

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.....	8
3 ОПИСАНИЕ АЛГОРИТМОВ.....	11
3.1 Алгоритм конструктора класса cl_1.....	11
3.2 Алгоритм метода Method класса cl_1.....	11
3.3 Алгоритм деструктора класса cl_1.....	11
3.4 Алгоритм конструктора класса cl_2.....	12
3.5 Алгоритм метода Method класса cl_2.....	12
3.6 Алгоритм конструктора класса cl_3.....	13
3.7 Алгоритм метода Method класса cl_3.....	13
3.8 Алгоритм конструктора класса cl_4.....	13
3.9 Алгоритм метода Method класса cl_4.....	14
3.10 Алгоритм функции main.....	14
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	16
5 КОД ПРОГРАММЫ.....	22
5.1 Файл cl_1.cpp.....	22
5.2 Файл cl_1.h.....	22
5.3 Файл cl_2.cpp.....	22
5.4 Файл cl_2.h.....	23
5.5 Файл cl_3.cpp.....	23
5.6 Файл cl_3.h.....	24
5.7 Файл cl_4.cpp.....	24
5.8 Файл cl_4.h.....	24
5.9 Файл main.cpp.....	25

6 ТЕСТИРОВАНИЕ.....	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	27

# 1 ПОСТАНОВКА ЗАДАЧИ

## Полиморфизм в иерархии классов

Описать четыре класса которые последовательно наследуют друг друга, с номерами классов 1, 2, 3, 4. В каждом классе реализовать виртуальный метод с открытым доступом и одинаковым именем. Метод вычисляет значение многочлена степени номера класса и возвращает полученный результат. Коэффициенты и переменная многочлена целочисленные.

В основной функции реализовать алгоритм, в котором использовать один указатель на объект класса. Алгоритм:

1. Объявление указателя на объект класса.
2. Объявление четырех целочисленных переменных  $a_1, a_2, a_3, a_4$ , которые соответствуют коэффициентам многочлена  $(a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + a_4 \cdot x^4)$ .
3. Объявление целочисленной переменной  $x$ , которая соответствует переменной многочлена.
4. Ввод значения переменных  $a_1, a_2, a_3, a_4$ .
5. Создание объекта класса 4 посредством параметризованного конструктора, передав в качестве аргументов  $a_1, a_2, a_3, a_4$ . Обеспечить передачу необходимых коэффициентов объектам согласно наследственности классов.
6. Начало цикла
  - 6.1. Реализовать ввод значения переменной  $x$ .
  - 6.2. Если значение  $x$  равно нулю, то завершить цикл.
  - 6.3. Иначе, реализовать ввод значения номера класса.
  - 6.4. Согласно номеру класса вызвать метод вычисления многочлена

посредством объекта, который соответствует номеру класса и результат вывести.

7. Конец цикла.

## **1.1 Описание входных данных**

**Первая строка:**

«целое число, значение a1» «целое число, значение a2» «целое число, значение a3» «целое число, значение a4»

**Начиная со второй строки, построчно:**

«целое число, значение x» «целое число, номер класса»

## **1.2 Описание выходных данных**

**Первая строка:**

a1 = «целое число» a2 = «целое число» a3 = «целое число» a4 = «целое число»

Наименование коэффициента отделяется от предыдущего целого числа четырьмя пробелами.

**Со второй строки и далее построчно:**

Class «номер класса» F( «значение переменной x» ) = «значение многочлена»

Фрагменту «F(» предшествует 4 пробела

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект `a` класса `cl_4` предназначен для указатель на объект класса `cl_1`;
- функция `new` для выделение памяти под объект;
- функция `delete` для освобождение выделенной под объект памяти;
- `cin` - объект стандартного потока ввода;
- `cout` - объект стандартного потока вывода;
- `switch` - множественный условный оператор;
- `case, default` - решение условного оператора;
- `if` - условный оператор;
- `while` - оператор цикла с условием.

Класс `cl_1`:

- свойства/поля:
  - поле Коэффициент для первой степени:
    - наименование — `a1`;
    - тип — `int`;
    - модификатор доступа — `private`;
- функционал:
  - метод `cl_1` — Конструктор;
  - метод `Method` — Значение многочлена;
  - метод `~cl_1` — Деструктор.

