МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**« МИРЭА Российский технологический университет»**

**РТУ МИРЭА**

Институт Информационных технологий

Кафедра Вычислительной техники

# УЧЕБНОЕ ЗАДАНИЕ

по дисциплине

« Объектно-ориентированное программирование»

Наименование задачи:

## « Задание 5_3_1 »

| | | |
|---|---|---|
| С тудент группы | ИКБО-13-21 | Черномуров С.А. |
| Руководитель практики | Ассистент | Асадова Ю.С. |
| Работа представлена | «___»_____ 2022 г. | _____ |
| | | *(подпись студента)* |
| Оценка | | _____ |
| | | *(подпись руководителя)* |

Москва 2022

**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

## 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 a1, a2, a3 a4, которые соответствуют коэффициентам многочлена (a1*x + a2*x*x + a3*x*x*x + a4*x*x*x*x).
3. Объявление целочисленной переменной x, которая соответствует переменной многочлена.
4. Ввод значения переменных a1, a2, a3 a4.
5. Создание объекта класса 4 посредством параметризированного конструктора, передав в качестве аргументов a1, a2, a3 a4. Обеспечить передачу необходимых коэффициентов объектам согласно наследственности классов.
6. Начало цикла
    1. Реализовать ввод значения переменной x.
    2. Если значение x равно нулю, то завершить цикл.
    3. Иначе, реализовать ввод значения номера класса.
    4. Согласно номеру класса вызвать метод вычисления многочлена посредством объекта, который соответствует номеру класса и

результат вывести.

7. Конец цикла.

**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»

**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

## Метод решения

Для решения задачи используются:

- Объекты стандартных потоков ввода и вывода cin и cout соответственно. Используются для ввода с клавиатуры и вывода на экран.
- Оператор цикла с предусловием while. Используется для множественного выполнения алгоритма программы.
- Условный оператор if .. else. Используется для выбора вызываемого метода и форматирования вывода.
- Объект obj класса Cl4. Используется для создания объекта.
- Объекты классов Cl1, Cl2, Cl3.
- **Класс Cl1:**
    - Свойства/поля:
        - Поле:
            - Наименование - a1;
            - Тип - целочисленный;
            - Модификатор доступа - защищенный.
    - Методы:
        - Метод Polynomial:
            - Функционал - параметризированный метод, вычисляющий значение многочлена первой степени от переданного в него значения и возвращающий полученный результат.
- **Класс Cl2:**
    - Свойства/поля:
        - Поле:
            - Наименование - a1;

- Тип - целочисленный;
- Модификатор доступа - защищенный.

- Поле:
  - Наименование - a2;
  - Тип - целочисленный;
  - Модификатор доступа - защищенный.

- Методы:
  - Метод Polynomial:
    - Функционал - параметризированный метод, вычисляющий значение многочлена второй степени от переданного в него значения и возвращающий полученный результат.

- **Класс Cl3:**
  - Свойства/поля:
    - Поле:
      - Наименование - a1;
      - Тип - целочисленный;
      - Модификатор доступа - защищенный.
    - Поле:
      - Наименование - a2;
      - Тип - целочисленный;
      - Модификатор доступа - защищенный.
    - Поле:
      - Наименование - a3;
      - Тип - целочисленный;
      - Модификатор доступа - защищенный.

- Методы:
    - Метод Polynomial:
        - Функционал - параметризированный метод, вычисляющий значение многочлена третьей степени от переданного в него значения и возвращающий полученный результат.

- **Класс Cl4:**
    - Свойства/поля:
        - Поле:
            - Наименование - a1;
            - Тип - целочисленный;
            - Модификатор доступа - защищенный.
        - Поле:
            - Наименование - a2;
            - Тип - целочисленный;
            - Модификатор доступа - защищенный.
        - Поле:
            - Наименование - a3;
            - Тип - целочисленный;
            - Модификатор доступа - защищенный.
        - Поле:
            - Наименование - a4;
            - Тип - целочисленный;
            - Модификатор доступа - защищенный.
    - Методы:
        - Метод Cl4:
            - Функционал - параметризированный конструктор, принимающий в себя четыре числовых значения -

коэффициентов многочлена и присваивающий их соответствующим полям всех классов.

- Метод Polynomial:
  - Функционал - параметризированный метод, вычисляющий значение многочлена четвертой степени от переданного в него значения и возвращающий полученный результат.

