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:

1. Наименование объекта по шаблону: «значение строкового параметра»_«номер класса»;
2. 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:

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

которому в качестве аргументов передаются введенный идентификатор и натуральное число.

- 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:

3. Объекты стандартного потока ввода/вывода cin/cout
4. Объект класса Class_4

Таблица ирархии классов

Таблица 1 – Иерархия наследования классов

№	Наименования	Классы наследники	Модификаторы доступа	Описание	Номер класса наследника	Клмментарий
1	Class_1			Базовый класс		
		Class_2	public		2	
2	Class_2			Класс, производный от Class_1		
		Class_3	public		3	
3	Class_3			Класс, производный от Class_2		
		Class_4	public		4	
4	Class_4			Класс, производный от Class_3		

- Класс Class_1, Class_2, Class_3, Class_4
  - Поля/свойства:
    - Строковое поле
      - Наименование: Name
      - Тип данных: строка
      - Модификатор доступа: закрытое
    - Целочисленнное поле
      - Наименование: Name
      - Тип данных: строка
      - Модификатор доступа: закрытое
  - Методы:
    - Конструктор Class_1, Class_2, Class_3, Class_4
      - Функционал: параметрический конструктор
    - Метод Print
      - Функционал: вывод значений полей

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 Class_1

Функционал: Манипуляции с name1/num1.

Параметры: string, name1; int, num1.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Class_1

№	Предикат	Действия	№ перехода
1		name = name1 + "_1", num = num1 * 1	∅

3.2 Алгоритм конструктора класса Class_2

Функционал: Манипуляции с name1/num1.

Параметры: string, name1; int, num1.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Class_2

№	Предикат	Действия	№ перехода
1		name = name1 + "_2", num = num1 * num1	∅

3.3 Алгоритм конструктора класса Class_3

Функционал: Манипуляции с name1/num1.

Параметры: string, name1; int, num1.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса Class_3

№	Предикат	Действия	№ перехода
1		name = name1 + "_3", num = num1 * num1 * num1	∅

3.4 Алгоритм конструктора класса Class_4

Функционал: Манипуляции с name1/num1.

Параметры: string, name1; int, num1.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса Class_4

№	Предикат	Действия	№ перехода
1		name = name1 + "_4", num = num1 * num1 * num1 * num1	∅

3.5 Алгоритм метода Print класса Class_1

Функционал: Вывод значений name, num.

Параметры: отсутствуют.

Возвращаемое значение: Текст, целый тип.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода Print класса Class_1

№	Предикат	Действия	№ перехода
1		Вывод значений name, num	∅

3.6 Алгоритм метода Print класса Class_2

Функционал: Вывод значений name, num.

Параметры: отсутствуют.

Возвращаемое значение: Текст, целый тип.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода Print класса Class_2

№	Предикат	Действия	№ перехода
1		Вывод значений name, num	∅

3.7 Алгоритм метода Print класса Class_3

Функционал: Вывод значений name, num.

Параметры: отсутствуют.

Возвращаемое значение: Текст, целый тип.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода Print класса Class_3

№	Предикат	Действия	№ перехода
1		Вывод значений name, num	∅

3.8 Алгоритм метода Print класса Class_4

Функционал: Вывод значений name, num.

Параметры: отсутствуют.

Возвращаемое значение: Текст, целый тип.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода Print класса Class_4

№	Предикат	Действия	№ перехода
1		Вывод значений name, num	∅

3.9 Алгоритм функции main

Функционал: основная фенкция.

Параметры: отсутствуют.

Возвращаемое значение: целый тип.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Создание переменной num типа int	2
2		Создание переменной name типа string	3
3		Ввод name, num	4
4	num >= 2 && num <= 10		5
			∅
5		Создание указателя на объект типа Class_1	6
6		Использование метода Print() из класса Class_1	7
7		Использование метода Print() из класса Class_2	8
8		Использование метода Print() из класса Class_3	9
9		Использование метода Print() из класса Class_4	10
10		Отчистка памяти выделенной под указатель	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