Класс `cl_2`:

- свойства/поля:
  - поле Коэффициент для первой степени:
    - наименование — `a2`;
    - тип — `int`;

- модификатор доступа — private;
- функционал:
  - о метод cl\_2 — Конструктор;
  - о метод Method — Значение многочлена.

Класс cl\_3:

- свойства/поля:
  - о поле Коэффициент для первой степени:
    - наименование — a3;
    - тип — int;
    - модификатор доступа — private;
- функционал:
  - о метод cl\_3 — Конструктор;
  - о метод Method — Значение многочлена.

Класс cl\_4:

- свойства/поля:
  - о поле Коэффициент для первой степени:
    - наименование — a4;
    - тип — int;
    - модификатор доступа — private;
- функционал:
  - о метод cl\_4 — Конструктор;
  - о метод Method — Значение многочлена.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	cl_1			Родительский класс	
		cl_2	public		2

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
2	cl_2			Производный от cl_1	
		cl_3	public		3
3	cl_3			Производный от cl_2	
		cl_4	public		4
4	cl_4			Производный от cl_3	



## 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 cl\_1

Функционал: Конструктор.

Параметры: int a1 - коэффициент.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl\_1

№	Предикат	Действия	№ перехода
1		присвоение полю a1=a1 - параметра	Ø

### 3.2 Алгоритм метода Method класса cl\_1

Функционал: Значение многочлена.

Параметры: int x - переменная.

Возвращаемое значение: int - значение многочлена.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Method класса cl\_1

№	Предикат	Действия	№ перехода
1		Возвращение a1*x	Ø

### 3.3 Алгоритм деструктора класса cl\_1

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса *cl\_1*

№	Предикат	Действия	№ перехода
1		Уничтожение объекта	Ø

### 3.4 Алгоритм конструктора класса *cl\_2*

Функционал: Конструктор.

Параметры: int a2,a1 - коэффициенты.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса *cl\_2*

№	Предикат	Действия	№ перехода
1		присвоение полю a2=a2 - параметра	Ø

### 3.5 Алгоритм метода *Method* класса *cl\_2*

Функционал: Значение многочлена.

Параметры: int x - переменная.

Возвращаемое значение: int - значение многочлена.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода *Method* класса *cl\_2*

№	Предикат	Действия	№ перехода
1		Возвращение $a2 \cdot x \cdot x$	Ø

### 3.6 Алгоритм конструктора класса cl\_3

Функционал: Конструктор.

Параметры: int a3,a2,a1 - коэффициенты.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса cl\_3

№	Предикат	Действия	№ перехода
1		присвоение полю a3=a3 - параметра	Ø

### 3.7 Алгоритм метода Method класса cl\_3

Функционал: Значение многочлена.

Параметры: int x - переменная.

Возвращаемое значение: int - значение многочлена.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода Method класса cl\_3

№	Предикат	Действия	№ перехода
1		Возвращение $a3*x*x*x$	Ø

### 3.8 Алгоритм конструктора класса cl\_4

Функционал: Конструктор.

Параметры: int a4,a3,a2,a1 - коэффициенты.

Алгоритм конструктора представлен в таблице 9.

Таблица 9 – Алгоритм конструктора класса cl\_4

№	Предикат	Действия	№ перехода
1		присвоение полю a4=a4 - параметра	Ø

### 3.9 Алгоритм метода Method класса cl\_4

Функционал: Значение многочлена.

Параметры: int x - переменная.

Возвращаемое значение: int - значение многочлена.

Алгоритм метода представлен в таблице 10.

Таблица 10 – Алгоритм метода Method класса cl\_4

№	Предикат	Действия	№ перехода
1		Возвращение $a4*x*x*x*x$	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 указателя на объект a класса cl_1	2
2		объявление int a1, a2, a3, a4, x, i_cl	3
3		ввод a1, a2, a3, a4 с клавиатуры	4
4		вывод с первой строки коэффициентов через пробел	5
5		присвоение указателю a = new cl_4 с параметрами a1, a2, a3, a4	6
6	cin>>x		8
			7

№	Предикат	Действия	№ перехода
7		освобождение памяти для указателя a	∅
8	x==0		∅
		ввод с клавиатуры i_cl	9
9		вывоод метода Method с параметром x для класса с именем "cl_i_class"	6

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-6.