Иерархия наследования:

| № | Имя класса | Классы-наследники | Модификатор доступа при наследовании | Описание | Номер | Комментарий |
|---|---|---|---|---|---|---|
| 1 | Cl1 | | | Класс 1, является головным для класса 2, объекты класса содержат поля для вычисления многочлена первой степени | | |
| | | Cl2 | public | | 2 | |
| 2 | Cl2 | | | Класс 2, является головным для класса 3, объекты класса содержат поля для | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | вычисления многочлена второй степени | | |
| | | Cl3 | public | | 3 | |
| 3 | Cl3 | | | Класс 3, является головным для класса 4, объекты класса содержат поля для вычисления многочлена третьей степени | | |
| | | Cl4 | public | | 4 | |
| 4 | Cl4 | | | Класс 4, является дочерним для класса 3, объекты класса содержат поля для вычисления многочлена четвертой степени | | |

## Описание алгоритма

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

Функция: main

Функционал: Основной алгоритм программы

Параметры: Отсутствуют

Возвращаемое значение: Целочисленный тип данных - код возврата

Алгоритм функции представлен в таблице 2.

Таблица 2. Алгоритм функции main

| № | Предикат | Действия | № перехода | Комментарий |
|---|---|---|---|---|
| 1 | | Объявление указателя obj на объект класса Cl4 | 2 | |
| 2 | | Объявление целочисленных переменных a1, a2, a3, a4, x | 3 | |
| 3 | | Считывание с клавиатуры значений переменных a1, a2, a3, a4 | 4 | |
| 4 | | Вывод на экран "a1 = ", a1, " " | 5 | |
| 5 | | Вывод на экран "a2 = ", a2, " " | 6 | |
| 6 | | Вывод на экран "a3 = ", a3, " " | 7 | |
| 7 | | Вывод на экран "a4 = ", a4 с последующим переносом на новую строку | 8 | |

| | | | | |
|---|---|---|---|---|
| 8 | | Динамическое создание объекта obj класса Cl4 путем вызова параметризированного конструктора с целочисленными параметрами a1, a2, a3, a4 | 9 | |
| 9 | | Объявление целочисленной переменной с инициализацией flag=0 | 10 | |
| 10 | Логическое значение "Истина" | Считывание с клавиатуры значения переменной x | 11 | Цикл выполняется, пока не будет прерван вручную |
| | | | ∅ | Выход из цикла (не произойдет до вызова оператора break) |
| 11 | Значение x не равно нулю | Объявление целочисленной переменной cl_num | 12 | |
| | | Вызов оператора break | ∅ | Ручное прекращение выполнения цикла |
| 12 | | Считывание с клавиатуры значения переменной cl_num | 13 | |
| 13 | Значение flag не равно нулю | Вывод на экран переноса на новую строку | 14 | |
| | | | 14 | |
| 14 | | Вывод на экран "Class ", cl_num, " " | 15 | |
| 15 | | Вывод на экран "F( ", x, " ) = " | 16 | |
| 16 | Значение cl_num равно 1 | Вывод на экран значения, возвращенного методом Polynomial с параметром x объекта класса Cl1 | 20 | |

| | | | | |
|---|---|---|---|---|
| | | | 17 | |
| 17 | Значение cl_num равно 2 | Вывод на экран значения, возвращенного методом Polynomial с параметром x объекта класса Cl2 | 20 | |
| | | | 18 | |
| 18 | Значение cl_num равно 3 | Вывод на экран значения, возвращенного методом Polynomial с параметром x объекта класса Cl3 | 20 | |
| | | | 19 | |
| 19 | Значение cl_num равно 4 | Вывод на экран значения, возвращенного методом Polynomial с параметром x объекта obj класса Cl4 | 20 | |
| | | | 20 | |
| 20 | | Инкрементирование flag | 10 | |

Класс объекта: Cl1

Модификатор доступа: public

Метод: Polynomial

Функционал: Параметризированный метод, вычисляющий значение многочлена первой степени от переданного в него значения и возвращающий полученный результат

Параметры: Целочисленный параметр x

Возвращаемое значение: Целочисленный тип данных - значение многочлена первой степени

