



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

Дамарад Д.В.

Руководитель практики

Ассистент

Асадова Ю.С.

Работа представлена

«__»_____ 2022 г.

(подпись студента)

Оценка

(подпись руководителя)

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	
Постановка задачи.....	
Метод решения.....	
Описание алгоритма.....	
Блок-схема алгоритма.....	
Код программы.....	
Тестирование.....	
ЗАКЛЮЧЕНИЕ.....	
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	

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 a_1, a_2, a_3, a_4 , которые соответствуют коэффициентам многочлена ($a_1*x + a_2*x*x + a_3*x*x*x + a_4*x*x*x*x$).
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
 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 реализации данной задачи необходимо использовать:

1. Объекты стандартных потоков ввода и вывода `cin` и `cout`, используются для ввода с клавиатуры и вывода на экран.
2. Оператор цикла `while`.
3. Условный оператор `if`.
4. Объекты классов `Class1`, `Class2`, `Class3`, `Class4`.

Класс `Class1`:

- Поля:
 - Поле, хранящее значение первого коэффициента.
 - Наименование - `a1`;
 - Тип - целочисленный;
 - Модификатор доступа - `protected`.
- Методы:
 - Метод `gerRes()` - Вычисление значения многочлена и возврат полученного результата.

Класс `Class2`:

- Поля:
 - Поле, хранящее значение второго коэффициента.
 - Наименование - `a2`;
 - Тип - целочисленный;
 - Модификатор доступа - `protected`.
- Методы:
 - Метод `gerRes()` - Вычисление значения многочлена и возврат полученного результата.

Класс Class3:

- Поля:
 - Поле, хранящее значение третьего коэффициента.
 - Наименование - a3;
 - Тип - целочисленный;
 - Модификатор доступа - protected.
- Методы:
 - Метод gerRes() - Вычисление значения многочлена и возврат полученного результата.

Класс Class4:

- Поля:
 - Поле, хранящее значение четвертого коэффициента.
 - Наименование - a4;
 - Тип - целочисленный;
 - Модификатор доступа - protected.
- Методы:
 - Метод Class4 - параметризованный конструктор с целочисленными параметрами a1, a2, a3, a4.
 - Метод gerRes() - Вычисление значения многочлена и возврат полученного результата.

№	Наименование класса	Классы наследники	Модификатор доступа при наследовании	Описание	Номер	Комментарий
1	Class1			Вычисление первого значения многочлена		

		Class2	public		2	
2	Class2			Вычисление второго значения многочлена		
		Class3	public		3	
3	Class3			Вычисление третьего значения многочлена		
		Class4	public			
4	Class4			Вычисление четвертого значения многочлена и параметризированный конструктор.		

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: Целочисленное значение 0 - код возврата

Алгоритм функции представлен в таблице 2.

Таблица 2. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление указателя *ptr на объект класса Class4	2	
2		Объявление целочисленных переменных a1, a2, a3, a4, x, n	3	
3		Считывание значений переменных a1, a2, a3, a4 с клавиатуры	4	
4		Создание объекта obj класса Class4 посредством параметризованного конструктора с параметрами a1, a2, a3, a4	5	
5		Вывод на экран значений переменных a1, a2, a3, a4	6	
6	Истина	Считывание значения переменной x с клавиатуры	7	

			Ø	
7	x равно 0	Завершить цикл	Ø	
		Считывание с клавиатуры значения переменной n	8	
8	n равно 1	Вывод на экран значений переменных x, n и результата вызова метода getRes через указатель на класс Class1	6	
	n равно 2	Вывод на экран значений переменных x, n и результата вызова метода getRes через указатель на класс Class2	6	
	n равно 2	Вывод на экран значений переменных x, n и результата вызова метода getRes через указатель на класс Class3	6	
	n равно 2	Вывод на экран значений переменных x, n и результата вызова метода getRes через указатель на класс Class4	6	
		Вызов метода getRes через указатель на класс Class4	6	

