

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.....	9
3 ОПИСАНИЕ АЛГОРИТМОВ.....	13
3.1 Алгоритм конструктора класса cl1.....	13
3.2 Алгоритм деструктора класса cl1.....	13
3.3 Алгоритм конструктора класса cl2.....	13
3.4 Алгоритм конструктора класса cl3.....	14
3.5 Алгоритм конструктора класса cl4.....	14
3.6 Алгоритм конструктора класса cl5.....	14
3.7 Алгоритм конструктора класса cl6.....	15
3.8 Алгоритм конструктора класса cl7.....	15
3.9 Алгоритм конструктора класса cl8.....	15
3.10 Алгоритм функции main.....	16
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	17
5 КОД ПРОГРАММЫ.....	22
5.1 Файл cl1.cpp.....	22
5.2 Файл cl1.h.....	22
5.3 Файл cl2.cpp.....	23
5.4 Файл cl2.h.....	23
5.5 Файл cl3.cpp.....	23
5.6 Файл cl3.h.....	24
5.7 Файл cl4.cpp.....	24
5.8 Файл cl4.h.....	24
5.9 Файл cl5.cpp.....	25

5.10 Файл cl5.h.....	25
5.11 Файл cl6.cpp.....	26
5.12 Файл cl6.h.....	26
5.13 Файл cl7.cpp.....	26
5.14 Файл cl7.h.....	27
5.15 Файл cl8.cpp.....	27
5.16 Файл cl8.h.....	27
5.17 Файл main.cpp.....	28
6 ТЕСТИРОВАНИЕ.....	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	30

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 параметризованный конструктор с одним параметром строкового типа и закрытое свойство строкового типа для хранения наименования объекта класса. Значение данного свойства определяется в параметризованном 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 параметризованного 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 `a` класса `cl_8` предназначен для адрес для указателя;
- объект `b` класса `cl_8*` предназначен для указатель на объект;
- функция `delete` для освобождение памяти;
- `cin` - объект стандартного потока ввода;
- `cout` - объект стандартного потока вывода.

Класс `cl1`:

- свойства/поля:
 - поле имя объекта:
 - наименование — `name`;
 - тип — `string`;
 - модификатор доступа — `private`;
- функционал:
 - метод `cl1` — Конструктор;
 - метод `~cl1` — Деструктор.

Класс `cl2`:

- свойства/поля:
 - поле имя объекта:
 - наименование — `name`;
 - тип — `string`;
 - модификатор доступа — `private`;
- функционал:
 - метод `cl2` — Конструктор.

Класс `cl3`:

- свойства/поля:

- о поле имя объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
- функционал:
 - о метод cl3 — конструктор.

Класс cl4:

- свойства/поля:
 - о поле имя объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
- функционал:
 - о метод cl4 — Конструктор.

Класс cl5:

- свойства/поля:
 - о поле имя объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
- функционал:
 - о метод cl5 — Конструктор.

Класс cl6:

- свойства/поля:
 - о поле имя объекта:
 - наименование — name;
 - тип — string;

- модификатор доступа — private;
- функционал:
 - о метод cl6 — Конструктор.

Класс cl7:

- свойства/поля:
 - о поле имя объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
- функционал:
 - о метод cl7 — Конструктор.

Класс cl8:

- свойства/поля:
 - о поле имя объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
- функционал:
 - о метод cl8 — Конструктор.

Таблица 1 – Иерархия наследования классов

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

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
2	cl2			Производный от cl1	
		cl6	public		6
3	cl3			Производный от cl1	
		cl6	public		6
4	cl4			Производный от cl1	
		cl7	public		7
5	cl5			Производный от cl1	
		cl7	public		7
6	cl6			производный от cl2 и cl3	
7	cl7			Производный от cl4 и cl5	
		cl8	public		8
8	cl8			Производный от cl7 и cl8	
		cl8	public		8

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.

Параметры: string name1- имя объекта.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.2 Алгоритм деструктора класса cl1

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса cl1

№	Предикат	Действия	№ перехода
1		Уничтожение объекта	Ø

3.3 Алгоритм конструктора класса cl2

Функционал: Конструктор.

Параметры: string name1 - имя объекта + "_1".

