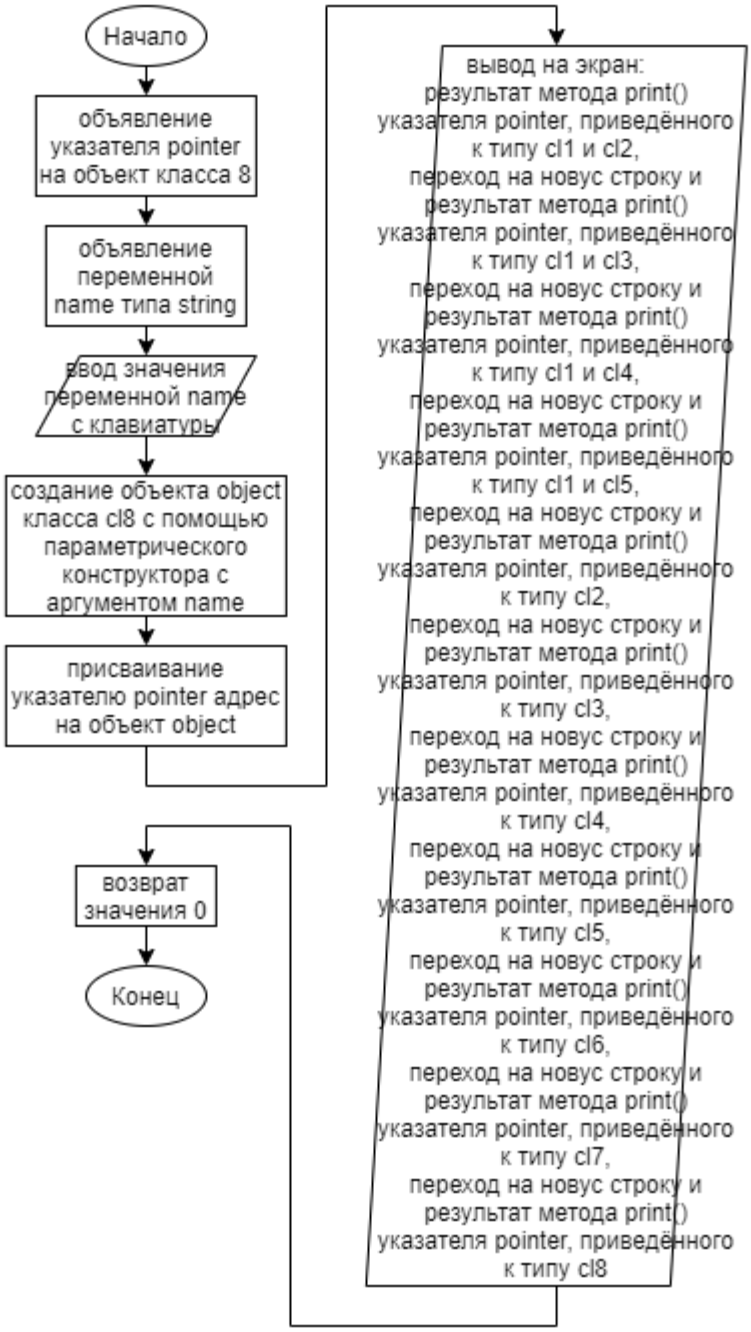

Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl1.cpp

Листинг 1 – cl1.cpp

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

cl1::cl1(std::string name)
{ this -> name = name + "_1"; }
std::string cl1::print()
{ return name; }
```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```
#ifndef __CL1__H
#define __CL1__H

#include <iostream>

class cl1
{
private:
   std::string name;
public:
   cl1(std::string name);
   std::string print();
};

#endif
```

5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

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

cl2::cl2(std::string name): cl1(name + "_2")
{ this -> name = name + "_2"; }
std::string cl2::print()
{ return name; }
```

5.4 Файл cl2.h

Листинг 4 – cl2.h

```
#ifndef __CL2__H
#define __CL2__H

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

class cl2: public cl1
{
private:
   std::string name;
public:
   cl2(std::string name);
   std::string print();
};