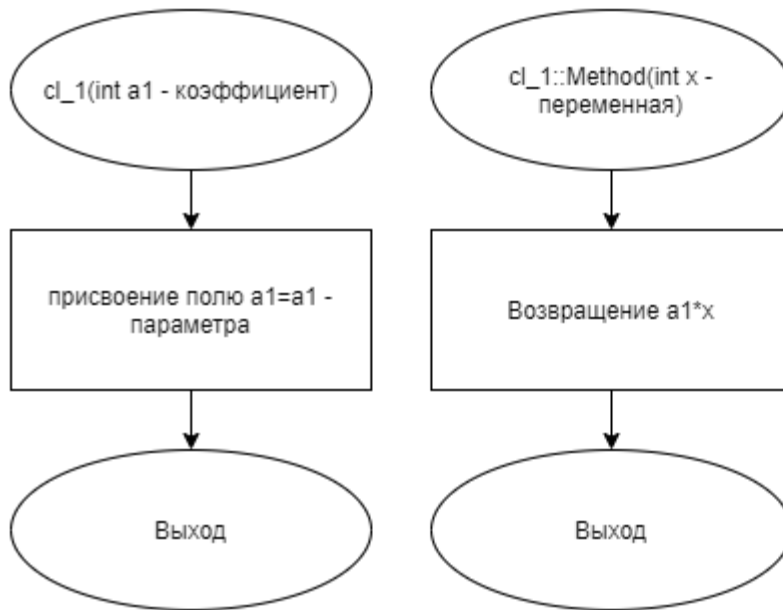
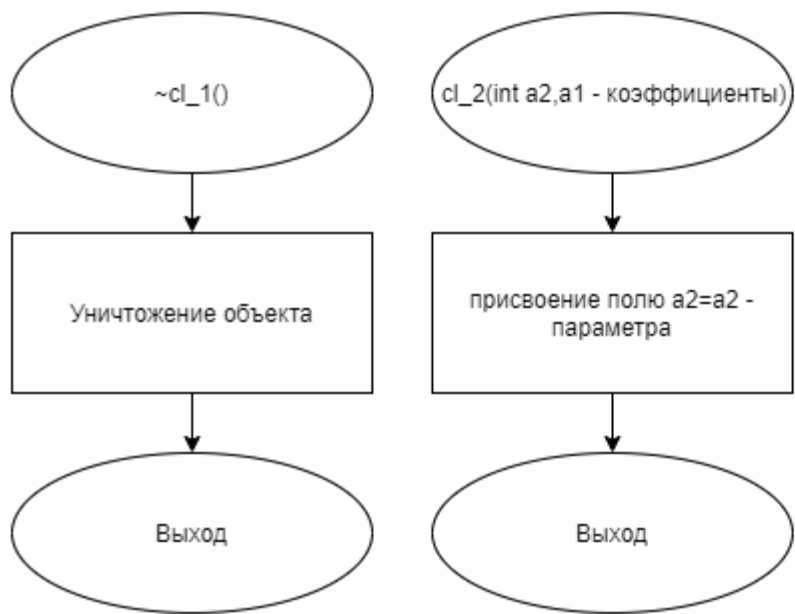
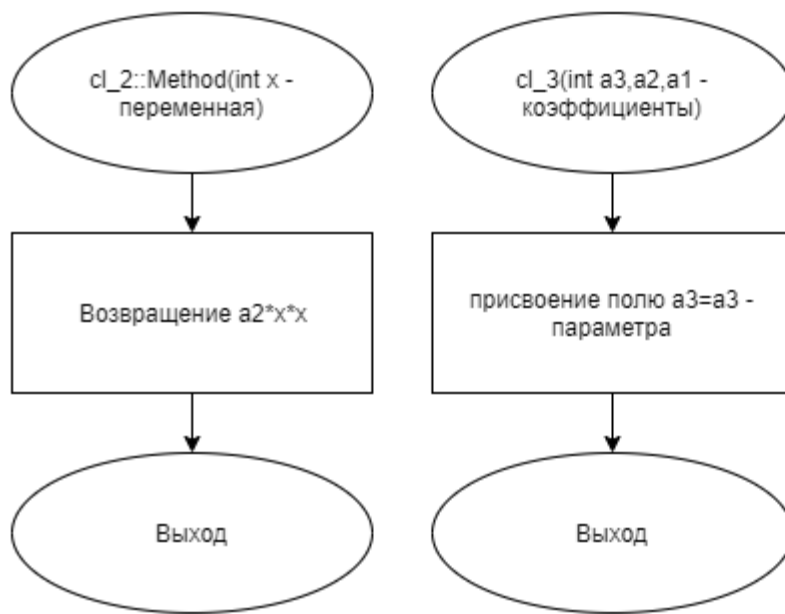