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода Polynomial класса Cl1

| № | Предикат | Действия | № перехода | Комментарий |
|---|---|---|---|---|
| 1 | | Возврат методом значения a1*x | ∅ | |

Класс объекта: Cl2

Модификатор доступа: public

Метод: Polynomial

Функционал: Параметризированный метод, вычисляющий значение многочлена второй степени от переданного в него значения и возвращающий полученный результат

Параметры: Целочисленный параметр x

Возвращаемое значение: Целочисленный тип данных - значение многочлена второй степени

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода Polynomial класса Cl2

| № | Предикат | Действия | № перехода | Комментарий |
|---|---|---|---|---|
| 1 | | Возврат методом значения a1*x + a2*x*x | ∅ | |

Класс объекта: Cl3

Модификатор доступа: public

Метод: Polynomial

Функционал: Параметризированный метод, вычисляющий значение многочлена третьей степени от переданного в него значения и возвращающий полученный результат

Параметры: Целочисленный параметр x

Возвращаемое значение: Целочисленный тип данных - значение многочлена третьей степени

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода Polynomial класса Cl3

| № | Предикат | Действия | № перехода | Комментарий |
|---|---|---|---|---|
| 1 | | Возврат методом значения a1*x + a2*x*x + a3*x*x*x | ∅ | |

Класс объекта: Cl4

Модификатор доступа: public

Метод: Polynomial

Функционал: Параметризированный метод, вычисляющий значение многочлена четвертой степени от переданного в него значения и возвращающий полученный результат

Параметры: Целочисленный параметр x

Возвращаемое значение: Целочисленный тип данных - значение многочлена четвертой степени

Алгоритм метода представлен в таблице 6.

Таблица 6. Алгоритм метода Polynomial класса Cl4

| № | Предикат | Действия | № перехода | Комментарий |
|---|---|---|---|---|
| 1 | | Возврат методом значения a1*x | ∅ | |

| | | | | |
|---|---|---|---|---|
| | | + a2*x*x + a3*x*x*x + a4*x*x*x*x | | |

Конструктор класса: Cl4

Модификатор доступа: public

Функционал: Параметризированный конструктор, принимающий в себя четыре числовых значения - коэффициентов многочлена и присваивающий их соответствующим полям всех классов.

Параметры: Целочисленные параметры a1, a2, a3, a4

Алгоритм конструктора представлен в таблице 7.

Таблица 7. Алгоритм конструктора класса Cl4

| № | Предикат | Действия | № перехода | Комментарий |
|---|---|---|---|---|
| 1 | | Присвоение целочисленным полям a1 классов Cl1, Cl2, Cl3, Cl4 значения a1 | 2 | |
| 2 | | Присвоение целочисленным полям a2 классов Cl2, Cl3, Cl4 значения a2 | 3 | |
| 3 | | Присвоение целочисленным полям a3 классов Cl3, Cl4 значения a3 | 4 | |
| 4 | | Присвоение целочисленному полю a4 класса Cl4 значения a4 | ∅ | |

## Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

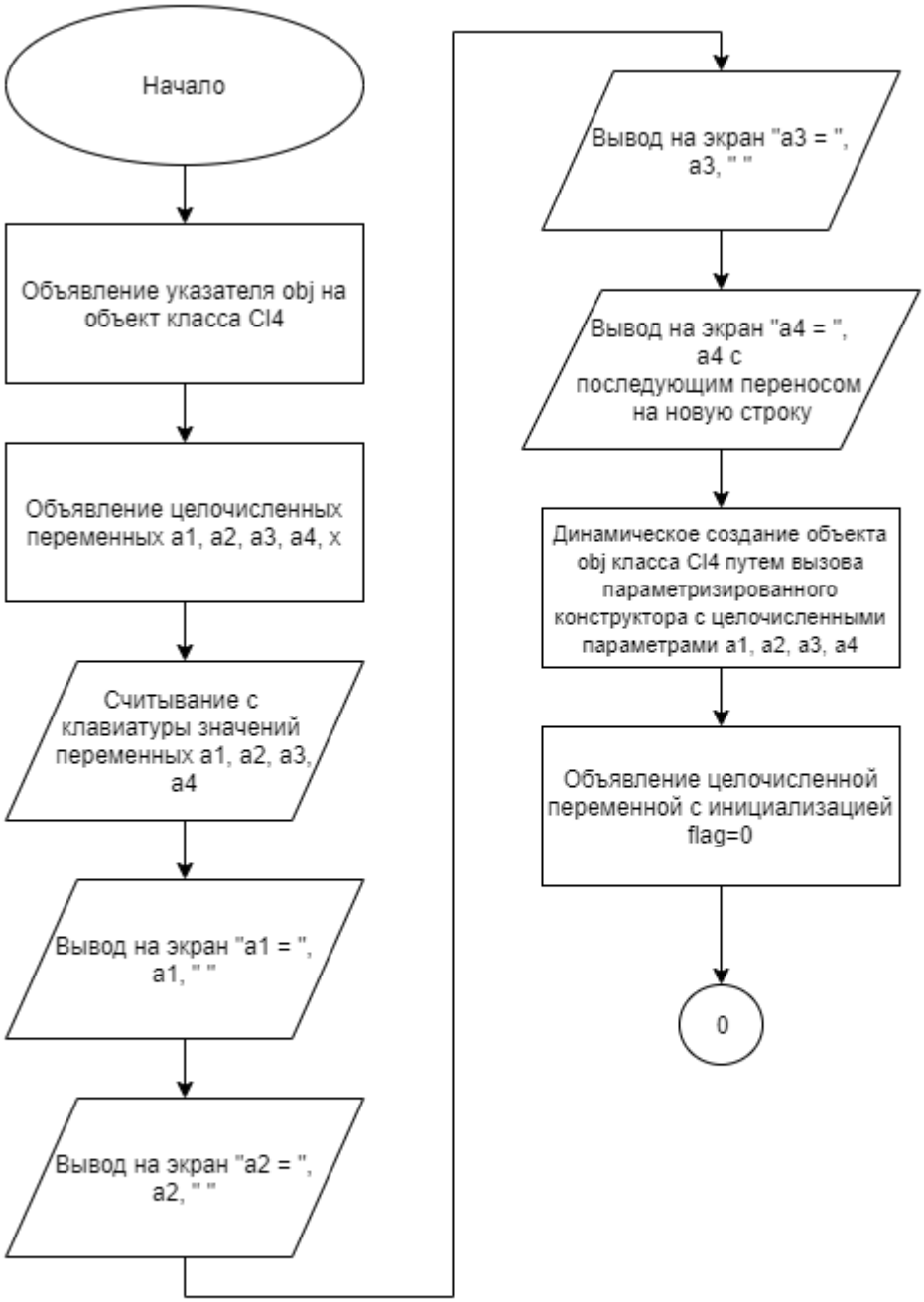


Рис. 1. Блок-схема алгоритма.

