

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
3.0 Алгоритм функции main.....	8
3.1 Алгоритм конструктора класса cl.....	9
3.2 Алгоритм метода f1 класса cl.....	9
3.3 Алгоритм метода f2 класса cl.....	10
3.4 Алгоритм метода f3 класса cl.....	10
3.5 Алгоритм метода f4 класса cl.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	17
5.0 Файл main.cpp.....	17
6 ТЕСТИРОВАНИЕ.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	20

1 ПОСТАНОВКА ЗАДАЧИ

Спроектировать объект, в котором есть доступные и скрытые элементы (свойства и методы). Есть одно доступное свойство целого типа и одно скрытое свойство целого типа.

У объекта есть параметризованный конструктор с параметром целого типа. В конструкторе доступному свойству присваивается значение параметра, а скрытому свойству утроенное значение параметра.

У объекта есть доступные методы со следующим функционалом:

- метод изменения значения доступного и скрытого свойства. Доступному свойству добавляется 4, скрытому свойству добавляется 1;
- метод вызова закрытого метода;
- метод вывода состояния, выводит значение доступного и скрытого свойства.

У объекта есть один скрытый метод со следующим функционалом:

- метод доступному свойству добавляется 7, скрытому свойству добавляется 5.

Написать программу, которая состоит из описания класса выше представленного объекта и основной функции, в которой реализован следующий алгоритм:

1. Ввод целочисленного значения переменной `i_data`.
2. Создание объекта посредством оператора функции `new` и использованием указателя на объект, параметризованному конструктору в качестве аргумента передается переменная `i_data`.
3. Вывод исходного состояния объекта.
4. Вызов метода изменения значений свойств объекта.

5. Вывод текущего состояния объекта.
6. Ввод целочисленного значения переменной `i_data`.
7. Если значение `i_data` больше, чем значение доступного свойства объекта,
то
 - 7.1. Непосредственное изменение доступного свойства объекта посредством значения выражения `i_data * 8` и переход к пункту 9.
8. Иначе
 - 8.1. Переход к пункту 9.
9. Вывод текущего состояния объекта.
10. Вызов метода объекта, который вызывает скрытый метод объекта.
11. Вывод текущего состояния объекта.

1.1 Описание входных данных

Первая строка

«Целочисленное значение»

Вторая строка

«Целочисленное значение»

1.2 Описание выходных данных

Метод вывода состояния, первый вывод делает в первой строке, а далее с новой. Шаблон вывода:

Value of the available property «значение доступного свойства»; Value of a hidden property «значение закрытого свойства»

2 МЕТОД РЕШЕНИЯ

Для решения задачи понадобится:
используется измененный класс из задачи 2.2.1
условный оператор if
используется операторы функции new

Класс cl:

Поля:

доступный элемент i_data_open

скрытый элемент i_data_close

Методы

открытые:

f1() - действия над элементами

f2() - вызов метода f4()

f3() - вывод надписи на экран

закрытые

f4() - действия над элементами

3 ОПИСАНИЕ АЛГОРИТМОВ

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

3.0 Алгоритм функции main

Функционал: главный метод программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Ввод целочисленного значения переменной i_data.	2
2		Создание объекта посредством оператора функции new и использованием указателя на объект, параметризованному конструктору в качестве аргумента передается переменная i_data.	3
3		Вывод исходного состояния объекта.	4
4		Вызов метода изменения значений свойств объекта.	5
5		Вывод текущего состояния объекта.	6
6		Ввод целочисленного значения переменной i_data.	7
7	значение i_data больше, чем значение доступного свойства объекта	Непосредственное изменение доступного свойства объекта посредством значения выражения i_data * 8	8
			8
8		Вывод текущего состояния объекта.	9
9		Вызов метода объекта, который вызывает скрытый	10

№	Предикат	Действия	№ перехода
		метод объекта.	
1 0		Вывод текущего состояния объекта	∅

3.1 Алгоритм конструктора класса *cl*

Функционал: действия над доступным и скрытым элементами.

