



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 4_1_2 »

С тудент группы

ИКБО-27-21

Родионов А.А.

Руководитель практики

Ассистент

Морозов В.А.

Работа представлена

«__»_____ 2022 г.

(подпись студента)

Оценка

(подпись руководителя)

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Постановка задачи.....	5
Метод решения.....	8
Описание алгоритма.....	12
Блок-схема алгоритма.....	18
Код программы.....	24
Тестирование.....	27
ЗАКЛЮЧЕНИЕ.....	28
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	29

ВВЕДЕНИЕ

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 параметризованный 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. Вводится идентификатор и натуральное число от 2 до 10.

2. Создать объект класса 4, используя параметризованный 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.

Описание входных данных

Первая

строка:

«идентификатор»«натуральное число»

Пример ввода:

Object 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

Метод решения

Для выполнения задачи нам потребуется:

- переменные типа integer
- переменные типа string
- модификаторы доступа public
- объекты ввода\вывода потока данных cin\cout (iostream)
- указатель this

№	Имя класса	Класссы наследники	Модификатор доступа при наследовании	Описание	Номер	Коментарий
1	C1	C2	public	Базовый класс	2	
2	C2	C3	public	Дочерний класс, а также базовый класс	3	
3	C3	C4	public	Дочерний класс, а также базовый класс	4	
4	C4		public	Дочерний класс		

Класс C1:

Поля:

- Наименование - name
- Тип - string

- Мод. доступа - public
- Функционал - хранит имя объекта
- Наименование - val
- Тип - int
- Мод. доступа - public
- Функционал - хранит данные объекта

Функционал:

- Наименование - print
- Тип - void
- Мод. доступа - public
- Функционал - выводит имя и данные объекта
- Наименование - C1
- Функционал - конструктор класса

Класс C2:

Поля:

- Наименование - name
- Тип - string
- Мод. доступа - public
- Функционал - хранит имя объекта
- Наименование - val
- Тип - int
- Мод. доступа - public
- Функционал - хранит данные объекта

Функционал:

- Наименование - print
- Тип - void
- Мод. доступа - public
- Функционал - выводит имя и данные объекта
- Наименование - C2
- Функционал - конструктор класса

Класс C3:

Поля:

- Наименование - name
- Тип - string
- Мод. доступа - public
- Функционал - хранит имя объекта
- Наименование - val
- Тип - int
- Мод. доступа - public
- Функционал - хранит данные объекта

Функционал:

- Наименование - print
- Тип - void
- Мод. доступа - public
- Функционал - выводит имя и данные объекта
- Наименование - C3
- Функционал - конструктор класса

Класс C4:

Поля:

- Наименование - name
- Тип - string
- Мод. доступа - public
- Функционал - хранит имя объекта
- Наименование - val
- Тип - int
- Мод. доступа - public
- Функционал - хранит данные объекта

Функционал:

- Наименование - print
- Тип - void
- Мод. доступа - public
- Функционал - выводит имя и данные объекта
- Наименование - C4
- Функционал - конструктор класса

Описание алгоритма

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

Конструктор класса: C1

Модификатор доступа: public

Функционал: конструктор класса

Параметры: string name, int val

Алгоритм конструктора представлен в таблице 2.

Таблица 2. Алгоритм конструктора класса C1

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация поля класса name, присваивая ему name + "_1"	2	
2		Инициализация поля класса val, присваивая ему val	Ø	

Класс объекта: C1

Модификатор доступа: public

Метод: print

Функционал: Выводит данные класса

Параметры: нет

Возвращаемое значение: void

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода print класса C1

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод значений полей класса в формате "(name) (val)"	Ø	

Конструктор класса: C2

Модификатор доступа: public

Функционал: конструктор класса

Параметры: string name, int val

Алгоритм конструктора представлен в таблице 4.

Таблица 4. Алгоритм конструктора класса C2

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация поля класса name, присваивая ему name + "_2"	2	
2		Инициализация поля класса val, присваивая ему val * val	Ø	

Класс объекта: C2

Модификатор доступа: public

Метод: print

Функционал: Выводит данные класса

Параметры: нет

Возвращаемое значение: void

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода print класса C2

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод значений полей класса в формате "(name) (val)"	Ø	

Конструктор класса: C3

Модификатор доступа: public

Функционал: конструктор класса

Параметры: string name, int val

Алгоритм конструктора представлен в таблице 6.

