Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

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 вызвать 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 параметризированный 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:

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

которому в качестве аргументов передаются введенный идентификатор и натуральное число.

3. 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

Для решения задачи используется:

- 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 класса:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле целочисленное значение класса:
        - наименование — value;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод cl1 — параметризированный конструктор;
    - метод print — вывод закрытых полей на экран.

Класс cl2:

- свойства/поля:
    - поле наименование класса:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле целочисленное значение класса:
        - наименование — value;

        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод cl2 — параметризированный конструктор;
    - метод print — вывод закрытых полей на экран.

Класс cl3:

- свойства/поля:
    - поле наименование класса:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле целочисленное значение класса:
        - наименование — value;
        - тип — int;
        - модификатор доступа — private;
- функционал:
    - метод cl3 — параметризированный конструктор;
    - метод print — вывод закрытых полей на экран.

Класс cl4:

- свойства/поля:
    - поле наименование класса:
        - наименование — name;
        - тип — string;
        - модификатор доступа — private;
    - поле целочисленное значение класса:
        - наименование — value;
        - тип — int;

        - модификатор доступа — private;
- функционал:
    - метод cl4 — параметризированный конструктор;
    - метод print — вывод закрытых полей на экран.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl1			Первый класс	
		cl2	private		2
2	cl2			Второй класс	
		cl3	private		3
3	cl3			Третий класс	
		cl4	private		4
4	cl4			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 cl1

Функционал: параметризированный конструктор.

Параметры: string name, int value.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		присваивание полю name значение параметра name и _1	2
2		присваивание полю value значение параметра value в 1 степени	∅

3.2 Алгоритм метода print класса cl1

Функционал: вывод закрытых полей на экран.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода print класса cl1

№	Предикат	Действия	№ перехода
1		вывод на экран значений закрытых полей объекта через пробел и переход на новую строку	∅

3.3 Алгоритм конструктора класса cl2

Функционал: параметризированный конструктор.

Параметры: string name, int value.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl2

№	Предикат	Действия	№ перехода
1		присваивание полю name значение параметра name и _2	2
2		присваивание полю value значение параметра value во 2 степени	∅

3.4 Алгоритм метода print класса cl2

Функционал: вывод закрытых полей на экран.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода print класса cl2

№	Предикат	Действия	№ перехода
1		вывод на экран значений закрытых полей объекта через пробел и переход на новую строку	∅

3.5 Алгоритм конструктора класса cl3

Функционал: параметризированный конструктор.

Параметры: string name, int value.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса cl3

№	Предикат	Действия	№ перехода
1		присваивание полю name значение параметра name и _3	2
2		присваивание полю value значение параметра value в 3 степени	∅

3.6 Алгоритм метода print класса cl3

Функционал: вывод закрытых полей на экран.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода print класса cl3

№	Предикат	Действия	№ перехода
1		вывод на экран значений закрытых полей объекта через пробел и переход на новую строку	∅

3.7 Алгоритм конструктора класса cl4

Функционал: параметризированный конструктор.

Параметры: string name, int value.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса cl4

№	Предикат	Действия	№ перехода
1		присваивание полю name значение параметра name и _4	2
2		присваивание полю value значение параметра value в 4 степени	∅

3.8 Алгоритм метода print класса cl4

Функционал: вывод закрытых полей на экран.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода print класса cl4

№	Предикат	Действия	№ перехода
1		вывод на экран значений закрытых полей объекта через пробел и переход на новую строку	∅

3.9 Алгоритм функции main

Функционал: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		инициализация переменных name типа string и value типа int	2
2		ввод значений name и value с клавиатуры	3
3		создание указателя object на класс cl1 с помощьй параметризированного конструктора класса cl4	4
4		вызов метода print() указателя object	5
5		вызов метода print() указателя object, переделанного под указатель на класс cl2	6
6		вызов метода print() указателя object, переделанного под указатель на класс cl3	7

№	Предикат	Действия	№ перехода
7		вызов метода print() указателя object, переделанного под указатель на класс cl4	8
8		возврат значения 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