Параметры: int i_data.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *cl*

№	Предикат	Действия	№ перехода
1		доступный элемент присваивается значение параметра	2
2		скрытый элемент утроенное значение параметра	∅

3.2 Алгоритм метода *f1* класса *cl*

Функционал: действия над доступным и скрытым элементами.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *f1* класса *cl*

№	Предикат	Действия	№ перехода
1		доступный элемент добавляется 4	2
2		скрытый элемент добавляется 1	∅

3.3 Алгоритм метода f2 класса cl

Функционал: вызов скрытого метода.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода f2 класса cl

№	Предикат	Действия	№ перехода
1		вызов скрытого метода	Ø

3.4 Алгоритм метода f3 класса cl

Функционал: вывод значение доступного и скрытого элемента.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода f3 класса cl

№	Предикат	Действия	№ перехода
1		объявление счетчика	2
2	счетчик меньше 3	выводим значения на экран с переносом	Ø
		выводим значения на экран без переноса	Ø

3.5 Алгоритм метода f4 класса cl

Функционал: действия над доступным и скрытым элементами.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода f_4 класса cl

№	Предикат	Действия	№ перехода
1		к доступному элементу добавляется 7	2
2		к скрытому элементу добавляется 5	\emptyset

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

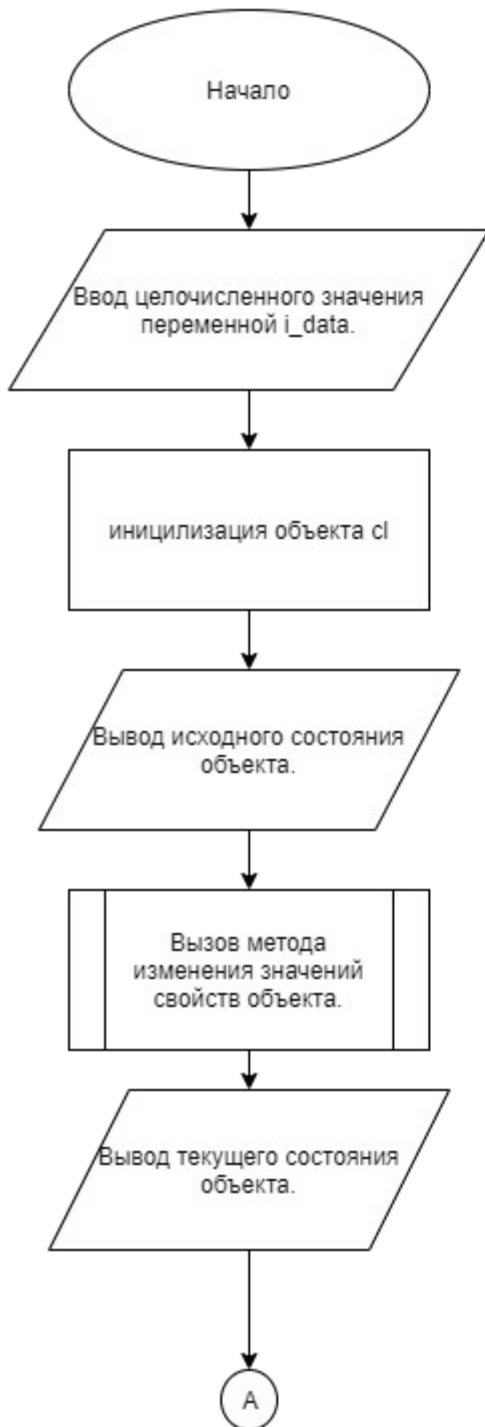


Рисунок 1 – Блок-схема алгоритма

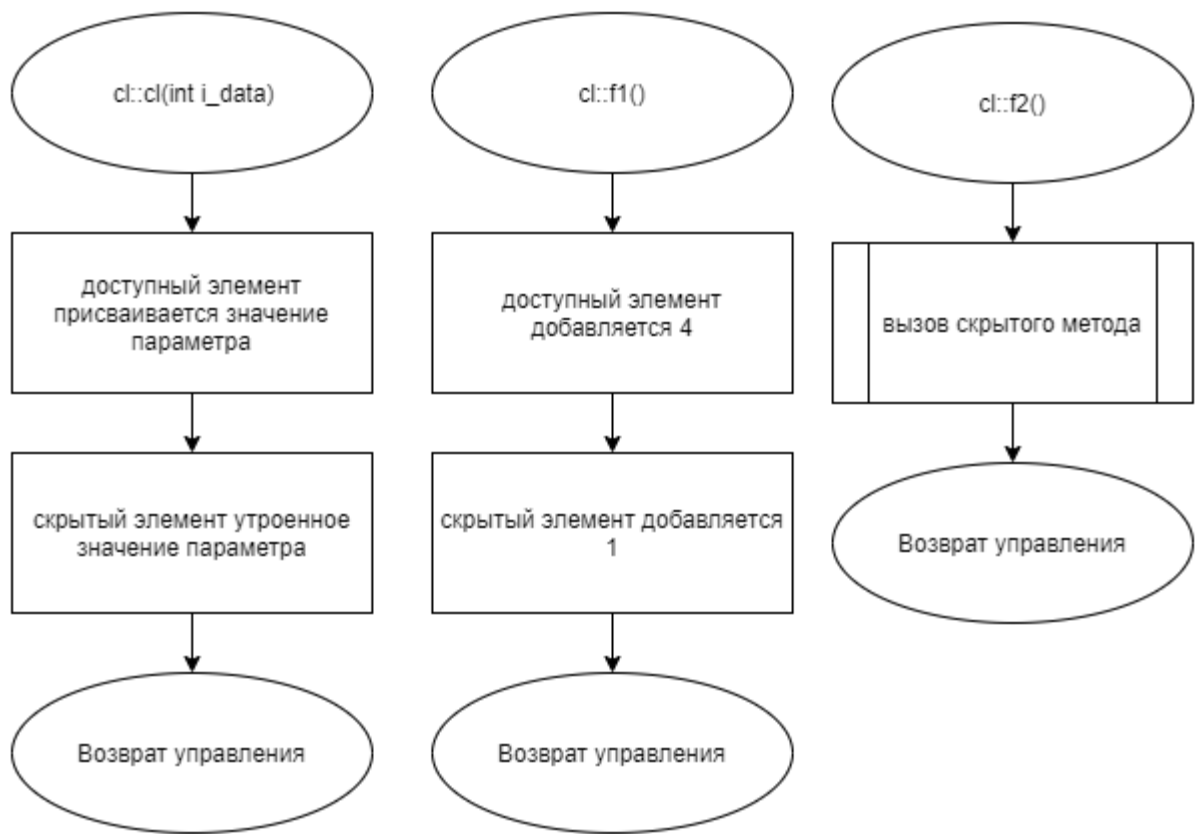


Рисунок 2 – Блок-схема алгоритма

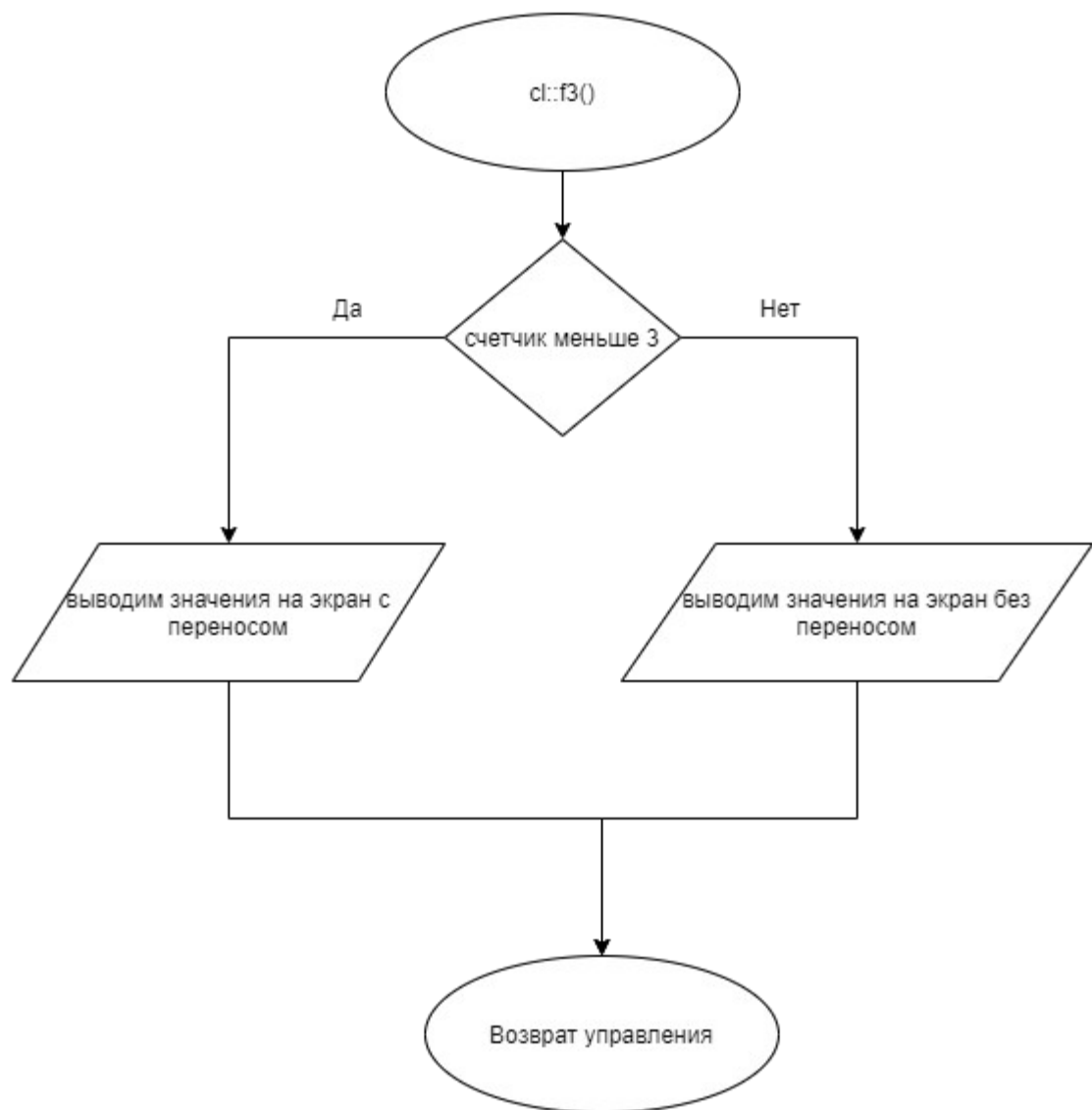


Рисунок 3 – Блок-схема алгоритма



Рисунок 4 – Блок-схема алгоритма

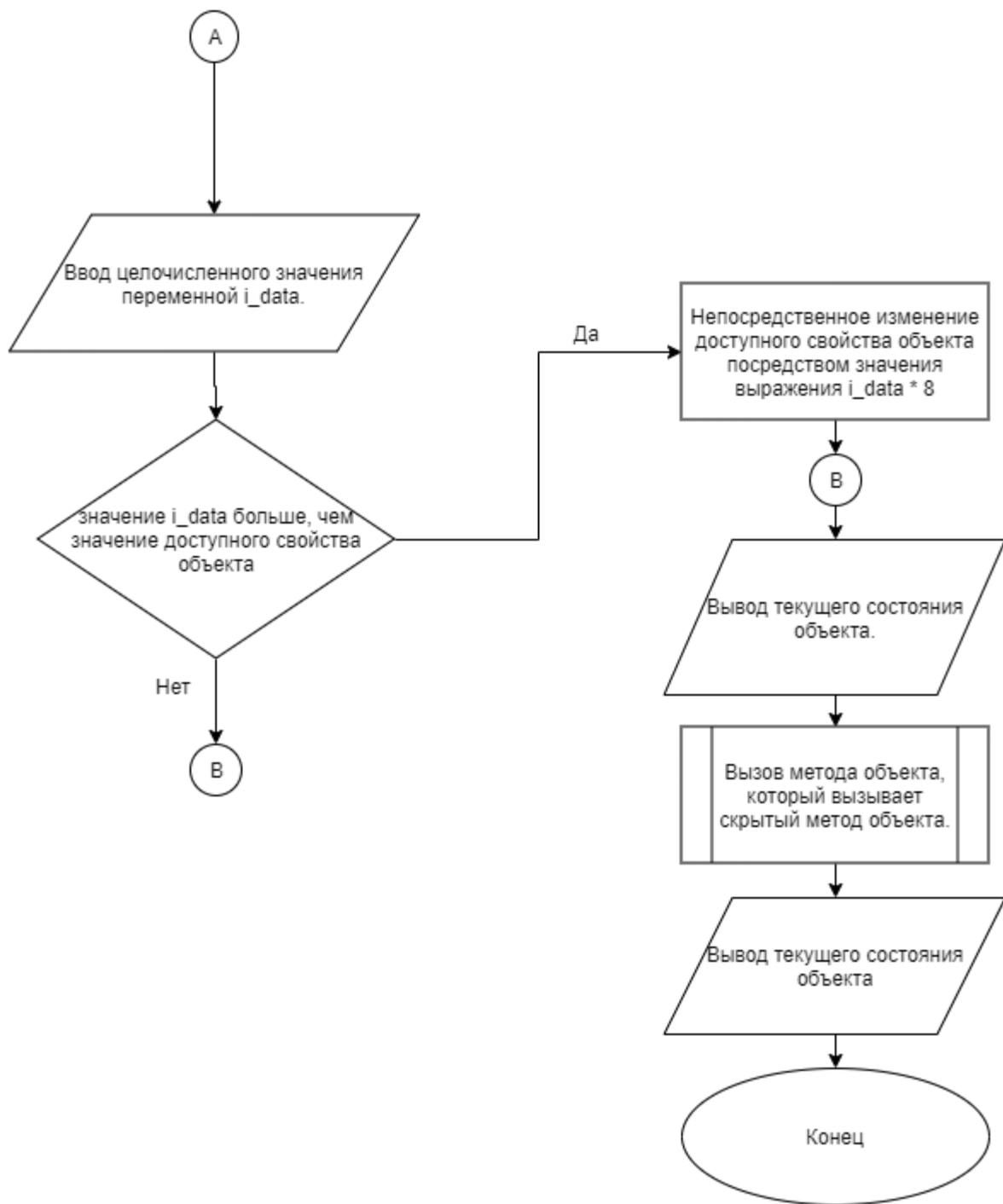


Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.0 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>

using namespace std;

class cl
{
private:
    int i_data;
    int i_data_close;
    void f4();
    int counter = 0;
public:
    cl(int i_data);
    void f1();
    void f2();
    void f3();
    int i_data_open;
};

cl::cl(int i_data)
{