Класс объекта: Class1

Модификатор доступа: public

Метод: getRes

Функционал: Вычисление значения первого многочлена

Параметры: Целочисленное значение переменной x

Возвращаемое значение: Целочисленное значение - первый многочлен

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода getRes класса Class1

№	Предикат	Действия	№ перехода	Комментарий
1		Возврат результата произведений значений параметра x и поля a1	∅	

Класс объекта: Class2

Модификатор доступа: public

Метод: getRes

Функционал: Вычисление значения второго многочлена

Параметры: Целочисленное значение переменной x

Возвращаемое значение: Целочисленное значение - второй многочлен

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода getRes класса Class2

№	Предикат	Действия	№ перехода	Комментарий
1		Возврат результата суммы метода getRes() класса Class1 и произведения значений параметра x, параметра x и поля a2	∅	

Класс объекта: Class3

Модификатор доступа: public

Метод: getRes

Функционал: Вычисление значения третьего многочлена

Параметры: Целочисленное значение переменной x

Возвращаемое значение: Целочисленное значение - третий многочлен

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода getRes класса Class3

№	Предикат	Действия	№ перехода	Комментарий
1		Возврат результата суммы метода getRes() класса Class2 и произведения значений параметра x, параметра x, параметра x и поля a3	∅	

Класс объекта: Class4

Модификатор доступа: public

Метод: getRes

Функционал: Вычисление значения четвертого многочлена

Параметры: Целочисленное значение переменной x

Возвращаемое значение: Целочисленное значение - четвертый многочлен

Алгоритм метода представлен в таблице 6.

Таблица 6. Алгоритм метода getRes класса Class4

№	Предикат	Действия	№ перехода	Комментарий
1		Возврат результата суммы метода getRes() класса Class4 и произведения значений параметра x, параметра x, параметра x, параметра x и поля a4	∅	

Конструктор класса: Class4

Модификатор доступа: public

Функционал: Параметризированный конструктор

Параметры: Целочисленные значения переменных a1, a2, a3, a4

Алгоритм конструктора представлен в таблице 7.

Таблица 7. Алгоритм конструктора класса Class4

№	Предикат	Действия	№ перехода	Комментарий
1		Присваивание полям a1, a2, a3, a4 значений параметров a1, a2, a3, a4	∅	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

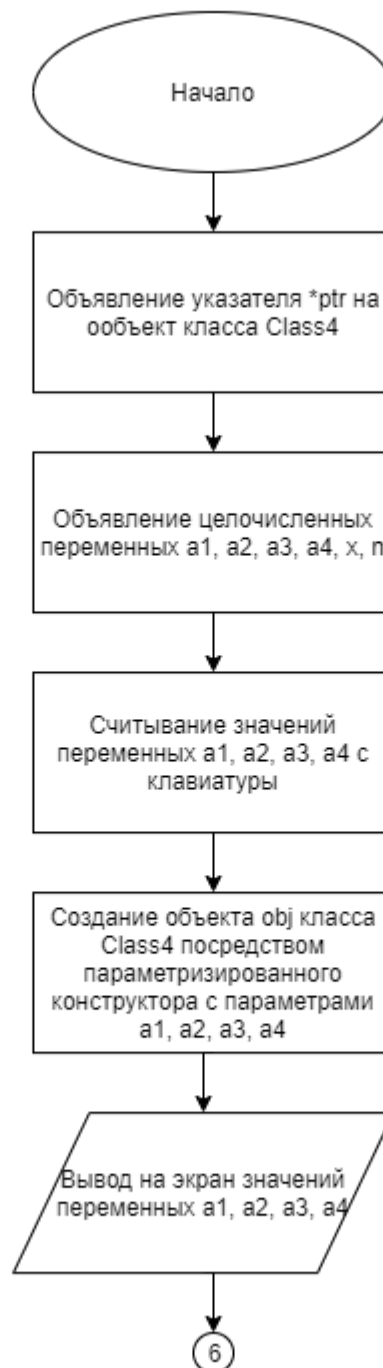


Рис. 1. Блок-схема алгоритма.