Таблица 6. Алгоритм конструктора класса C3

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация поля класса name, присваивая ему name + "_3"	2	
2		Инициализация поля класса val, присваивая ему val * val * val	Ø	

Класс объекта: C3

Модификатор доступа: public

Метод: print

Функционал: выводит данные класса

Параметры: нет

Возвращаемое значение: void

Алгоритм метода представлен в таблице 7.

Таблица 7. Алгоритм метода print класса C3

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод значений полей класса в формате "(name) (val)"	Ø	

Конструктор класса: C4

Модификатор доступа: public

Функционал: конструктор класса

Параметры: string name, int val

Алгоритм конструктора представлен в таблице 8.

Таблица 8. Алгоритм конструктора класса C4

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация поля класса name, присваивая ему name + "_4"	2	
2		Инициализация поля класса val, присваивая ему val * val * val * val	Ø	

Класс объекта: C4

Модификатор доступа: public

Метод: print

Функционал: выводит данные класса

Параметры: нет

Возвращаемое значение: void

Алгоритм метода представлен в таблице 9.

Таблица 9. Алгоритм метода print класса C4

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод значений полей класса в формате "(name) (val)"	Ø	

Функция: main

Функционал: главная функция программы

Параметры: нет

Возвращаемое значение: int, код возврата

Алгоритм функции представлен в таблице 10.

Таблица 10. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация переменной типа string string name	2	
2		Инициализация переменной типа int int val	3	
3		Ввод значения переменной name и val	4	
4		Инициализация объекта класса C1 C1 *obj = new C4(name, val)	5	

5		Вызов метода объекта созданного по унаследованному конструктору объекта obj ((C1*)obj)->print();	6	
6		Вызов метода объекта созданного по унаследованному конструктору объекта obj ((C2*)obj)->print();	7	
7		Вызов метода объекта созданного по унаследованному конструктору объекта obj ((C3*)obj)->print();	8	
8		Вызов метода объекта созданного по унаследованному конструктору объекта obj ((C4*)obj)->print();	9	
9		код возврата	10	
10			Ø	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

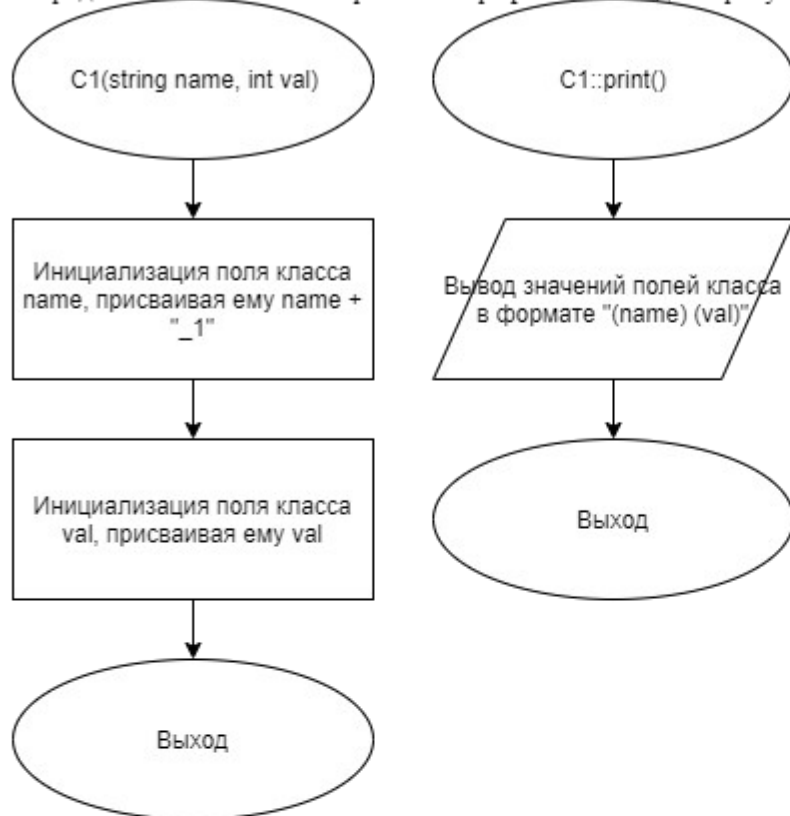


Рис. 1. Блок-схема алгоритма.