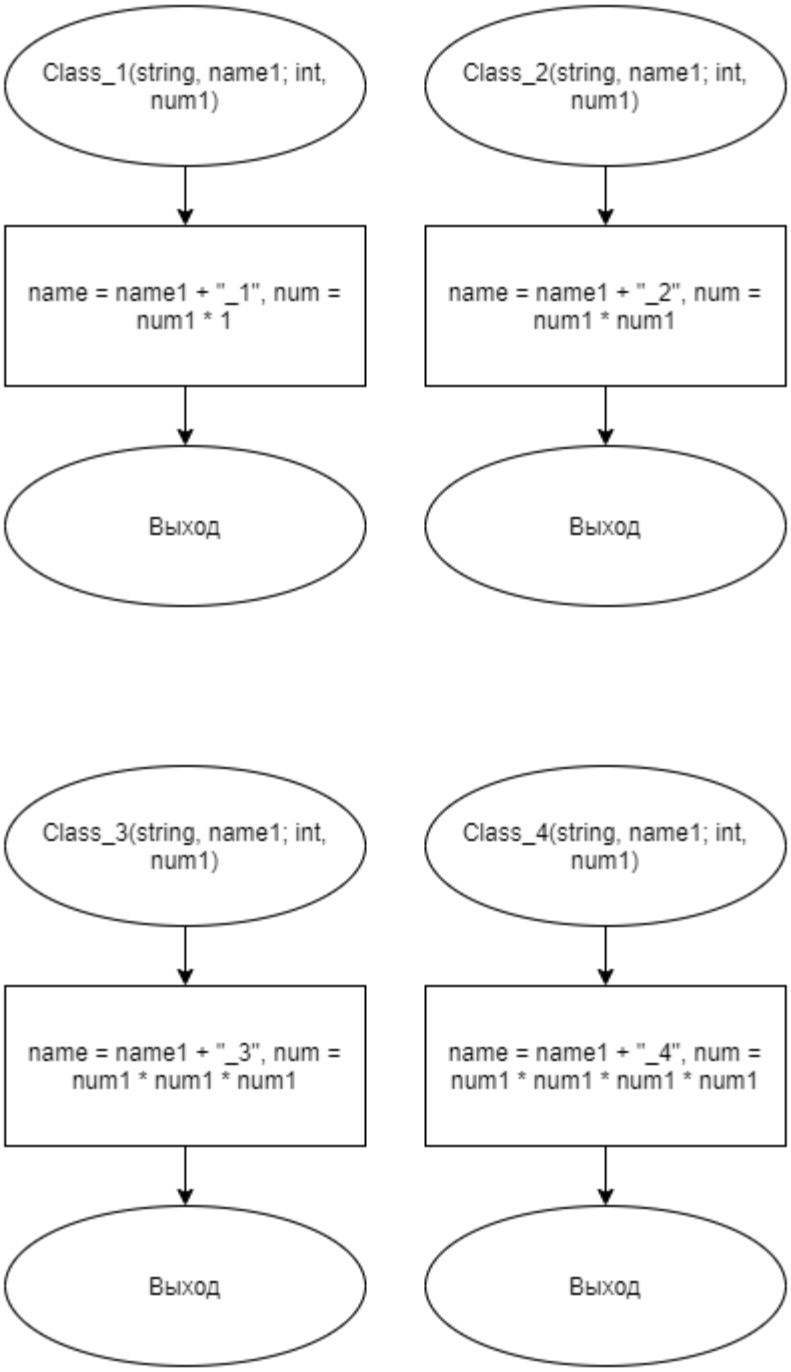

Рисунок 1 – Блок-схема алгоритма

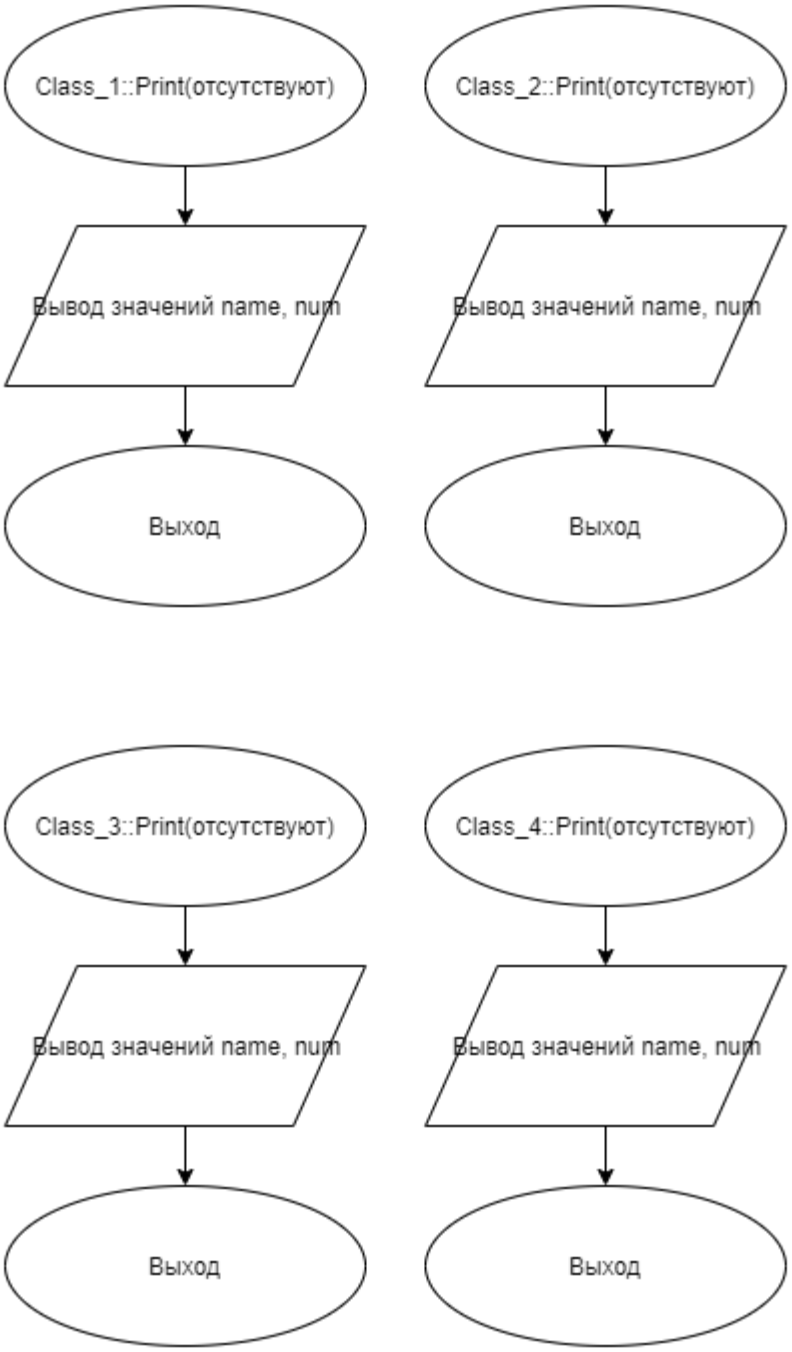

Рисунок 2 – Блок-схема алгоритма

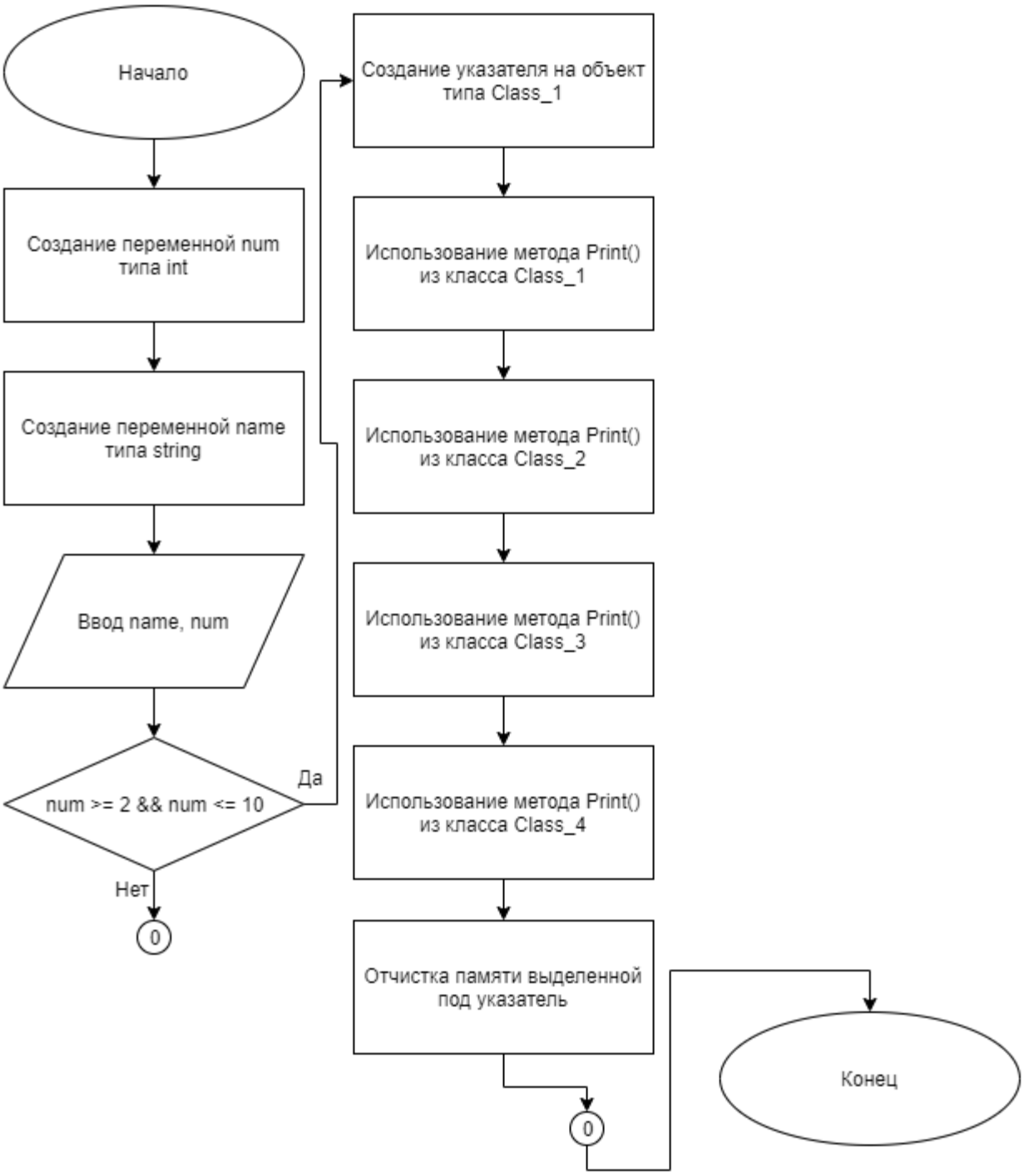

Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Class_1.cpp

Листинг 1 – Class_1.cpp

```
#include "Class_1.h"


Class_1::Class_1(string name1, int num1):name(name1 + "_1"), num(num1){}

void Class_1::Print()
{
	cout << name + " " << num;
}
```

5.2 Файл Class_1.h

Листинг 2 – Class_1.h

```
#ifndef __CLASS_1_H__
#define __CLASS_1_H__
#include <iostream>
#include <string>
using namespace std;

class Class_1
{
public:
	Class_1(string name1, int num1);
	void Print();
private:
	string name;
	int num;
};
#endif
```

5.3 Файл Class_2.cpp

Листинг 3 – Class_2.cpp

```
#include "Class_2.h"


Class_2::Class_2(string name1, int num1):Class_1(name1, num1), name(name1 + "_2"),
num(num1*num1){}