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса *cl2*

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.4 Алгоритм конструктора класса *cl3*

Функционал: конструктор.

Параметры: string name1 - имя объекта+"_1".

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса *cl3*

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.5 Алгоритм конструктора класса *cl4*

Функционал: Конструктор.

Параметры: string name1- имя объекта+"_1".

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса *cl4*

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.6 Алгоритм конструктора класса *cl5*

Функционал: Конструктор.

Параметры: string name1 - имя объекта+"_1".

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса *cl5*

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.7 Алгоритм конструктора класса *cl6*

Функционал: Конструктор.

Параметры: string name1 - имя объекта+"_2", + "_3".

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса *cl6*

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.8 Алгоритм конструктора класса *cl7*

Функционал: Конструктор.

Параметры: string name1 - имя объекта+"_4", + "_5", + "_1".

Алгоритм конструктора представлен в таблице 9.

Таблица 9 – Алгоритм конструктора класса *cl7*

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.9 Алгоритм конструктора класса *cl8*

Функционал: Конструктор.

Параметры: string name1 - имя объекта+"_6", + "_7", + "_1".

Алгоритм конструктора представлен в таблице 10.

Таблица 10 – Алгоритм конструктора класса *cl8*

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю name=name1	Ø

3.10 Алгоритм функции *main*

Функционал: Основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: *int* - индикатор корректности завершения алгоритма.

Алгоритм функции представлен в таблице 11.

Таблица 11 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		объявление <i>string name1</i>	2
2		ввод <i>name1</i> с клавиатуры	3
3		создание объекта <i>a</i> класса <i>cl8</i> с параметром <i>name1+"_1"</i>	4
4		присвоение указателю на класс <i>cl8</i> объекта <i>b</i> = адрес на объект <i>a</i>	5
5		последовательный вывод с новых строк наименований объектов каждого класса вниз по иерархии	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.