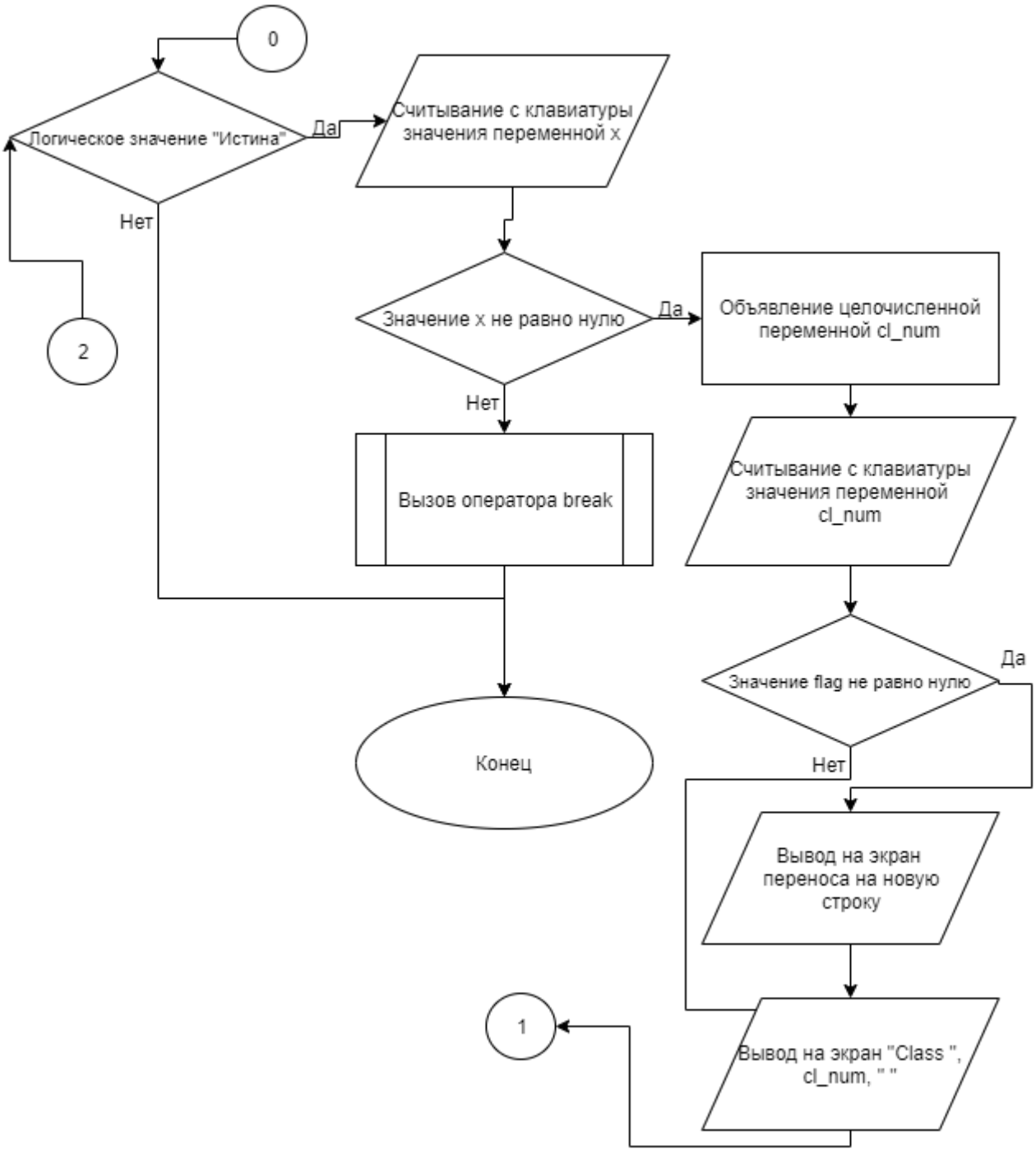


Рис. 2. Блок-схема алгоритма.

