

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.1 Алгоритм функции main.....	8
3.2 Алгоритм конструктора класса ArrayObject.....	9
3.3 Алгоритм метода print_array класса ArrayObject.....	10
3.4 Алгоритм деструктора класса ArrayObject.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	15
5.1 Файл ArrayObject.cpp.....	15
5.2 Файл ArrayObject.h.....	15
5.3 Файл main.cpp.....	16
6 ТЕСТИРОВАНИЕ.....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18

# 1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В открытом доступе имеется указатель на массив целого типа.

Конструктору объекта передается целочисленный параметр. Параметр должен иметь значение больше 4. По значению параметра определяется размерность целочисленного массива и каждому элементу присваивается это же значение.

Объект имеет функциональность, по которой выводит содержимое целочисленного массива. Вывод производит последовательно, разделяя значения двумя пробелами.

Спроектировать систему, которая содержит два объекта. Для построения системы последовательно, с новых строк вводятся целочисленные значения. Если очередное введенное значение меньше или равно 4, то создание системы прекращается, выводится сообщение и программа завершает работу. Если очередное введенное значение больше 4, то посредством параметризованного конструктора создается первый объект, а затем, второй.

Далее система функционирует по алгоритму:

1. ...
2. Первому объекту присвоить второй объект.
3. ...
4. С первой строки вывести содержимое массива первого объекта.
5. ...
6. Со второй строки вывести содержимое массива второго объекта.

## 1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число»

Пример.

5  
8

## 1.2 Описание выходных данных

Если система была построена, то в первой строке:

«Целое число» «Целое число» . . .

Во второй строке:

«Целое число» «Целое число» . . .

Если система не была построена, то в первой строке выводится некорректное значение и вопросительный знак:

«Целое число»?

**Пример вывода.**

5 5 5 5 5  
8 8 8 8 8 8 8 8

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект `first_object` класса `ArrayObject` предназначен для функционирования системы;
- объект `second_object` класса `ArrayObject` предназначен для функционирования системы;
- объект `set_second_object` класса `ArrayObject` метод предназначен для функционирования системы;
- объект `cout` потока вывода класса `cout` предназначен для функционирования системы;
- объект `cin` потока ввода класса `cin` предназначен для функционирования системы;
- оператор освобождения памяти `delete[]`;
- оператор выделения памяти `new`.

Класс `ArrayObject`:

- свойства/поля:
  - поле, которое хранит указатель на массив:
    - наименование — `array`;
    - тип — указатель на целочисленный массив;
    - модификатор доступа — `private`;
- функционал:
  - метод `ArrayObject` — Конструктор;
  - метод `~ArrayObject` — Деструктор;
  - метод `print_array` — Вывод значений.

## 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 функции main

Функционал: основной алгоритм работы программы.

Параметры: отсутствуют .

Возвращаемое значение: int - индикатор корректности завершения работы программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление целочисленных переменных input_one, input_two	2
2		ввод значения переменных input_one, input_two с клавиатуры	3
3	input_one <= 4	выводим "?", input_one	∅
			4
4		объявление объекта first_object класса ArrayObject с передачей параметров конструктору	5
5		создание объекта second_object класса ArrayObject с размером, указанным в переменной input_two	6
6		создание временной переменной, которая хранит в себе указатель на массив "array" в объекте "first_object"	7
7		копирование значений из "second_object" в	8

№	Предикат	Действия	№ перехода
		"first_object", включая указатель на массив	
8		установка указателя "array" объекта "first_object" на адрес, хранящийся в "temp_pointer"	9
9		вызывается метод print_array() для объекта first_object, выводящий содержимое его массива на экран	10
10		вызывается метод print_array() для объекта second_object, выводящий содержимое его массива на экран	∅

### 3.2 Алгоритм конструктора класса ArrayObject

Функционал: конструктор.

Параметры: int size.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса ArrayObject

№	Предикат	Действия	№ перехода
1		по адресу указателя создаем динамический массив размерностью size+1	2
2		инициализация целочисленной i = 0	3
3	i < size	i++, присваивание каждому элементу значение 'size'	3
			4
4		присвоение элементу массива "array" с индексом "size" значение "-1"	∅

### 3.3 Алгоритм метода print\_array класса ArrayObject

Функционал: вывод значения.

Параметры: отсутствуют.

Возвращаемое значение: отсутствуют.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода print\_array класса ArrayObject

№	Предикат	Действия	№ перехода
1		определение метода 'print_array()'	2
2		инициализация переменной i = 0	3
3	array[i] != -1	выводим текущий элемента массива 'array'	3
			4
4	array[i + 1] != -1	выводим пробел после элемента массива	4
			5
5		оператор endl	Ø

### 3.4 Алгоритм деструктора класса ArrayObject

Функционал: Деструктор.