Рисунок 1 – Блок-схема алгоритма

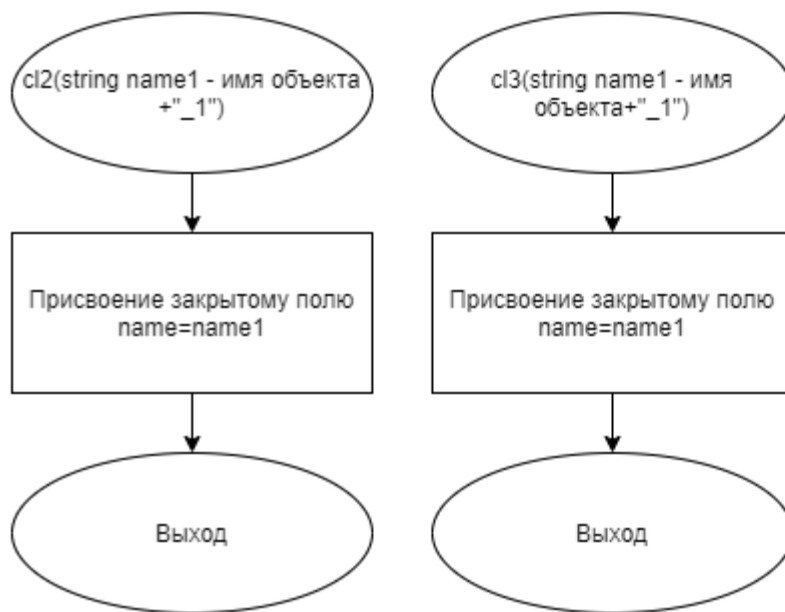


Рисунок 2 – Блок-схема алгоритма

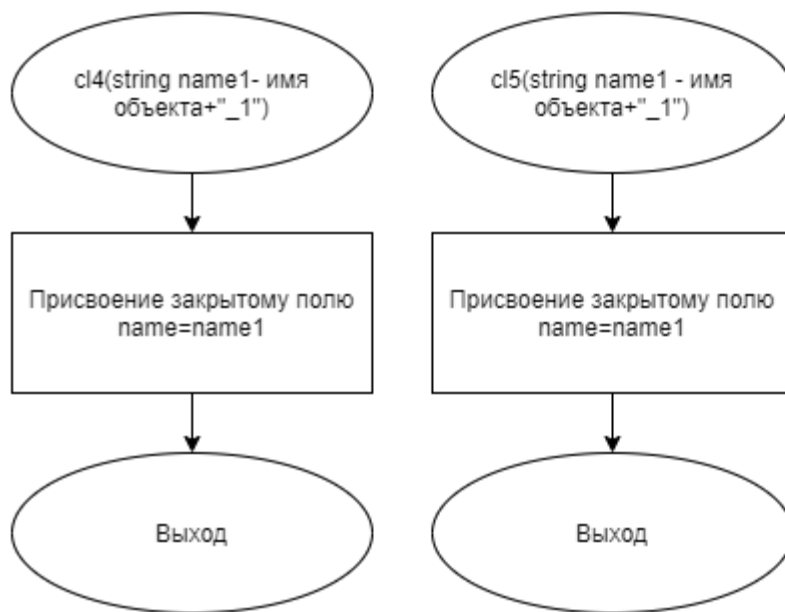


Рисунок 3 – Блок-схема алгоритма

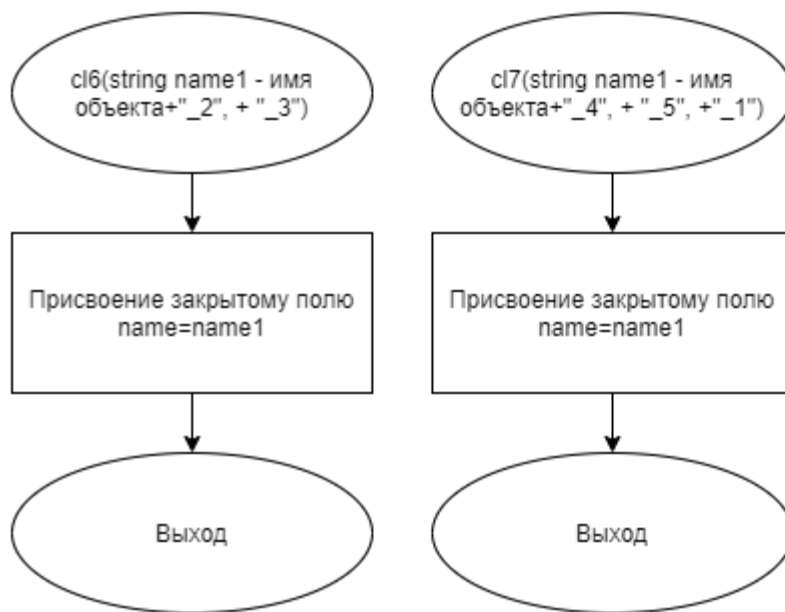


Рисунок 4 – Блок-схема алгоритма



Рисунок 5 – Блок-схема алгоритма

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 name1){
    name=name1;
}
string cl1::Out(){
    return name;
}
cl1::~~cl1(){}

```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```
#ifndef __CL1__H
#define __CL1__H
#include <iostream>
using namespace std;
class cl1{
private:
    string name;
public:
    cl1(string name1);
    string Out();
    ~cl1();
};
#endif

```

5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

```
#include "cl2.h"
cl2::cl2(string name1):cl1(name1+"_1"){
    name=name1;
}
string cl2::Out(){
    return name;
}
```

5.4 Файл cl2.h

Листинг 4 – cl2.h

```
#ifndef __CL2__H
#define __CL2__H
#include <iostream>
#include "cl1.h"
using namespace std;
class cl2:public cl1{
    private:
        string name;
    public:
        cl2(string name1);
        string Out();
};
#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

```
#include "cl3.h"
cl3::cl3(string name1):cl1(name1+"_1"){
    name=name1;
}
string cl3::Out(){
    return name;
}
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3__H
#define __CL3__H
#include "cl1.h"
using namespace std;
class cl3:public cl1{
private:
    string name;
public:
    cl3(string name1);
    string Out();
};

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

```
#include "cl4.h"
cl4::cl4(string name1):cl1(name1+"_1"){
    name=name1;
}
string cl4::Out(){
    return name;
}
```

5.8 Файл cl4.h

Листинг 8 – cl4.h

```
#ifndef __CL4__H
#define __CL4__H
#include "cl1.h"
using namespace std;
class cl4:virtual public cl1{
private:
    string name;
public:
```

```
        cl4(string name1);  
        string Out();  
};  
  
#endif
```

