



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:

**« Задача 4\_3\_1 »**

С тудент группы

ИКБО-27-21

Шевелёв И.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.....	7
Описание алгоритма.....	9
Блок-схема алгоритма.....	13
Код программы.....	17
Тестирование.....	20
ЗАКЛЮЧЕНИЕ.....	21
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	22

## **ВВЕДЕНИЕ**

## Постановка задачи

Создать базовый класс, содержащий только свойства. На его базе создать производный класс 1 и производный класс 2, содержащие только методы. В первом производном классе реализовать метод ввода размерности целочисленного массива и значений элементов массива. Во втором классе реализовать метод упорядочения целочисленного массива по не убыванию и метод вывода массива.

Посредством множественного наследования классов 1 и 2 создать класс 3. В классе 3 реализовать метод с алгоритмом:

1. Вызов метода ввода целочисленного массива.
2. Вызов метода вывода массива.
3. Вызов метода упорядочения целочисленного массива.
4. Вызов метода вывода массива.

## Описание входных данных

**Первая строка:**

«размерность целочисленного массива»

Размерность массива натуральное число больше или равно 1 и меньше или равно 100.

**Вторая строка:**

«последовательность целых чисел»

Количество целых чисел равно размерности массива и разделены пробелами. Значение каждого числа принадлежит интервалу  $[-100, 100]$ .

## Описание выходных данных

**Первая строка:**

Array dimension: «размерность целочисленного массива»

**Вторая**

**строка:**

The original array: «последовательность целых чисел»

Для вывода каждого числа выделяется поле из 5 позиции.

**Третья**

**строка:**

An ordered array: «последовательность целых чисел»

Для вывода каждого числа выделяется поле из 5 позиции.

## Метод решения

Ввод размерности массива, ввод его элементы, сортировка и вывод.

Основная программа:

Объект prog класса class\_3

Класс base:

Поля

Модификатор доступа protected

Указатель mas

Целочисленный тип данных int

Целочисленная переменная n

Класс class\_1:

Поля

Объект ввода/вывода (библиотека <iostream>)

Целочисленная переменная input

Модификатор доступа public

Объявление функции put()

Тип данных void

Цикл for

Массив

Класс class\_2 :

Поля

Объект ввода/вывода (библиотека <iostream>)

Целочисленная переменная temp

Модификатор доступа public

Объявление функции sort(), output()

Тип данных void

Цикл for

Условный оператор if

Массив

функция setw (библиотеки <iomanip>)

Класс class\_3 :

Поля

Модификатор доступа public

Объявление конструктора

Вызов функций

## Описание алгоритма

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

Класс объекта: class\_1

Модификатор доступа: public

Метод: put

Функционал: Ввод значений элементов массива

Параметры: нет

Возвращаемое значение: void, ничего

Алгоритм метода представлен в таблице 1.

Таблица 1. Алгоритм метода put класса class\_1

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной input	2	
2		Ввод значения свойства n класса base	3	
3		Присвоение указателю mas количество элементов массива n	4	
4	i < n	i++	5	
			7	Иначе
5		Ввод значения input	6	
6		Присвоению элементов массива mas значение input	4	
7		Вывод "Array dimension: " n	Ø	

Класс объекта: class\_2



Модификатор доступа: public

Метод: sort

Функционал: Соортировка методом пузырьком

Параметры: нет

Возвращаемое значение: void, ничего

Алгоритм метода представлен в таблице 2.

Таблица 2. Алгоритм метода sort класса class\_2

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной temp	2	
2	$i < n - 1$	$i++$	3	
			Ø	Иначе
3	$j < n - i - 1$	$j++$	4	
			2	Иначе
4	$mas[j] > mas[j + 1]$		5	
			3	
5		Присвоение temp значение элемента массива mas[j]	6	
6		Присвоение элементу массива mas[j] элемент mas[j+1]	7	
7		Присвоение элементу массива mas[j] значение temp	3	

Класс объекта: class\_2

Модификатор доступа: public

Метод: output

Функционал: Вывод элементов массива

Параметры: нет

Возвращаемое значение: void, ничего

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода output класса class\_2

№	Предикат	Действия	№ перехода	Комментарий
1	i < n	i++	2	
			Ø	Иначе
2		Вывод setw(5), mas[i]	1	

Конструктор класса: class\_3

Модификатор доступа: public

Функционал: Конструктор

Параметры: нет

Алгоритм конструктора представлен в таблице 4.

