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. Вводиться 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 понадобится:

- объект потока ввода cin и объект потока вывода cout
- библиотека string
- указатель this
- переменная типа int
- 4 класса
- объекта четвертого класса

Класс cl1 базовый класс

Поля:

скрытые элементы:

string s1

int n1

методы:

открытые:

cl1(string s1, int n1) - конструктор класса;

защищенные:

void power() - вывод значений.

Класс cl2 наследует класс cl1

Поля:

скрытые элементы:

string s2

int n2

методы:

открытые:

cl2(string s2, int n2) - конструктор класса;

защищенные:

void power() - вывод значений.

Класс cl3 наследует класс cl2

Поля:

скрытые элементы:

string s3

int n3

методы:

открытые:

cl3(string s3, int n3) - конструктор класса;

защищенные:

void power() - вывод значений.

Класс cl4 наследует класс cl3

Поля:

скрытые элементы:

string s4

int n4

методы:

открытые:

cl4(string s4, int n4) - конструктор класса;

void power() - вывод значений.

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		ввод идентификатора и натуральное число от 2 до 10	2
2		создание объекта класса 4	3
3		вывод наименование объекта класса и значение целочисленного свойства	∅

3.2 Алгоритм конструктора класса cl1

Функционал: присваивание строкового параметра, номер класса и возведение в степень номер класса.

Параметры: string s1, int n1.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		присваивание строкового параметра, номер класса	2
2		возведение в степень номер класса	∅

3.3 Алгоритм метода power класса cl1

Функционал: вывод идентификатора и натурального числа.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода power класса cl1

№	Предикат	Действия	№ перехода
1		вывод идентификатора и натурального числа	∅

3.4 Алгоритм конструктора класса cl2

Функционал: присваивание строкового параметра, номер класса и возведение в степень номер класса.

Параметры: string s2, int n2.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса cl2

№	Предикат	Действия	№ перехода
1		присваивание строкового параметра, номер класса	2
2		возведение в степень номер класса	∅

3.5 Алгоритм метода power класса cl2

Функционал: вывод идентификатора и натурального числа.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода power класса cl2

№	Предикат	Действия	№ перехода
1		вывод идентификатора и натурального числа	∅

3.6 Алгоритм конструктора класса cl3

Функционал: присваивание строкового параметра, номер класса и возведение в степень номер класса.

Параметры: string s3, int n3.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса cl3

№	Предикат	Действия	№ перехода
1		присваивание строкового параметра, номер класса	2
2		возведение в степень номер класса	∅

3.7 Алгоритм метода power класса cl3

Функционал: вывод идентификатора и натурального числа.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода power класса cl3

№	Предикат	Действия	№ перехода
1		вывод идентификатора и натурального числа	∅

3.8 Алгоритм конструктора класса cl4

Функционал: присваивание строкового параметра, номер класса и возведение в степень номер класса.

Параметры: string s4, int n4.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса cl4

№	Предикат	Действия	№ перехода
1		присваивание строкового параметра, номер класса	2
2		возведение в степень номер класса	∅

3.9 Алгоритм метода power класса cl4

Функционал: вывод идентификатора и натурального числа.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода power класса cl4

№	Предикат	Действия	№ перехода
1		вывод идентификатора и натурального числа	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