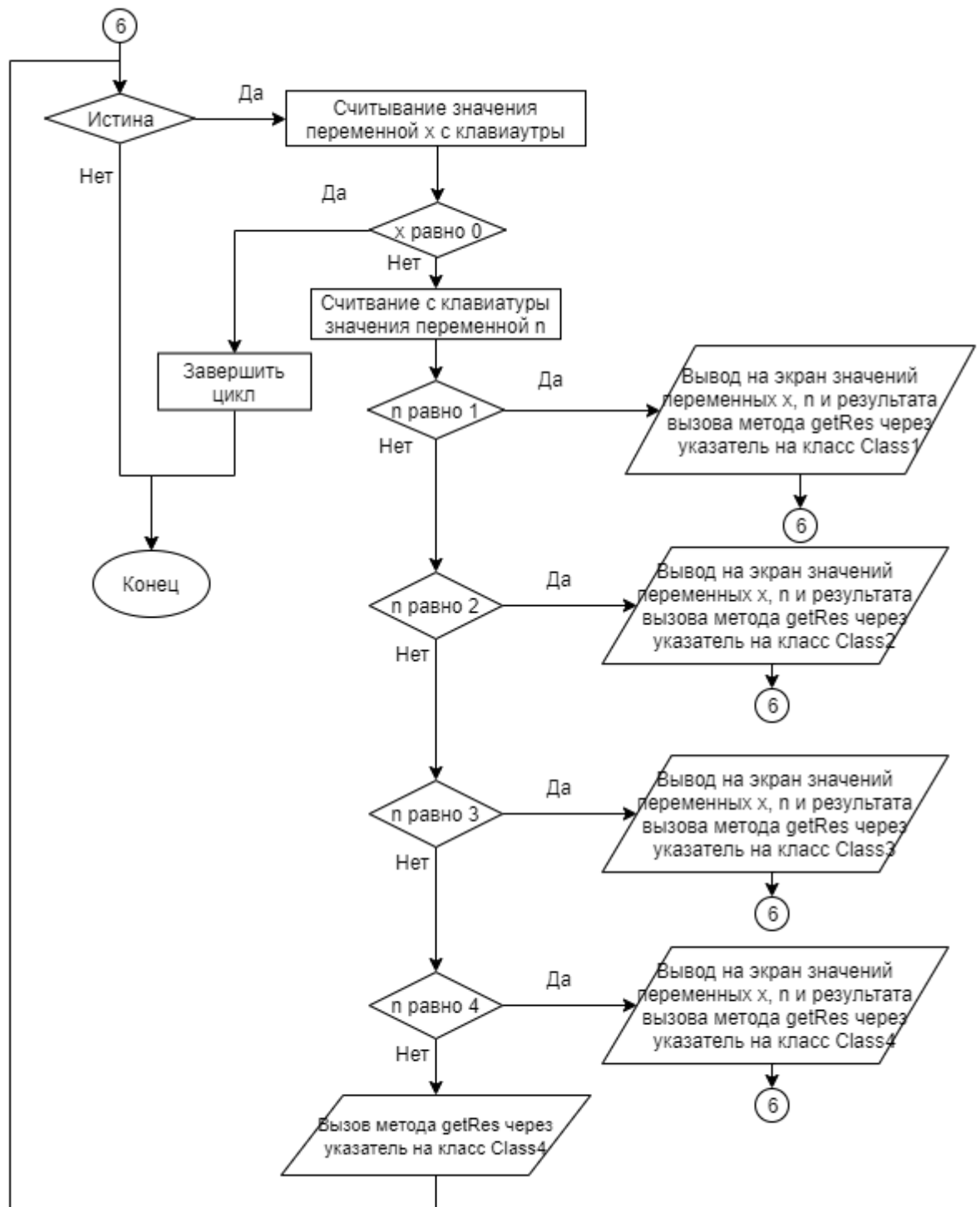


Рис. 2. Блок-схема алгоритма.