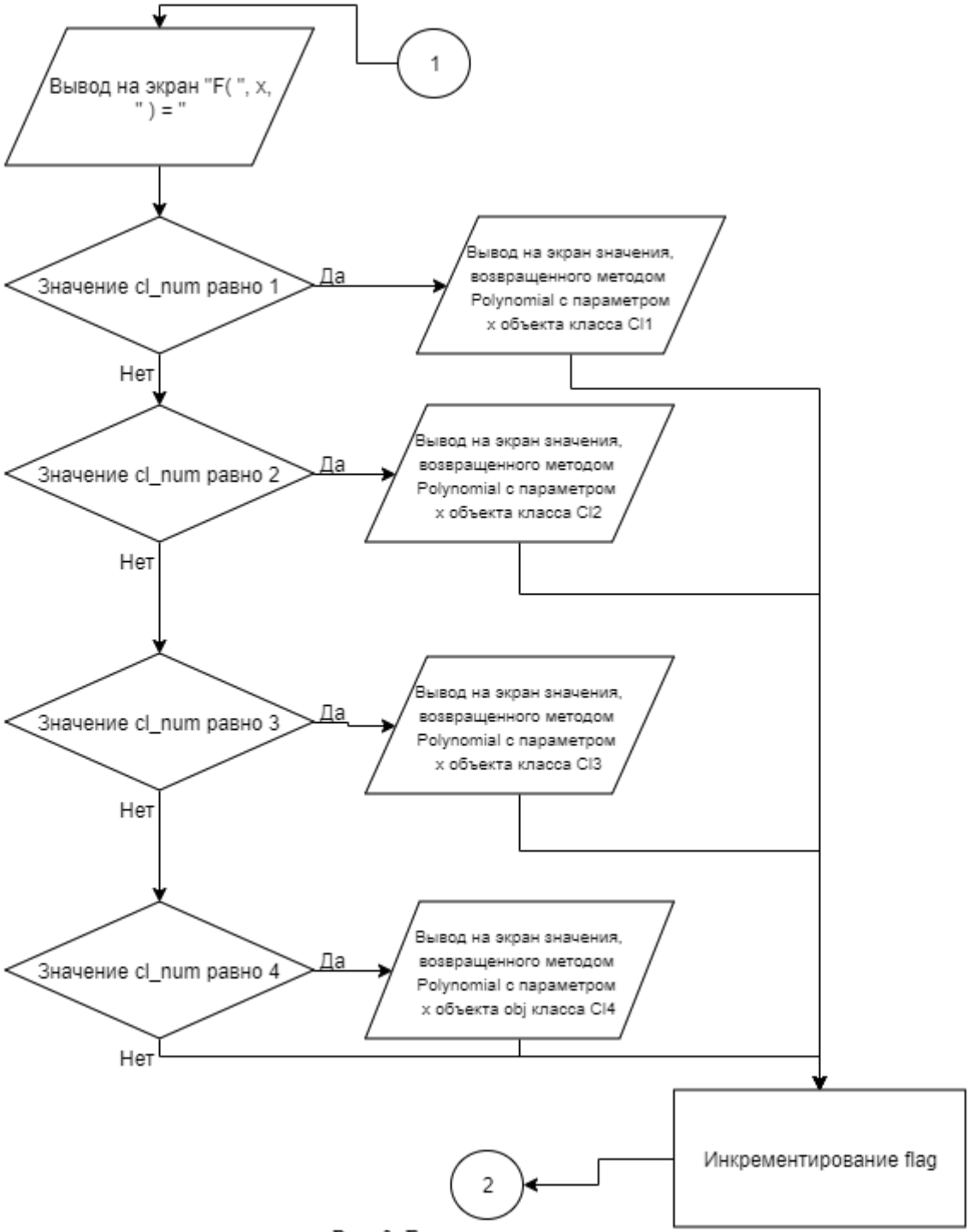


Рис. 3. Блок-схема алгоритма.


Рис. 4. Блок-схема алгоритма.

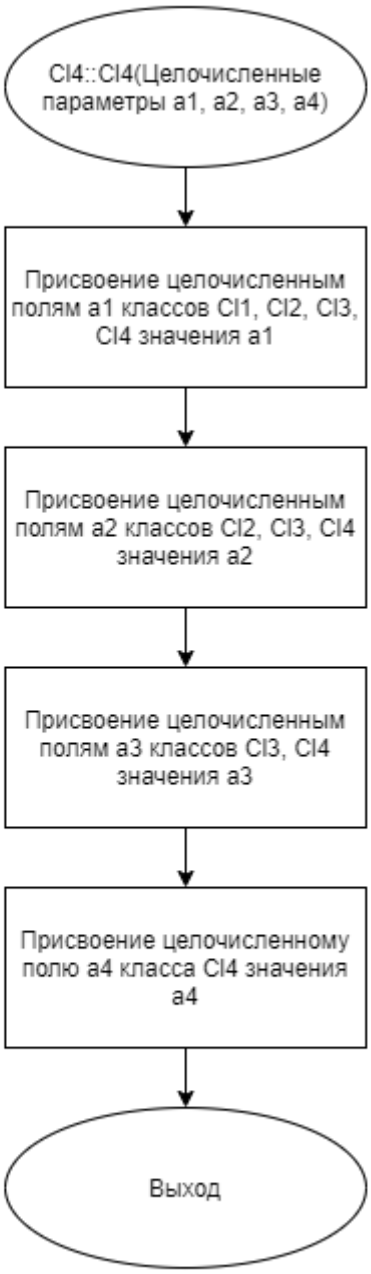


Рис. 5. Блок-схема алгоритма.

## Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

### Файл Cl1.cpp

```
#include <iostream>
using namespace std;

#include "Cl1.h"

int Cl1 :: Polynomial(int x){
        return a1*x;
}
```

### Файл Cl1.h

```
#ifndef CL1_H
#define CL1_H

class Cl1{
        protected:
                int a1;
        public:
                virtual int Polynomial(int x);
};

#endif
```

### Файл Cl2.cpp

```
#include <iostream>
using namespace std;

#include "Cl2.h"

int Cl2 :: Polynomial(int x){
        return a1*x + a2*x*x;
}
```