Таблица 4. Алгоритм конструктора класса class\_3

№	Предикат	Действия	№ перехода	Комментарий
1		Вызов метода put() класса class_1	2	
2		Вывод "\nThe original array: "	3	
3		Вызов метода output() класса class_2	4	
4		Вызов метода sort() класса class_2	5	
5		Вывод "\nAn ordered array: "	6	

6		Вызов метода output() класса class_2	∅	
---	--	--------------------------------------	---	--

Функция: main

Функционал: Главная функция программы

Параметры: нет

Возвращаемое значение: int, 0 (код возврата)

Алгоритм функции представлен в таблице 5.

Таблица 5. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Создание объекта prog класса class_3	∅	

## Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

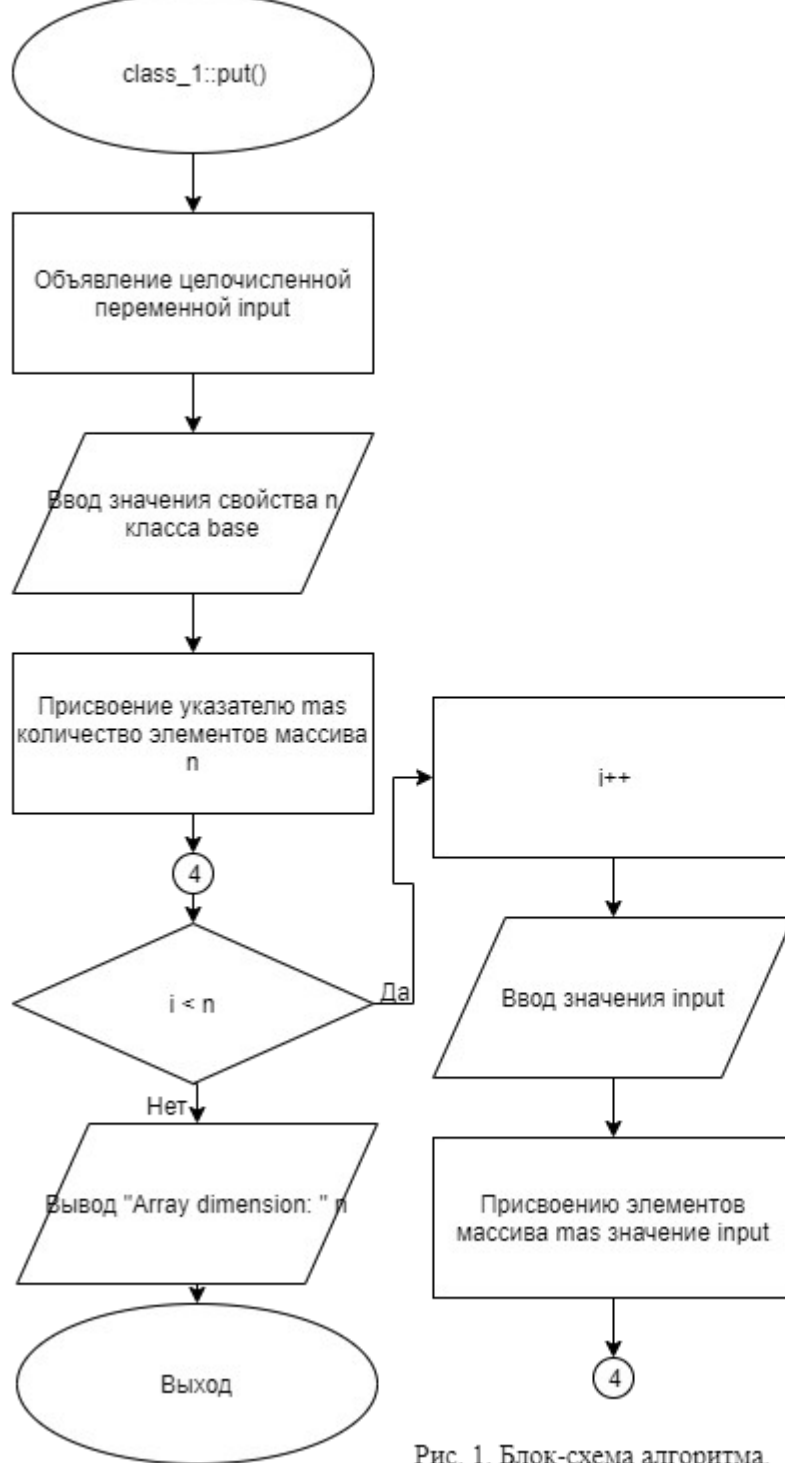


Рис. 1. Блок-схема алгоритма.