5.9 Файл cl5.cpp

Листинг 9 – cl5.cpp

```
#include "cl5.h"  
cl5::cl5(string name1):cl1(name1+"_1"){  
    name=name1;  
}  
string cl5::Out(){  
    return name;  
}
```

5.10 Файл cl5.h

Листинг 10 – cl5.h

```
#ifndef __CL5__H  
#define __CL5__H  
#include "cl1.h"  
using namespace std;  
class cl5:virtual public cl1{  
    private:  
        string name;  
    public:  
        cl5(string name1);  
        string Out();  
};  
  
#endif
```

5.11 Файл cl6.cpp

Листинг 11 – cl6.cpp

```
#include "cl6.h"
cl6::cl6(string name1):cl2(name1+"_2"),cl3(name1+"_3"){
    name=name1;
}
string cl6::Out(){
    return name;
}
```

5.12 Файл cl6.h

Листинг 12 – cl6.h

```
#ifndef __CL6__H
#define __CL6__H
#include "cl2.h"
#include "cl3.h"
using namespace std;
class cl6:public cl2, public cl3{
    private:
        string name;
    public:
        cl6(string name1);
        string Out();
};

#endif
```

5.13 Файл cl7.cpp

Листинг 13 – cl7.cpp

```
#include "cl7.h"
cl7::cl7(string name1):cl4(name1+"_4"),cl5(name1+"_5"),cl1(name1+"_1"){
    name=name1;
}
string cl7::Out(){
    return name;
}
```


5.14 Файл cl7.h

Листинг 14 – cl7.h

```
#ifndef __CL7__H
#define __CL7__H
#include "cl4.h"
#include "cl5.h"
using namespace std;
class cl7:public cl4, public cl5{
private:
    string name;
public:
    cl7(string name1);
    string Out();
};

#endif
```

5.15 Файл cl8.cpp

Листинг 15 – cl8.cpp

```
#include "cl8.h"
cl8::cl8(string name1):cl6(name1+"_6"),cl7(name1+"_7"),cl1(name1+"_1"){
    name=name1;
}
string cl8::Out(){
    return name;
}
```

5.16 Файл cl8.h

Листинг 16 – cl8.h

```
#ifndef __CL8__H
#define __CL8__H
#include "cl6.h"
```

```

#include "cl7.h"
using namespace std;
class cl8:public cl6, public cl7{
    private:
        string name;
    public:
        cl8(string name1);
        string Out();
};

#endif

```

5.17 Файл main.cpp

Листинг 17 – main.cpp

```

#include <stdlib.h>
#include <stdio.h>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"
#include "cl5.h"
#include "cl6.h"
#include "cl7.h"
#include "cl8.h"
#include <iostream>
using namespace std;
int main()
{
    string name1;
    cin>>name1;
    cl8 a(name1+"_8");
    cl8 * b = &a;
    cout<<((cl1*)((cl2*)(b)))->Out()<<endl;
    cout<<((cl1*)((cl3*)(b)))->Out()<<endl;
    cout<<((cl1*)((cl4*)(b)))->Out()<<endl;
    cout<<((cl1*)((cl5*)(b)))->Out()<<endl;
    cout<<((cl2*)((cl6*)(b)))->Out()<<endl;
    cout<<((cl3*)((cl6*)(b)))->Out()<<endl;
    cout<<((cl4*)((cl7*)(b)))->Out()<<endl;
    cout<<((cl5*)((cl7*)(b)))->Out()<<endl;
    cout<<((cl6*)(b))->Out()<<endl;
    cout<<((cl7*)(b))->Out()<<endl;
    cout<<b->Out()<<endl;
    return(0);
}

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 12.

Таблица 12 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
ob	ob_8_6_2_1 ob_8_6_3_1 ob_8_1 ob_8_1 ob_8_6_2 ob_8_6_3 ob_8_7_4 ob_8_7_5 ob_8_6 ob_8_7 ob_8	ob_8_6_2_1 ob_8_6_3_1 ob_8_1 ob_8_1 ob_8_6_2 ob_8_6_3 ob_8_7_4 ob_8_7_5 ob_8_6 ob_8_7 ob_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-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).