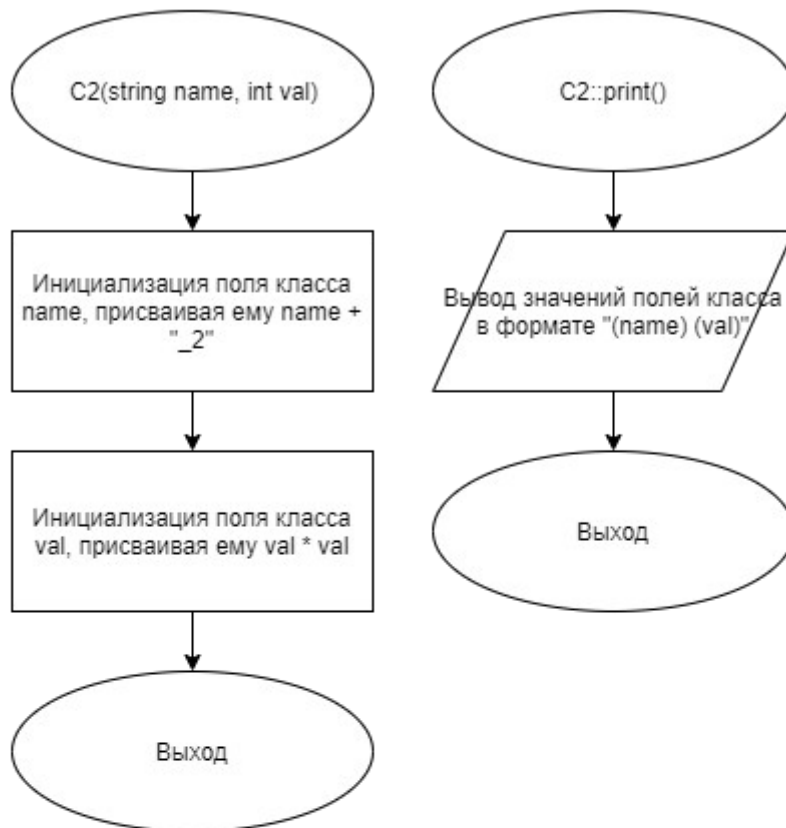


Рис. 2. Блок-схема алгоритма.

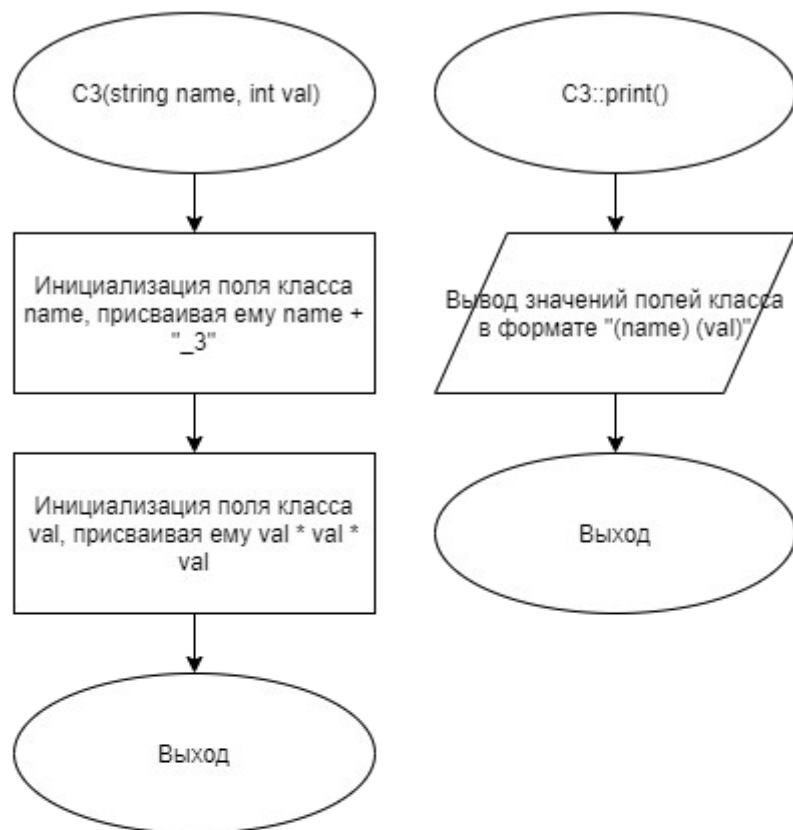


Рис. 3. Блок-схема алгоритма.