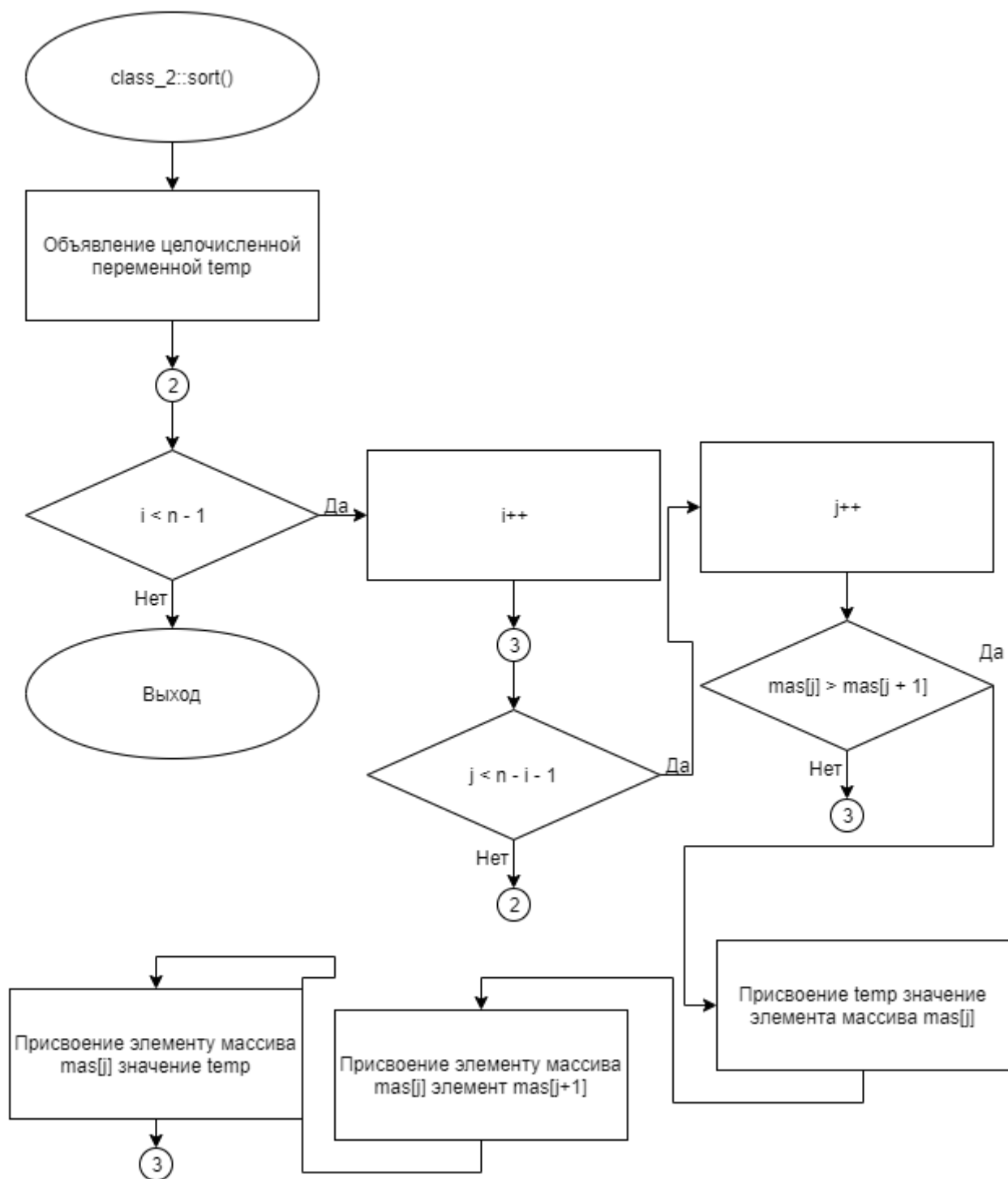


Рис. 2. Блок-схема алгоритма.