    i_data_open = i_data;
    i_data_close = i_data * 3;
}

void cl::f1()
{
    i_data_open += 4;
    i_data_close += 1;
}

void cl::f2()
{
    f4();
}

void cl::f3()
{
    if (counter < 3)
```

```

        {
            cout << "Value of the available property " << i_data_open << "; Value
of a hidden property " << i_data_close;
            cout << endl;
        }
        else
        {
            cout << "Value of the available property " << i_data_open << "; Value
of a hidden property " << i_data_close;
        }
        counter++;
        //cout << "Value of the available property " << i_data_open << "; Value of a
hidden property " << i_data_close;
        //cout << endl;
    }

void cl::f4()
{
    i_data_open += 7;
    i_data_close += 5;
}

int main()
{
    int i_data(0);
    cin >> i_data;
    cl* object = new cl(i_data);

    object -> f3();
    object -> f1();
    object -> f3();

    cin >> i_data;

    if (i_data > object -> i_data_open)
    {
        object -> i_data_open = i_data * 8;
    }

    object -> f3();
    object -> f2();
    object -> f3();

    return(0);
}

```


6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2	Value of the available property 2; Value of a	Value of the available property 2; Value of a
2	hidden property 6	hidden property 6
	Value of the available property 6; Value of a	Value of the available property 6; Value of a
	hidden property 7	hidden property 7
	Value of the available property 6; Value of a	Value of the available property 6; Value of a
	hidden property 7	hidden property 7
	Value of the available property 13; Value of a	Value of the available property 13; Value of a
	hidden property 12	hidden property 12

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на С++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. С++: 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-avroora.ru/student/files/methodicheskoe_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-avroora.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).