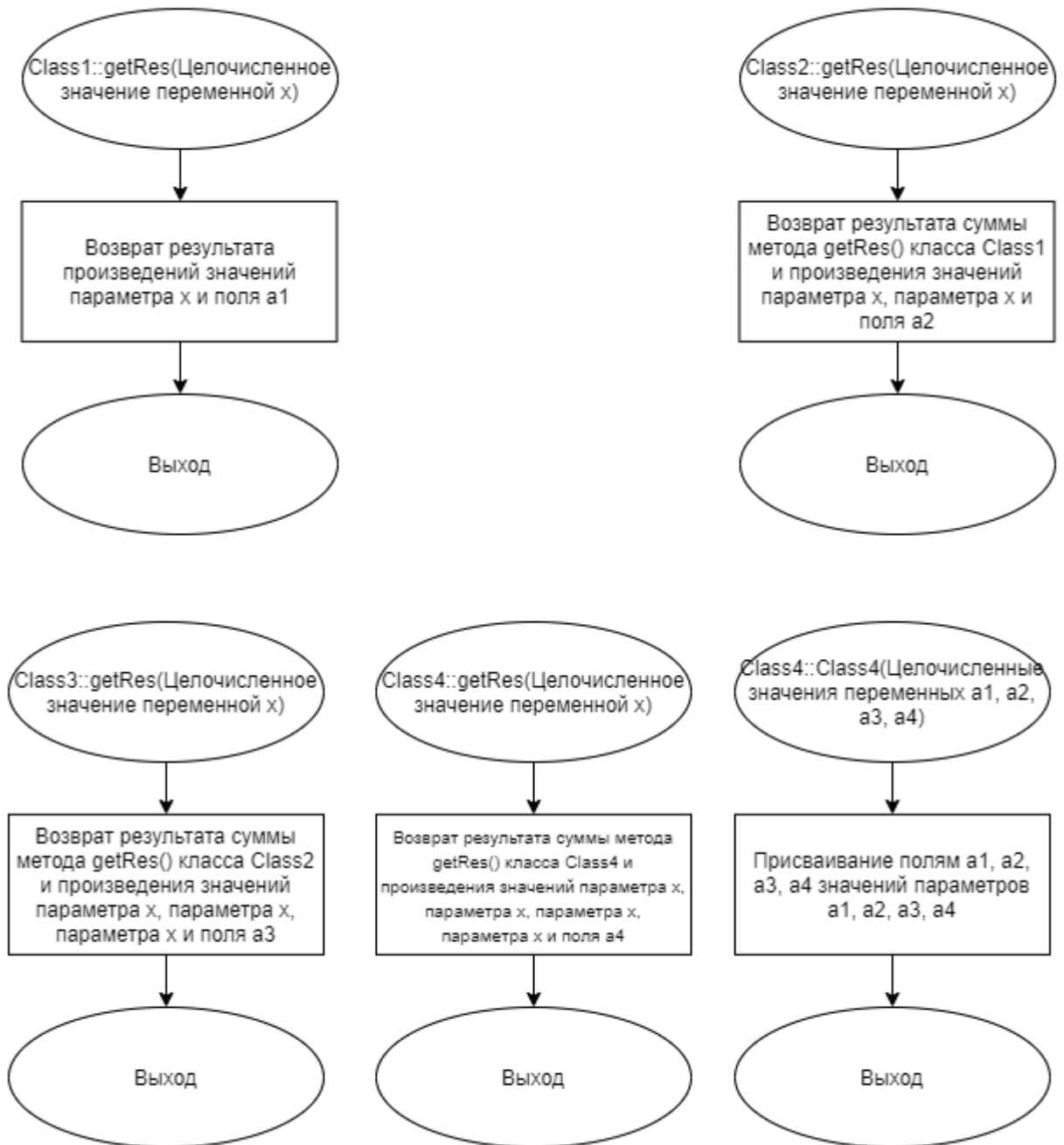


Рис. 3. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл Class1.cpp

```
#include "Class1.h"
int Class1::getRes(int x){
    return x*a1;
}
```