Рисунок 1 – Блок-схема алгоритма

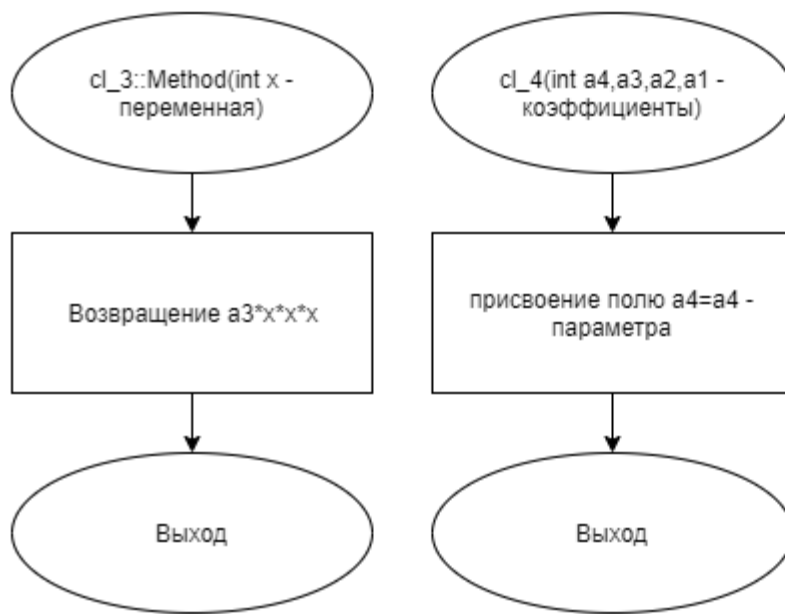


**Рисунок 2 – Блок-схема алгоритма**



**Рисунок 3 – Блок-схема алгоритма**





**Рисунок 4 – Блок-схема алгоритма**

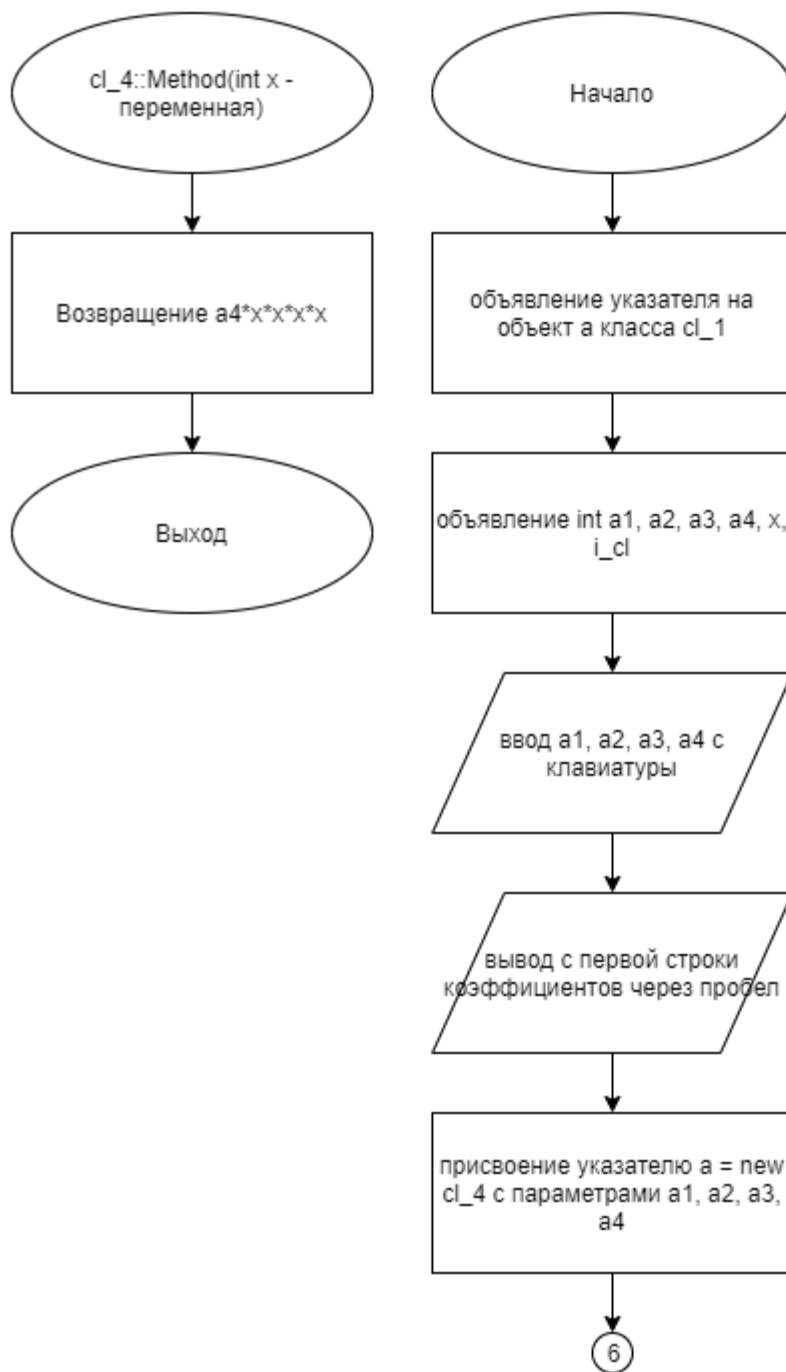


Рисунок 5 – Блок-схема алгоритма