Параметры: отсутствуют.

Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса ArrayObject

№	Предикат	Действия	№ перехода
1		освобождение памяти, выделенную для массива "array"	Ø



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

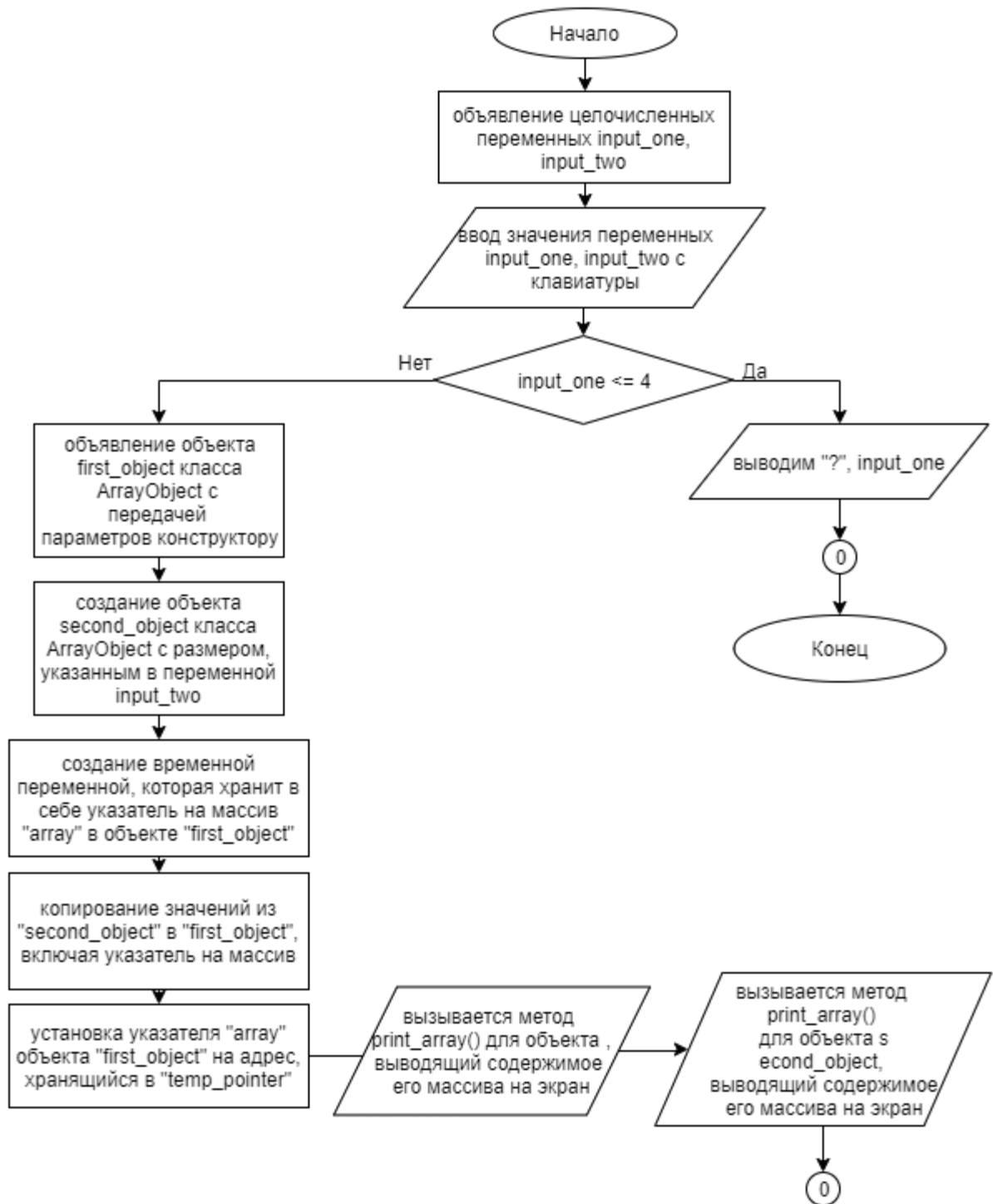
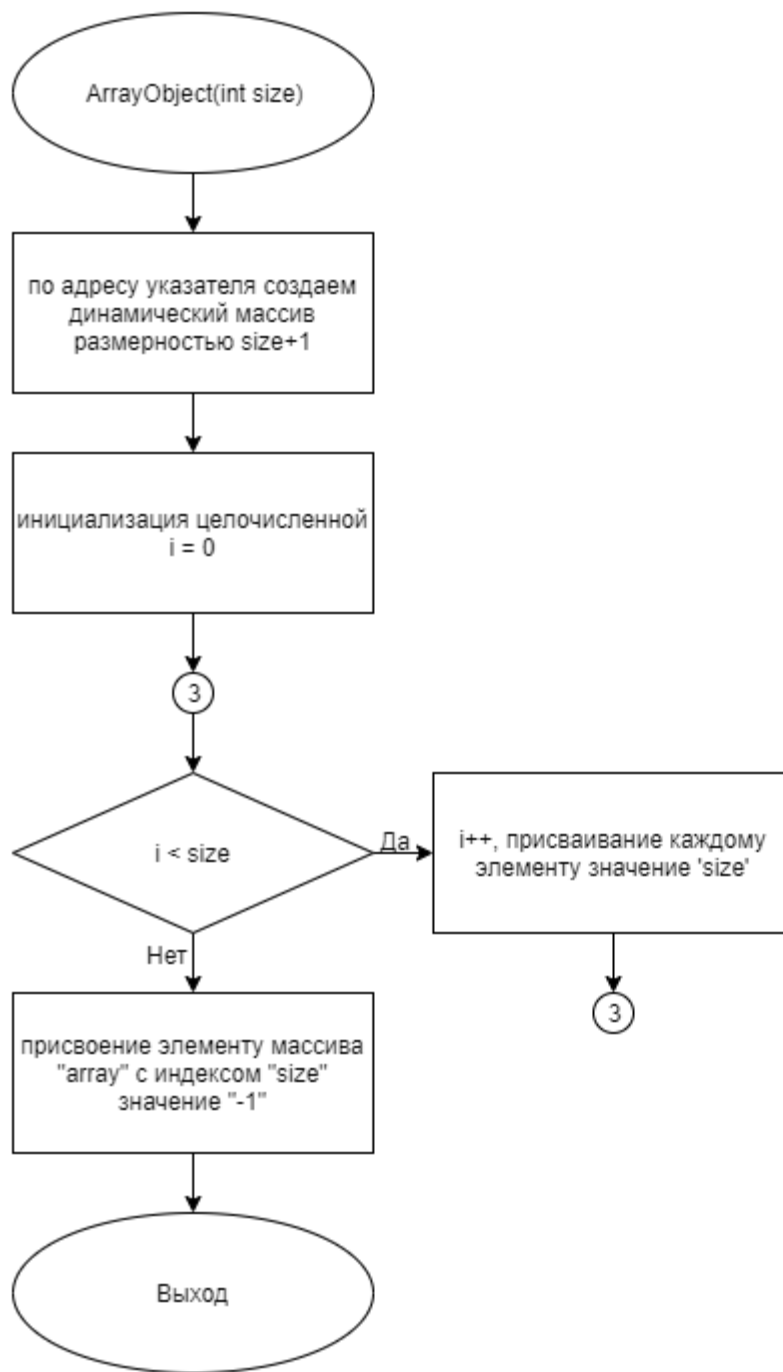