#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

```
#include "cl3.h"
#include <iostream>

cl3::cl3(std::string name): cl1(name + "_3")
{ this -> name = name + "_3"; }
std::string cl3::print()
```

```
{ return name; }
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3__H
#define __CL3__H

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

class cl3: public cl1
{
private:
   std::string name;
public:
   cl3(std::string name);
   std::string print();
};

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

```
#include "cl4.h"
#include <iostream>

cl4::cl4(std::string name): cl1(name + "_4")
{ this -> name = name + "_4"; }
std::string cl4::print()
{ return name; }
```

5.8 Файл cl4.h

Листинг 8 – cl4.h

```
#ifndef __CL4__H
```

```
#define __CL4__H

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

class cl4: virtual public cl1
{
private:
   std::string name;
public:
   cl4(std::string name);
   std::string print();
};

#endif
```

5.9 Файл cl5.cpp

Листинг 9 – cl5.cpp

```
#include "cl5.h"
#include <iostream>

cl5::cl5(std::string name): cl1(name + "_5")
{ this -> name = name + "_5"; }
std::string cl5::print()
{ return name; }
```

5.10 Файл cl5.h

Листинг 10 – cl5.h

```
#ifndef __CL5__H
#define __CL5__H

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

class cl5: virtual public cl1
{
private:
   std::string name;
public:
   cl5(std::string name);
```

```
   std::string print();
};

#endif
```

5.11 Файл cl6.cpp

Листинг 11 – cl6.cpp

```
#include "cl6.h"
#include <iostream>

cl6::cl6(std::string name): cl2(name + "_6"), cl3(name + "_6")
{ this -> name = name + "_6"; }
std::string cl6::print()
{ return name; }
```

5.12 Файл cl6.h

Листинг 12 – cl6.h

```
#ifndef __CL6__H
#define __CL6__H

#include <iostream>
#include "cl2.h"
#include "cl3.h"

class cl6: public cl2, public cl3
{
private:
   std::string name;
public:
   cl6(std::string name);
   std::string print();
};

#endif
```

5.13 Файл cl7.cpp

Листинг 13 – cl7.cpp

```
#include "cl7.h"
#include <iostream>

cl7::cl7(std::string name): cl4(name + "_7"), cl5(name + "_7"), cl1(name +
"_7")
{ this -> name = name + "_7"; }
std::string cl7::print()
{ return name; }
```

5.14 Файл cl7.h

Листинг 14 – cl7.h

```
#ifndef __CL7__H
#define __CL7__H

#include <iostream>
#include "cl4.h"
#include "cl5.h"

class cl7: public cl4, public cl5
{
private:
   std::string name;
public:
   cl7(std::string name);
   std::string print();
};

#endif
```

5.15 Файл cl8.cpp

Листинг 15 – cl8.cpp

```
#include "cl8.h"
#include <iostream>

cl8::cl8(std::string name): cl6(name + "_8"), cl7(name + "_8"), cl1(name +
```

```
"_8")
{ this -> name = name + "_8"; }
std::string cl8::print()
{ return name; }
```

5.16 Файл cl8.h

Листинг 16 – cl8.h

```
#ifndef __CL8__H
#define __CL8__H

#include <iostream>
#include "cl6.h"
#include "cl7.h"

class cl8: public cl6, public cl7
{
private:
   std::string name;
public:
   cl8(std::string name);
   std::string print();
};

#endif
```

5.17 Файл main.cpp

Листинг 17 – main.cpp

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

int main()
{
```

```
        cl8* pointer;
        std::string name;
        std::cin >> name;
        cl8 object(name);
        pointer = &object;
        std::cout << ((cl1*)(cl2*) pointer) -> print()
           << std::endl << ((cl1*)(cl3*) pointer) -> print()
           << std::endl << ((cl1*)(cl4*) pointer) -> print()
           << std::endl << ((cl1*)(cl5*) pointer) -> print()
           << std::endl << ((cl2*) pointer) -> print()
           << std::endl << ((cl3*) pointer) -> print()
           << std::endl << ((cl4*) pointer) -> print()
           << std::endl << ((cl5*) pointer) -> print()
           << std::endl << ((cl6*) pointer) -> print()
           << std::endl << ((cl7*) pointer) -> print()
           << std::endl << ((cl8*) pointer) -> print();
        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`

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).