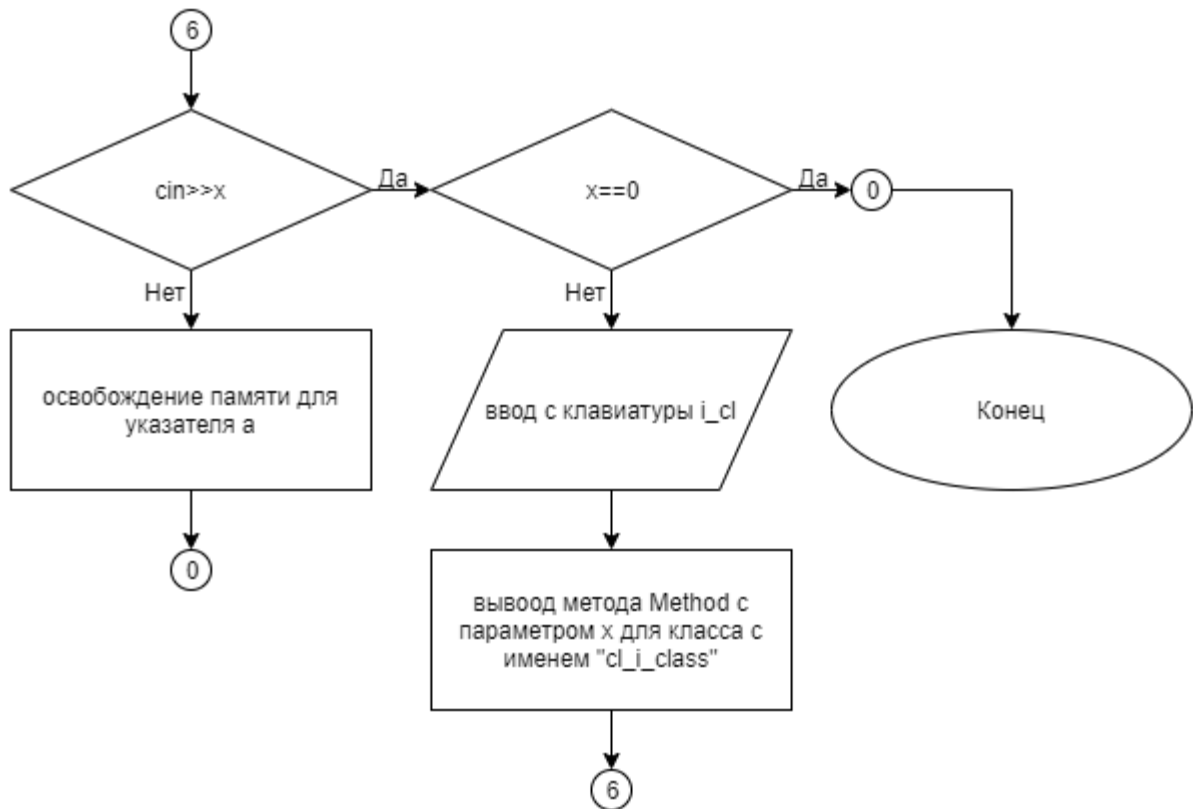


Рисунок 6 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл cl\_1.cpp

*Листинг 1 – cl\_1.cpp*

```
#include "cl_1.h"
cl_1::cl_1(int a1){
    this->a1=a1;
}
int cl_1::Method(int x){
    return a1*x;
}
cl_1::~~cl_1(){}