### Файл Cl2.h

```
#ifndef CL2_H
#define CL2_H

#include "Cl1.h"
class Cl2 : public Cl1{
        protected:
                int a1;
                int a2;
        public:
                virtual int Polynomial(int x) override;
};

#endif
```

## Файл Cl3.cpp

```
#include <iostream>
using namespace std;

#include "Cl3.h"

int Cl3 :: Polynomial(int x){
        return a1*x + a2*x*x + a3*x*x*x;
}
```

## Файл Cl3.h

```
#ifndef CL3_H
#define CL3_H

#include "Cl2.h"
class Cl3 : public Cl2{
        protected:
                int a1;
                int a2;
                int a3;
        public:
                virtual int Polynomial(int x) override;
};

#endif
```

## Файл Cl4.cpp

```
#include <iostream>
using namespace std;

#include "Cl4.h"

Cl4 :: Cl4(int a1, int a2, int a3, int a4){
        Cl1 :: a1 = a1;
        Cl2 :: a1 = a1;
        Cl3 :: a1 = a1;
        Cl4 :: a1 = a1;

        Cl2 :: a2 = a2;
        Cl3 :: a2 = a2;
        Cl4 :: a2 = a2;

        Cl3 :: a3 = a3;
        Cl4 :: a3 = a3;

        Cl4 :: a4 = a4;
}

int Cl4 :: Polynomial(int x){
        return a1*x + a2*x*x + a3*x*x*x + a4*x*x*x*x;
}
```

## Файл Cl4.h

```
#ifndef CL4_H
#define CL4_H

#include "Cl3.h"
class Cl4 : public Cl3{
        protected:
                int a1;
                int a2;
                int a3;
                int a4;
        public:
                Cl4(int a1, int a2, int a3, int a4);
                virtual int Polynomial(int x) override;
};

#endif
```

## Файл main.cpp

```
#include <iostream>
using namespace std;

#include "Cl1.h"
#include "Cl2.h"
```

```
#include "Cl3.h"
#include "Cl4.h"

int main(){
        Cl4* obj;
        int a1, a2, a3, a4, x;

        cin >> a1 >> a2 >> a3 >> a4;

        cout << "a1 = " << a1 << "    ";
        cout << "a2 = " << a2 << "    ";
        cout << "a3 = " << a3 << "    ";
        cout << "a4 = " << a4 << "\n";

        obj = new Cl4(a1, a2, a3, a4);

        int flag = 0;

        while (true){

                cin >> x;

                if (x != 0) {
                        int cl_num;

                        cin >> cl_num;

                        if (flag != 0) cout << "\n";

                        cout << "Class " << cl_num << "    ";
                        cout << "F( " << x << " ) = ";

                        if (cl_num == 1) cout << obj->Cl1 :: Polynomial(x);
                        else
                        if (cl_num == 2) cout << obj->Cl2 :: Polynomial(x);
                        else
                        if (cl_num == 3) cout << obj->Cl3 :: Polynomial(x);
                        else
                        if (cl_num == 4) cout << obj->Polynomial(x);

                }
                else break;
                flag++;
        }
        return 0;
}
```

## Тестирование

Результат тестирования программы представлен в следующей таблице.

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
|---|---|---|
| 1 1 1 1 1 1 1 2 1 3 1 4 0 | a1 = 1 a2 = 1 a3 = 1 a4 = 1 Class 1 F( 1 ) = 1 Class 2 F( 1 ) = 2 Class 3 F( 1 ) = 3 Class 4 F( 1 ) = 4 | a1 = 1 a2 = 1 a3 = 1 a4 = 1 Class 1 F( 1 ) = 1 Class 2 F( 1 ) = 2 Class 3 F( 1 ) = 3 Class 4 F( 1 ) = 4 |
| 4 3 2 1 1 4 2 3 3 2 4 1 0 | a1 = 4 a2 = 3 a3 = 2 a4 = 1 Class 4 F( 1 ) = 10 Class 3 F( 2 ) = 36 Class 2 F( 3 ) = 39 Class 1 F( 4 ) = 16 | a1 = 4 a2 = 3 a3 = 2 a4 = 1 Class 4 F( 1 ) = 10 Class 3 F( 2 ) = 36 Class 2 F( 3 ) = 39 Class 1 F( 4 ) = 16 |

# ЗАКЛЮЧЕНИЕ

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (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_laboratornyh_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).