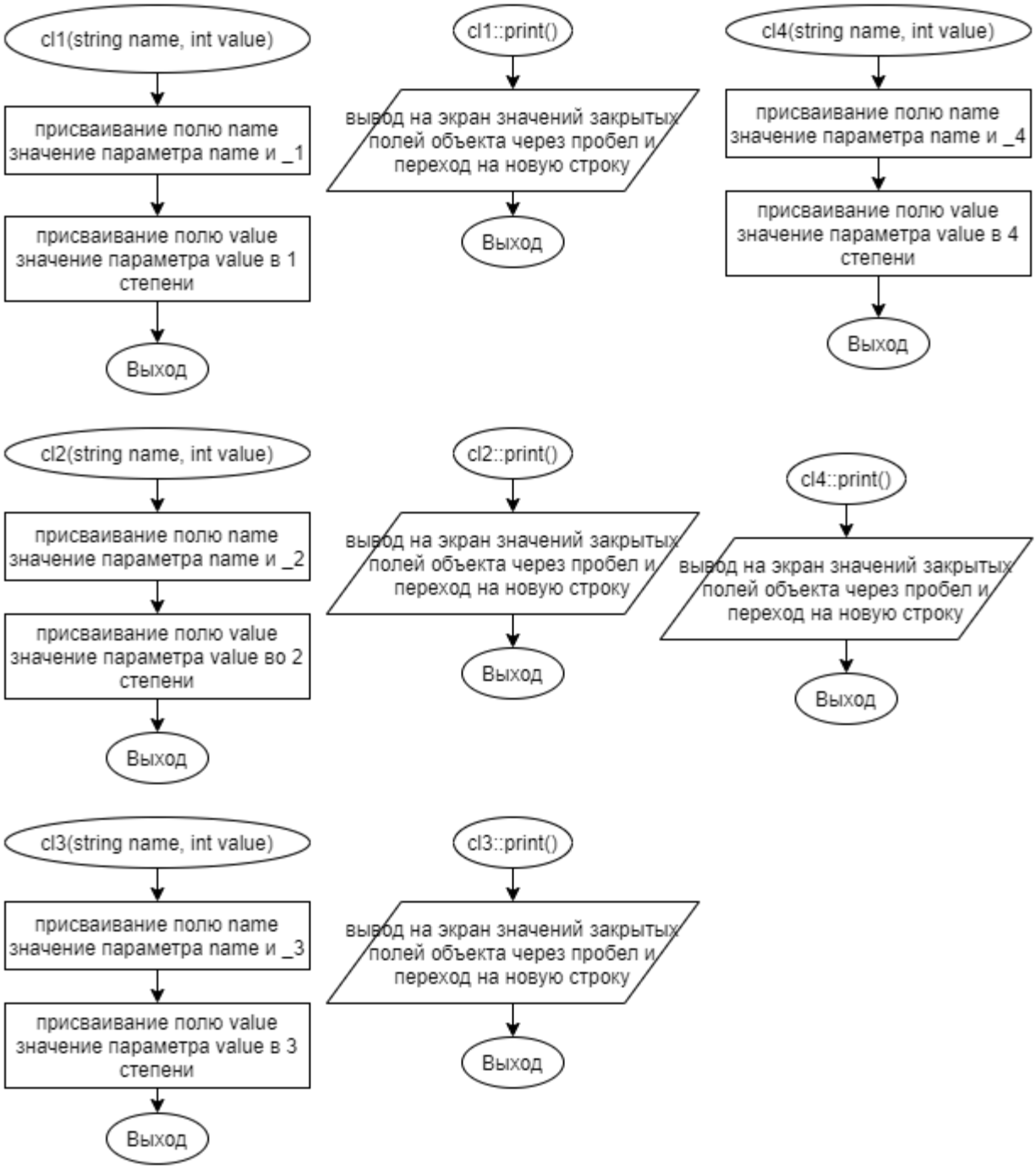

Рисунок 1 – Блок-схема алгоритма

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, int value)
{
   this -> name = name + "_1";
   this -> value = value;
}
void cl1::print()
{
   std::cout << name << ' ' << value << std::endl;
}
```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```
#ifndef __CL1__H
#define __CL1__H

#include <iostream>

class cl1
{
private:
   std::string name;
   int value;
public:
   cl1(std::string name, int value);
   void print();
};

#endif
```

5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

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

cl2::cl2(std::string name, int value): cl1(name, value)
{
   this -> name = name + "_2";
   this -> value = value * value;
}
void cl2::print()
{
   std::cout << name << ' ' << value << std::endl;
}
```

5.4 Файл cl2.h

Листинг 4 – cl2.h

```
#ifndef __CL2__H
#define __CL2__H

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

class cl2: private cl1
{
private:
   std::string name;
   int value;
public:
   cl2(std::string name, int value);
   void print();
};

#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

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

```
#include <iostream>

cl3::cl3(std::string name, int value): cl2(name, value)
{
   this -> name = name + "_3";
   this -> value = value * value * value;
}
void cl3::print()
{
   std::cout << name << ' ' << value << std::endl;
}
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3__H
#define __CL3__H

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

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

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

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

cl4::cl4(std::string name, int value): cl3(name, value)
{
   this -> name = name + "_4";
```

```
   this -> value = value * value * value * value;
}
void cl4::print()
{
   std::cout << name << ' ' << value << std::endl;
}
```

5.8 Файл cl4.h

Листинг 8 – cl4.h

```
#ifndef __CL4__H
#define __CL4__H

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

class cl4: private cl3
{
private:
   std::string name;
   int value;
public:
   cl4(std::string name, int value);
   void print();
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "cl1.h"
#include "cl2.h"
#include "cl3.h"
#include "cl4.h"

int main()
{
   std::string name; int value;
   std::cin >> name >> value;
   cl1* object = (cl1*) new cl4(name, value);
```

```
        object->print();
        ((cl2*) object)->print();
        ((cl3*) object)->print();
        ((cl4*) object)->print();
        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`

`Ob 5`	`Ob_1 5`
	`Ob_2 25`
	`Ob_3 125`
	`Ob_4 625`
		`Ob_1 5`
		`Ob_2 25`
		`Ob_3 125`
		`Ob_4 625`

`O 10`	`O_1 10`
	`O_2 100`
	`O_3 1000`
	`O_4 10000`
		`O_1 10`
		`O_2 100`
		`O_3 1000`
		`O_4 10000`

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).