```

### 5.2 Файл cl\_1.h

*Листинг 2 – cl\_1.h*

```
#ifndef __CL_1__H
#define __CL_1__H
class cl_1{
private:
    int a1;
public:
    cl_1(int a1);
    int Method(int x);
    ~cl_1();
};
#endif

```

### 5.3 Файл cl\_2.cpp

*Листинг 3 – cl\_2.cpp*

```
#include "cl_2.h"

```

```

cl_2::cl_2(int a2, int a1):cl_1(a1){
    this->a2=a2;
}
int cl_2::Method(int x){
    return a2*x*x;
}

```

## 5.4 Файл cl\_2.h

*Листинг 4 – cl\_2.h*

```

#ifndef __CL_2__H
#define __CL_2__H
#include "cl_1.h"
class cl_2:public cl_1{
private:
    int a2;
public:
    cl_2(int a2,int a1);
    int Method(int x);
};

#endif

```

## 5.5 Файл cl\_3.cpp

*Листинг 5 – cl\_3.cpp*

```

#include "cl_3.h"
cl_3::cl_3(int a3,int a2,int a1):cl_2(a2,a1){
    this->a3=a3;
}
int cl_3::Method(int x){
    return a3*x*x*x;
}

```

## 5.6 Файл cl\_3.h

Листинг 6 – cl\_3.h

```
#ifndef __CL_3__H
#define __CL_3__H
#include "cl_2.h"
class cl_3:public cl_2{
private:
    int a3;
public:
    cl_3(int a3,int a2,int a1);
    int Method(int x);
};
#endif
```

## 5.7 Файл cl\_4.cpp

Листинг 7 – cl\_4.cpp

```
#include "cl_4.h"
cl_4::cl_4(int a1,int a2,int a3,int a4):cl_3(a3,a2,a1){
    this->a4=a4;
}
int cl_4::Method(int x){
    return a4*x*x*x*x;
}
```

## 5.8 Файл cl\_4.h

Листинг 8 – cl\_4.h

```
#ifndef __CL_4__H
#define __CL_4__H
#include "cl_3.h"
class cl_4:public cl_3{
private:
    int a4;
public:
    cl_4(int a1,int a2,int a3,int a4);
    int Method(int x);
};
#endif
```

## 5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "cl_1.h"
#include "cl_2.h"
#include "cl_3.h"
#include "cl_4.h"
#include <iostream>
using namespace std;
int main()
{
    cl_1* a;
    int a1,a2,a3,a4;
    int x,i_cl;
    cin>>a1>>a2>>a3>>a4;
    cout<<"a1="<<a1<<" a2="<<a2<<" a3="<<a3<<" a4="<<a4<<endl;
    a=new cl_4(a1,a2,a3,a4);
    while (cin>>x){
        if (x==0){
            return (0);
        }
        cin>>i_cl;
        switch(i_cl){
            case(1):
                cout<<"Class 1      F("<<x<<" )="<<a->Method(x)<<endl;
                break;
            case(2):
                cout<<"Class      2              F("<<x<<" )="<<((cl_2*)a)-
>cl_2::Method(x)<<endl;
                break;
            case(3):
                cout<<"Class      3              F("<<x<<" )="<<((cl_3*)a)-
>cl_3::Method(x)<<endl;
                break;
            case(4):
                cout<<"Class      4              F("<<x<<" )="<<((cl_4*)a)-
>cl_4::Method(x)<<endl;
                break;
            default:
                break;
        }
    }
    delete a;
    return(0);
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 12.

Таблица 12 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 2 3 4 2 1	a1=1 a2=2 a3=3 a4=4 Class 1 F(2)=2	a1=1 a2=2 a3=3 a4=4 Class 1 F(2)=2



## 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](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](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).