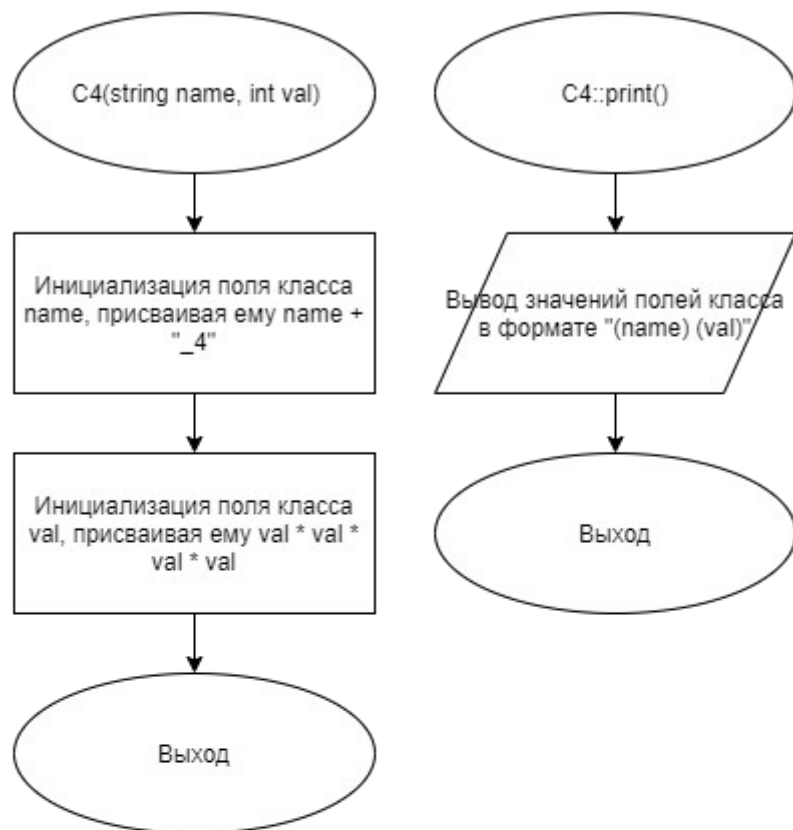


Рис. 4. Блок-схема алгоритма.