Файл Class1.h

```
#ifndef _CLASS1_H
#define _CLASS1_H
class Class1{
public:
    virtual int getRes(int x);
protected:
    int a1;
};
#endif
```

Файл Class2.cpp

```
#include "Class2.h"
int Class2::getRes(int x){
    return Class1::getRes(x)+x*x*a2;
}
```

Файл Class2.h

```
#ifndef _CLASS2_H
#define _CLASS2_H
#include "Class1.h"
class Class2: public Class1{
public:
```

```

        int getRes(int x) override;
    protected:
        int a2;
};
#endif

```

Файл Class3.cpp

```

#include "Class3.h"
int Class3::getRes(int x){
    return Class2::getRes(x)+x*x*x*a3;
}

```

Файл Class3.h

```

#ifndef _CLASS3_H
#define _CLASS3_H
#include "Class2.h"
class Class3: public Class2{
    public:
        int getRes(int x) override;
    protected:
        int a3;
};
#endif

```

Файл Class4.cpp

```

#include "Class4.h"
int Class4::getRes(int x){
    return Class3::getRes(x)+x*x*x*x*a4;
}
Class4::Class4(int a1, int a2, int a3, int a4){
    Class1::a1=a1;
    Class2::a2=a2;
    Class3::a3=a3;
    Class4::a4=a4;
}

```

Файл Class4.h

```

#ifndef _CLASS4_H
#define _CLASS4_H

```

```

#include "Class3.h"
class Class4: public Class3{
    public:
        int getRes(int x) override;
        Class4(int a1, int a2, int a3, int a4);
    protected:
        int a4;
};
#endif

```

Файл main.cpp

```

#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "Class1.h"
#include "Class2.h"
#include "Class3.h"
#include "Class4.h"
using namespace std;
int main()
{
    Class4 *ptr;
    int a1, a2, a3, a4, n, x;
    cin>>a1>>a2>>a3>>a4;
    ptr=new Class4(a1, a2, a3, a4);
    cout<<"a1 = "<<a1<<"    a2 = "<<a2<<"    a3 = "<<a3<<"    a4 = "<<a4;
    while (true){
        cin>>x;
        if (x==0){
            break;
        }
        else {
            cin>>n;
        }
        switch (n){
            case 1:
                cout<<"\nClass "<<n<<"    F( "<<x<<" ) =
                "<<ptr->Class1::getRes(x);
                break;
            case 2:
                cout<<"\nClass "<<n<<"    F( "<<x<<" ) =
                "<<ptr->Class2::getRes(x);
                break;
            case 3:
                cout<<"\nClass "<<n<<"    F( "<<x<<" ) =
                "<<ptr->Class3::getRes(x);
                break;
            case 4:
                cout<<"\nClass "<<n<<"    F( "<<x<<" ) =
                "<<ptr->Class4::getRes(x);

```

```
                break;
            default:
                break;
        }
        ptr->getRes(x);
    }
    return(0);
}
```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 3 3 7 1 1 0	a1 = 1 a2 = 3 a3 = 3 a4 = 7 Class 1 F(1) = 1	a1 = 1 a2 = 3 a3 = 3 a4 = 7 Class 1 F(1) = 1
1 2 3 4 1 1 1 2 1 3 1 4 0	a1 = 1 a2 = 2 a3 = 3 a4 = 4 Class 1 F(1) = 1 Class 2 F(1) = 3 Class 3 F(1) = 6 Class 4 F(1) = 10	a1 = 1 a2 = 2 a3 = 3 a4 = 4 Class 1 F(1) = 1 Class 2 F(1) = 3 Class 3 F(1) = 6 Class 4 F(1) = 10

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-avrrora.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-avrrora.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).