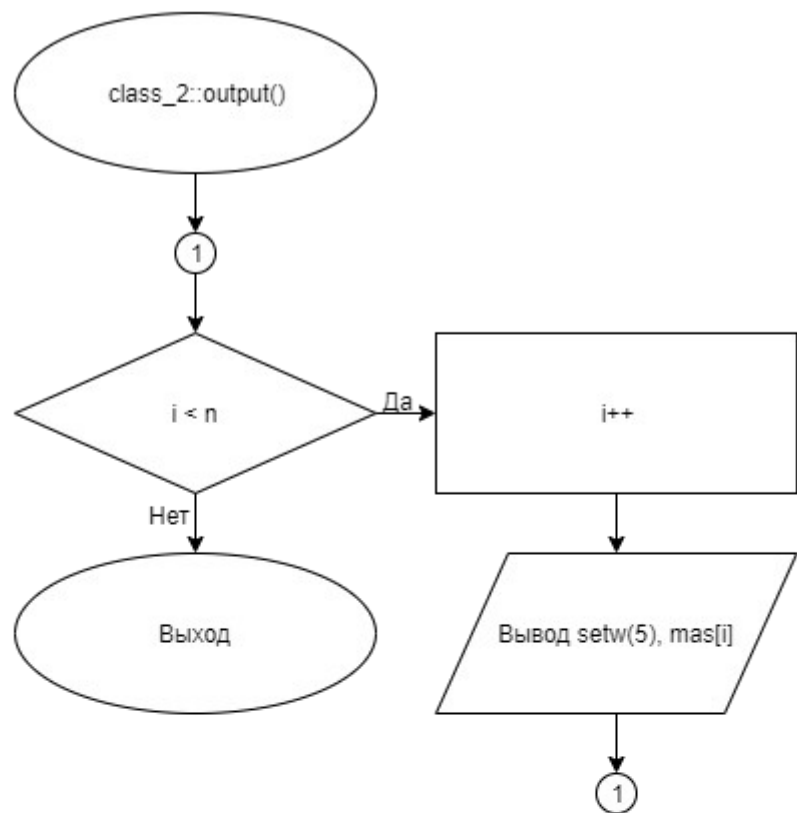


Рис. 3. Блок-схема алгоритма.