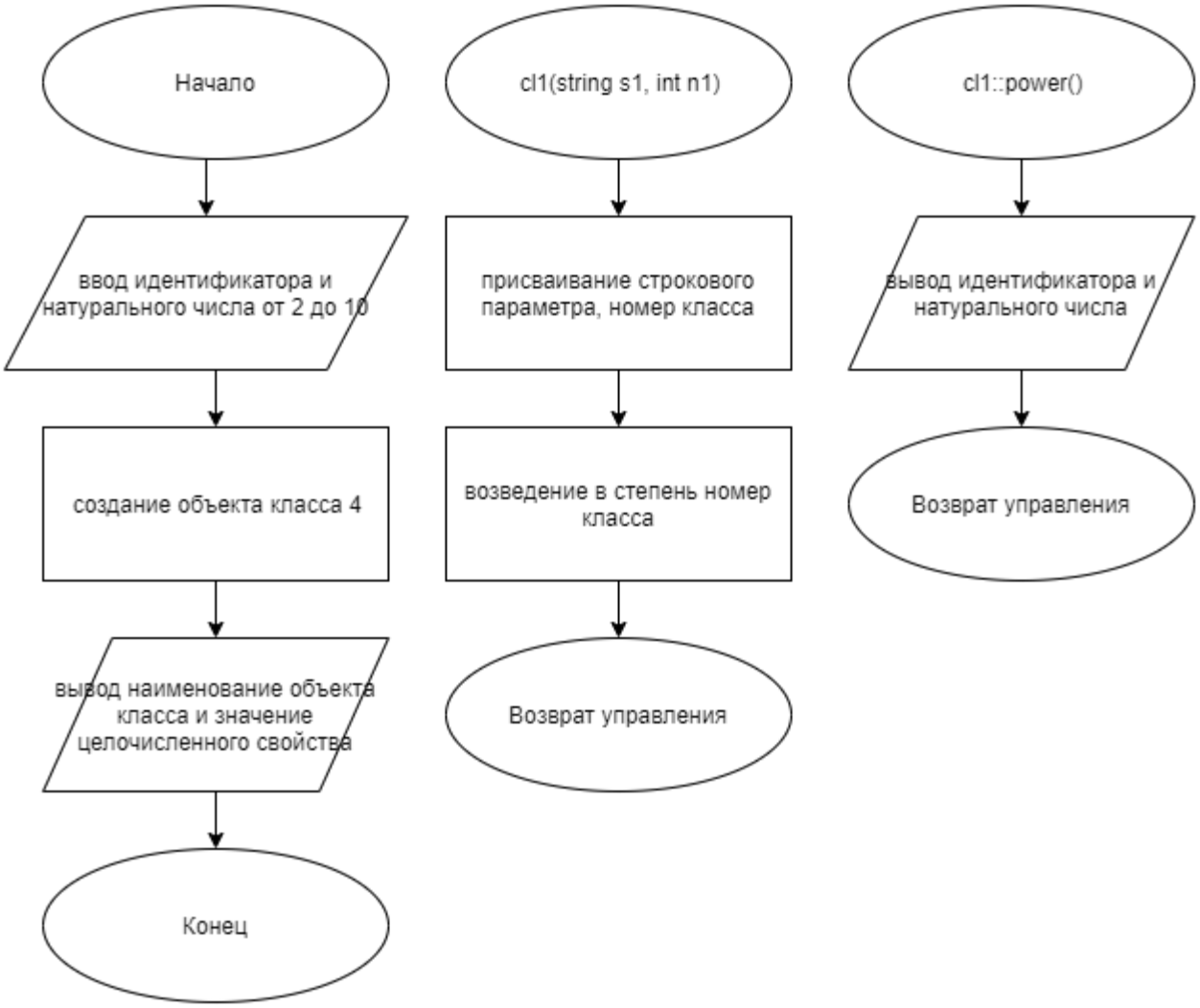

Рисунок 1 – Блок-схема алгоритма

Рисунок 2 – Блок-схема алгоритма

Рисунок 3 – Блок-схема алгоритма

Рисунок 4 – 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>
#include <string>

using namespace std;

cl1::cl1(string s1, int n1)
{
	this->s1 = s1 + "_1";
	this->n1 = n1;
}

void cl1::power()
{
	cout << s1 << " " << n1 << endl;
}
```

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 s1;
	int n1;
public:
	cl1(string s1, int n1);
	void power();
};
```

```
#endif
```

5.3 Файл cl2.cpp

Листинг 3 – cl2.cpp

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

using namespace std;

cl2::cl2(string s2, int n2) : cl1(s2, n2)
{
	this->s2 = s2 + "_2";
	this->n2 = n2 * n2;
}

void cl2::power()
{
	cout << s2 << " " << n2 << endl;
}
```

5.4 Файл cl2.h

Листинг 4 – cl2.h

```
#ifndef __CL2_H
#define __CL2_H
#include "cl1.h"
#include <iostream>
#include <string>

using namespace std;

class cl2 : public cl1
{
private:
	string s2;
	int n2;
public:
	cl2(string s2, int n2);
	void power();
};

#endif
```

5.5 Файл cl3.cpp

Листинг 5 – cl3.cpp

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

using namespace std;

cl3::cl3(string s3, int n3) : cl2(s3, n3)
{
	this->s3 = s3 + "_3";
	this->n3 = n3 * n3 * n3;
}

void cl3::power()
{
	cout << s3 << " " << n3 << endl;
}
```

5.6 Файл cl3.h

Листинг 6 – cl3.h

```
#ifndef __CL3_H
#define __CL3_H
#include "cl2.h"
#include <iostream>
#include <string>

using namespace std;

class cl3 : public cl2
{
private:
	string s3;
	int n3;
public:
	cl3(string s3, int n3);
	void power();
};

#endif
```

5.7 Файл cl4.cpp

Листинг 7 – cl4.cpp

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

using namespace std;

cl4::cl4(string s4, int n4) : cl3(s4, n4)
{
	this->s4 = s4 + "_4";
	this->n4 = n4 * n4 * n4 * n4;
}

void cl4::power()
{
	cout << s4 << " " << n4;
}
```

5.8 Файл cl4.h

Листинг 8 – cl4.h

```
#ifndef __CL4_H
#define __CL4_H
#include "cl3.h"
#include <iostream>
#include <string>

using namespace std;

class cl4 : public cl3
{
private:
	string s4;
	int n4;
public:
	cl4(string s4, int n4);
	void power();
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

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

using namespace std;

int main()
{
	string s;
	int n;
	cin >> s >> n;
	cl4 object(s, n);
	object.cl1::power();
	object.cl2::power();
	object.cl3::power();
	object.cl4::power();
	return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 10.

Таблица 10 – 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`

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).