void Class_2::Print()
{
	cout << name + " " << num;
}
```

5.4 Файл Class_2.h

Листинг 4 – Class_2.h

```
#ifndef __CLASS_2_H__
#define __CLASS_2_H__
#include "Class_1.h"

class Class_2:public Class_1
{
public:
	Class_2(string name1, int num1);
	void Print();
private:
	string name;
	int num;
};
#endif
```

5.5 Файл Class_3.cpp

Листинг 5 – Class_3.cpp

```
#include "Class_3.h"


Class_3::Class_3(string name1, int num1):Class_2(name1, num1), name(name1 + "_3"),
num(num1*num1*num1){}

void Class_3::Print()
{
	cout << name + " " << num;
```

```
}
```

5.6 Файл Class_3.h

Листинг 6 – Class_3.h

```
#ifndef __CLASS_3_H__
#define __CLASS_3_H__
#include "Class_2.h"

class Class_3:public Class_2
{
public:
	Class_3(string name1, int num1);
	void Print();
private:
	string name;
	int num;
};
#endif
```

5.7 Файл Class_4.cpp

Листинг 7 – Class_4.cpp

```
#include "Class_4.h"


Class_4::Class_4(string name1, int num1):Class_3(name1, num1), name(name1 + "_4"),
num(num1*num1*num1*num1){}

void Class_4::Print()
{
	cout << name + " " << num;
}
```

5.8 Файл Class_4.h

Листинг 8 – Class_4.h

```
#ifndef __CLASS_4_H__
#define __CLASS_4_H__
#include "Class_3.h"

class Class_4:public Class_3
```

```
{
public:
	Class_4(string name1, int num1);
	void Print();
private:
	string name;
	int num;
};
#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include "Class_4.h"

int main()
{
	int num;
	string name;
	cin >> name >> num;
	if (num >= 2 && num <= 10)
	{
		Class_1* obj = new Class_4(name, num);
		obj->Print();
		cout << endl;
		((Class_2*)obj)->Print();
		cout << endl;
		((Class_3*)obj)->Print();
		cout << endl;
		((Class_4*)obj)->Print();
		delete obj;
	}
}
```

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`

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).