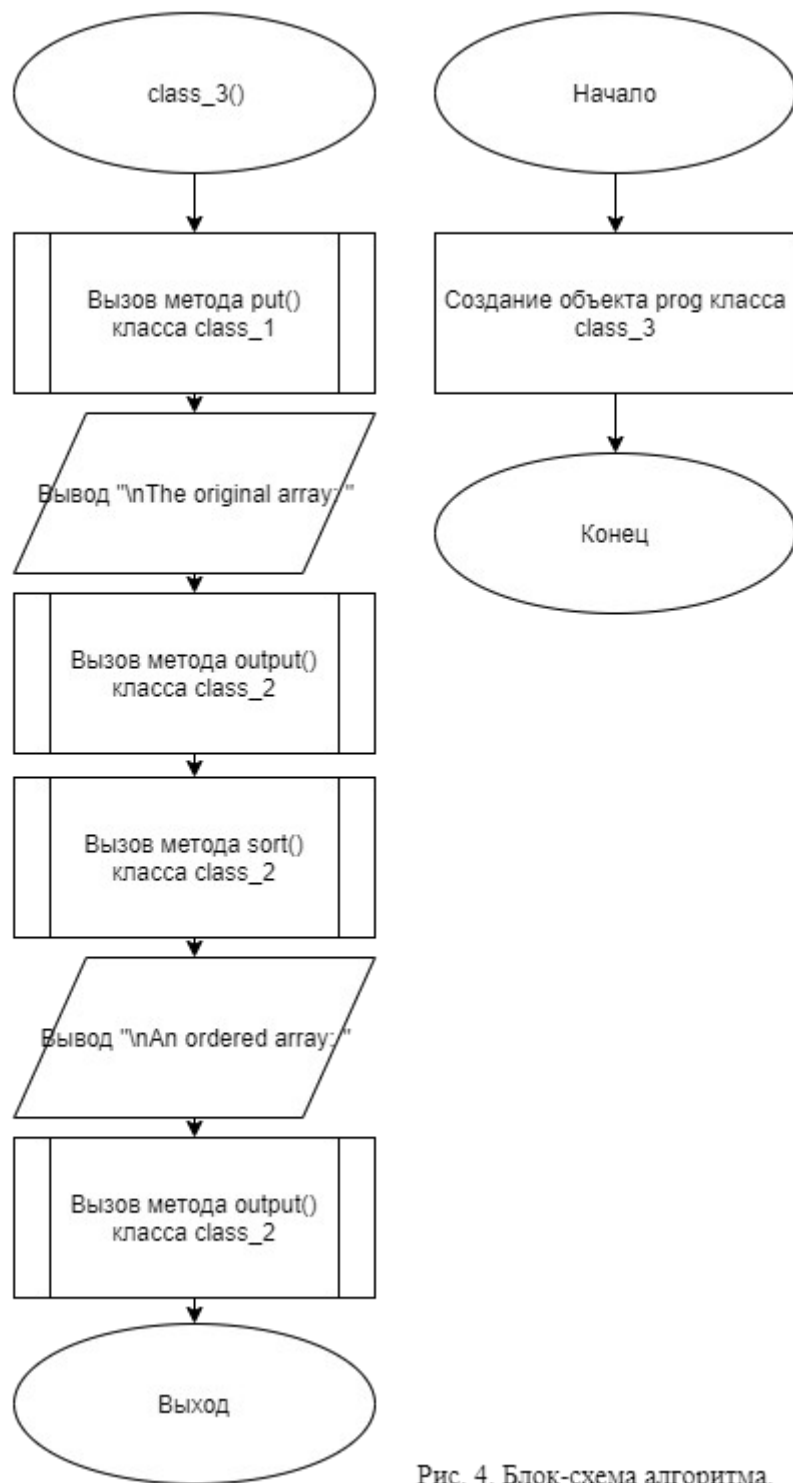


Рис. 4. Блок-схема алгоритма.

## Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

### Файл base.h

```
#ifndef BASE_H
#define BASE_H

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

class base{
protected:
    int* mas;
    int n;
};

#endif
```

### Файл class\_1.cpp

```
#include "class_1.h"

void class_1::put(){
    int input;

    cin >> n;
    mas = new int[n];

    for(int i = 0; i < n; i++){
        cin >> input;
        mas[i] = input;
    }

    cout << "Array dimension: " << n;
}
```

### Файл class\_1.h

```
#ifndef CLASS_1_H
#define CLASS_1_H

#include "base.h"
```



```

class class_1 : virtual public base{
public:
    void put();
};

#endif

```

## Файл class\_2.cpp

```

#include "class_2.h"

void class_2::sort(){ // Сортировка пузырьком
    int temp;
    for(int i = 0; i < n - 1; i++){
        for(int j = 0; j < n - i - 1; j++){
            if(mas[j] > mas[j + 1]){
                temp = mas[j];
                mas[j] = mas[j + 1];
                mas[j + 1] = temp;
            }
        }
    }
}

void class_2::output(){
    for (int i = 0; i < n; i++){
        cout << setw(5);
        cout << mas[i];
    }
}

```

## Файл class\_2.h

```

#ifndef CLASS_2_H
#define CLASS_2_H

#include "base.h"

class class_2 : virtual public base{
public:
    void sort();
    void output();
};

#endif

```

### Файл class\_3.cpp

```
#include "class_3.h"

class_3::class_3(){
    put();
    cout << "\nThe original array:";
    output();
    sort();
    cout << "\nAn ordered array:";
    output();
}
```

### Файл class\_3.h

```
#ifndef CLASS_3_H
#define CLASS_3_H

#include "class_1.h"
#include "class_2.h"

class class_3 : public class_1, public class_2{
public:
    class_3();
};

#endif
```

### Файл main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "base.h"
#include "class_1.h"
#include "class_2.h"
#include "class_3.h"

int main()
{
    // program here
    class_3 prog;
    return(0);
}
```

## Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
7 12 54 1 96 40 32 5	Array dimension: 7 The original array: 12 54 1 96 40 32 5 An ordered array: 1 5 12 32 40 54 96	Array dimension: 7 The original array: 12 54 1 96 40 32 5 An ordered array: 1 5 12 32 40 54 96
4 22 87 234 12	Array dimension: 4 The original array: 22 87 234 12 An ordered array: 12 22 87 234	Array dimension: 4 The original array: 22 87 234 12 An ordered array: 12 22 87 234
2 32 12	Array dimension: 2 The original array: 32 12 An ordered array: 12 32	Array dimension: 2 The original array: 32 12 An ordered array: 12 32

## **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](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](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).