Рисунок 1 – Блок-схема алгоритма



**Рисунок 2 – Блок-схема алгоритма**

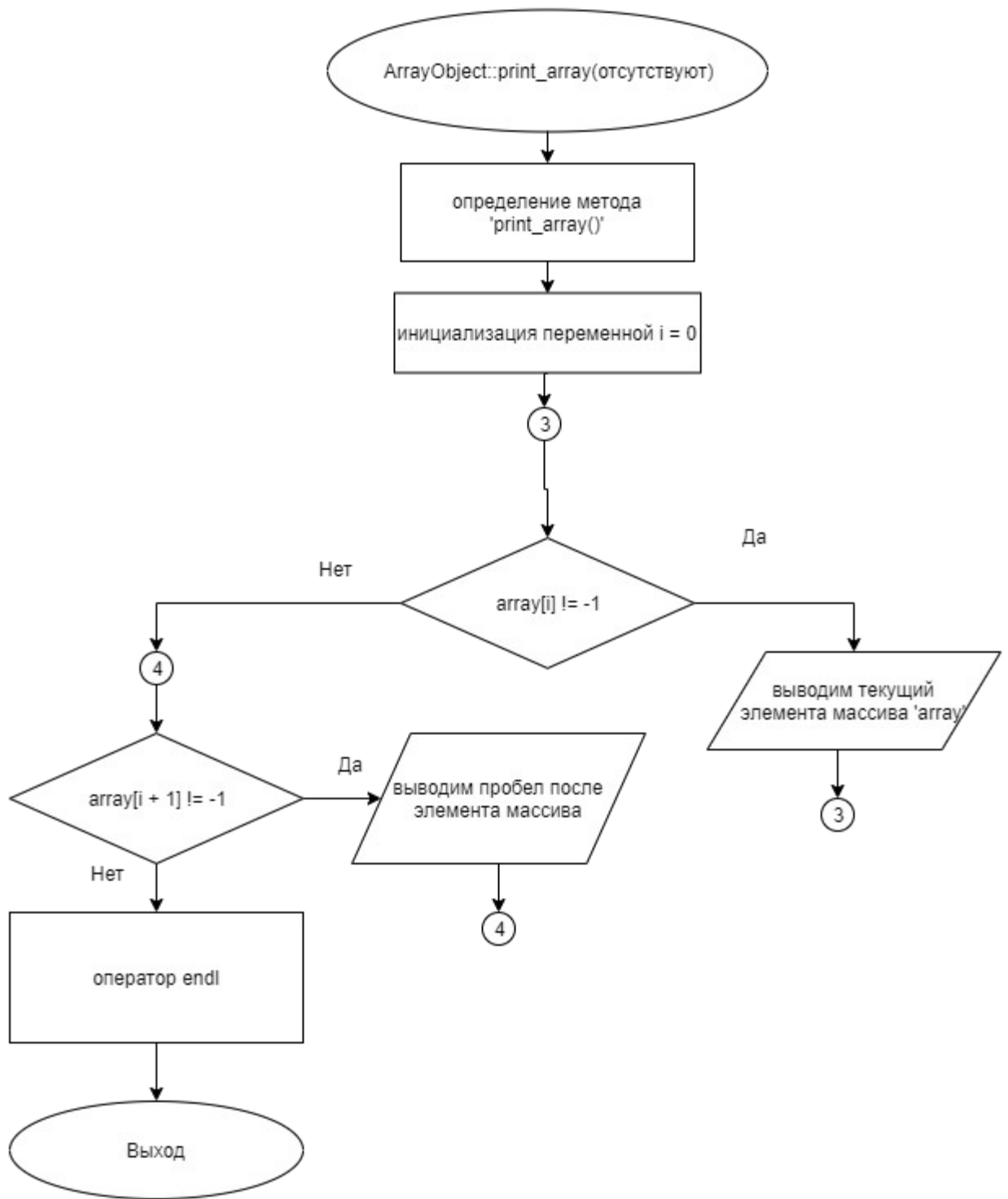


Рисунок 3 – Блок-схема алгоритма



**Рисунок 4 – Блок-схема алгоритма**

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл ArrayObject.cpp

*Листинг 1 – ArrayObject.cpp*

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

using namespace std;

ArrayObject::ArrayObject(int size){
    array = new int[size + 1];
    for(int i = 0; i < size; ++i) {
        array[i] = size;
    }
    array[size] = -1;
}

ArrayObject::~~ArrayObject() {
    delete[] array;
}

void ArrayObject::print_array() const {
    int i = 0;
    while(array[i] != -1) {
        cout << array[i];
        if(array[i + 1] != -1) {
            cout << " ";
        }
        ++i;
    }
    cout << endl;
}
```

### 5.2 Файл ArrayObject.h

*Листинг 2 – ArrayObject.h*

```
#ifndef __ARRAYOBJECT__H
```

```

#define __ARRAYOBJECT__H

using namespace std;

class ArrayObject {
public:
    int* array;
    ArrayObject(int size);
    ~ArrayObject();
    void print_array() const;
};

#endif

```

## 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```

#include "ArrayObject.h"
#include <iostream>
#include <stdlib.h>
#include <stdio.h>

using namespace std;

int main() {
    int input_one, input_two;
    cin >> input_one >> input_two;

    if(input_one <= 4 || input_two <= 4) {
        cout << (input_one <= 4 ? input_one : input_two) << "?";
        return 0;
    }

    ArrayObject first_object(input_one);
    ArrayObject second_object(input_two);

    int* temp_pointer = first_object.array;

    first_object = second_object;

    first_object.array = temp_pointer;

    first_object.print_array();
    second_object.print_array();

    return 0;
}

```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 8	5 5 5 5 5 8 8 8 8 8 8 8 8	5 5 5 5 5 8 8 8 8 8 8 8 8
5 8	5 5 5 5 5 8 8 8 8 8 8 8 8	5 5 5 5 5 8 8 8 8 8 8 8 8
5 8	5 5 5 5 5 8 8 8 8 8 8 8 8	5 5 5 5 5 8 8 8 8 8 8 8 8

## 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).