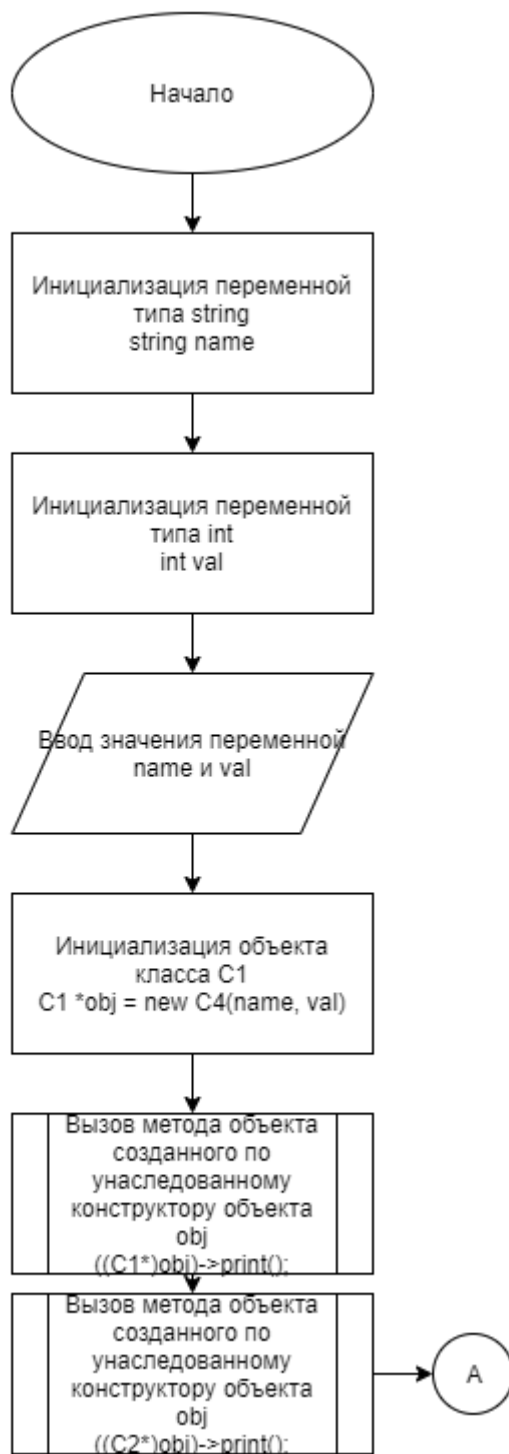


Рис. 5. Блок-схема алгоритма.

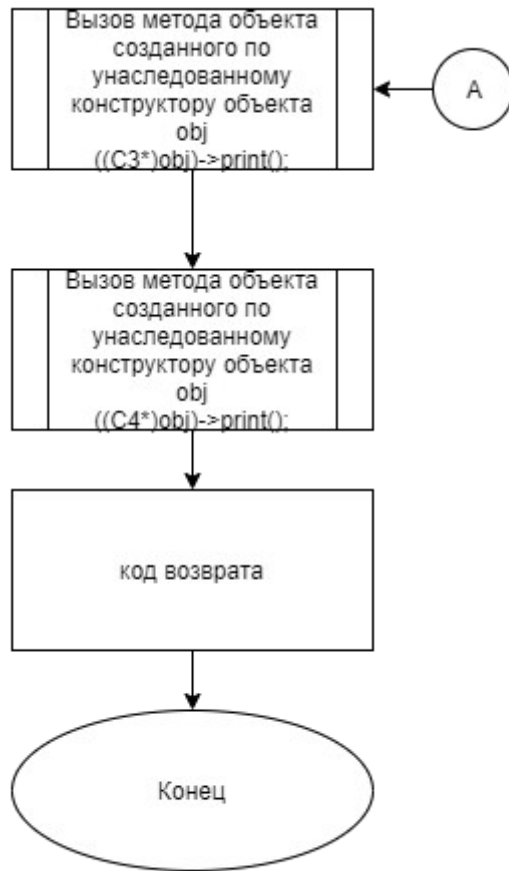


Рис. 6. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл c1.h

```
#ifndef CLASS_1
#define CLASS_1
#include <iostream>

using namespace std;

class C1{
    string name;
    int val;

    public:
        C1(string name, int val){
            this->name = name + "_1";
            this->val = val;
        };

        void print(){
            cout << name << " " << val << endl;
        };
};

#endif
```

Файл c2.h

```
#ifndef CLASS_2
#define CLASS_2
#include "c1.h"

class C2 : public C1{
    string name;
    int val;

    public:
        C2(string name, int val) : C1(name, val){
            this->name = name + "_2";
            this->val = val * val;
        };
};
```

```

        void print(){
            cout << name << " " << val << endl;
        };
};

#endif

```

Файл c3.h

```

#ifndef CLASS_3
#define CLASS_3
#include "c2.h"

class C3 : public C2{
    string name;
    int val;

public:
    C3(string name, int val) : C2(name, val){
        this->name = name + "_3";
        this->val = val * val * val;
    };

    void print(){
        cout << name << " " << val << endl;
    };
};

#endif

```

Файл c4.h

```

#ifndef CLASS_4
#define CLASS_4
#include "c3.h"

class C4 : public C3{
    string name;
    int val;

public:
    C4(string name, int val) : C3(name, val){
        this->name = name + "_4";
        this->val = val * val * val * val;
    };

    void print(){
        cout << name << " " << val;
    };
};

```



```
#endif
```

Файл main.cpp

```
#include <iostream>
#include "c4.h"

using namespace std;

int main()
{
    string name;
    int val;

    cin >> name >> val;

    C1 *obj = new C4(name, val);

    ((C1*)obj)->print();
    ((C2*)obj)->print();
    ((C3*)obj)->print();
    ((C4*)obj)->print();

    return(0);
}
```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
class 2	class_1 2 class_2 4 class_3 8 class_4 16	class_1 2 class_2 4 class_3 8 class_4 16
a 0	a_1 0 a_2 0 a_3 0 a_4 0	a_1 0 a_2 0 a_3 0 a_4 0

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).