



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_4_1 »

С тудент группы

ИКБО-27-21

Осипов М.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.....	11
Блок-схема алгоритма.....	15
Код программы.....	21
Тестирование.....	25
ЗАКЛЮЧЕНИЕ.....	26
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	27

ВВЕДЕНИЕ

Постановка задачи

Создать базовый класс, содержащий метод наименованием `in_out` без параметров.

Метод организует ввод исходных данных: размерности целочисленного массива и значений элементов целочисленного массива.

На базе базового класса создать производный класс 1 содержащий метод целого типа наименованием `func`.

Метод от значения первого элемента целочисленного массива последовательно вычитает значения остальных элементов. Полученный результат возвращает.

На базе класса 1 создать производный класс 2 содержащий метод целого типа наименованием `func`.

Метод суммирует элементы целочисленного массива. Полученный результат возвращает.

На базе класса 2 создать производный класс 3 содержащий метод наименованием `in_out` без параметров для вывода исходных данных и вычисленных значений.

В основной программе использовать один указатель на объект класса и один объект.

В основной программе реализовать алгоритм:

1. Объявить указатель на объект класса.
2. Создать объект оператором `new`.
3. Ввод исходных данных: размерности целочисленного массива и значений элементов данного массива.
4. Вычисление значение вычитания от первого элемента массива всех

остальных.

5. Вычисление суммы элементов массива.

6. Вывод полученных результатов.

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:

Min: «значение вычитания из первого элемента массива всех остальных»

Четвертая строка:

Sum: «значение суммы элементов массив»

Метод решения

Основная программа:

Указатель на объект obj класса class_3 с динамической памятью

Оператор new

Вызов метода in_out() класса base

Вызов метода func() класса class_1

Вызов метода func() класса class_2

Вызов метода in_out() класса class_3

Класс base:

Модификатор доступа protected

Целочисленная переменная n

Целочисленный указатель mas

Модификатор доступа public

Метод in_out() с типом данных void

Целочисленный тип данных

Объект ввода/вывода - cin/cout потока данных (iostream)

Динамический массив

Цикл for

Класс class_1:

Наследование класса base

Модификатор доступа public

Метод func() с целочисленным типом данных

Цикл for

Целочисленный тип данных

Класс class_2:

наследование класса class_1

Модификатор доступа public

Метод func() с целочисленным типом данных

Цикл for

Целочисленный тип данных

Класс class_3:

Наследование класса class_2

Модификатор доступа public

Метод in_out() с типом данных void

Объект ввода/вывода - cin/cout потока данных (iostream)

Цикл for

Манипулятор setw

Описание алгоритма

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

Класс объекта: base

Модификатор доступа: public

Метод: in_out

Функционал: Ввод размера и элементов массива

Параметры: нет

Возвращаемое значение: void, ничего

Алгоритм метода представлен в таблице 1.

Таблица 1. Алгоритм метода in_out класса base

№	Предикат	Действия	№ перехода	Комментарий
1		Ввод значения n	2	
2		динамическому массиву mas задаём размер переменной n	3	
3	i < n	Инкремент i	4	
			Ø	
4		Ввод значения input	5	
5		Элементу массива mas[i] задаём значения input	3	

Класс объекта: class_1

Модификатор доступа: public

Метод: func

Функционал: Вычитание элементов массива от первого элемента

Параметры: нет

Возвращаемое значение: int, Значение вычитаний элементов массива от первого элемента

Алгоритм метода представлен в таблице 2.

Таблица 2. Алгоритм метода func класса class_1

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация переменной number значением mas[0]	2	
2	i < n	Инкремент i	3	
			4	
3		Присваиваем переменной number значение (number - mas[i])	2	
4		Возвращение значения number	Ø	

Класс объекта: class_2

Модификатор доступа: public

Метод: func

Функционал: Сумма всех элементов массива

Параметры: нет

Возвращаемое значение: int, Значение суммы всех элементов массива

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода func класса class_2

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация переменной summ значением 0	2	
2	i < n	Инкремент i	3	
			4	
3		Переменной summ присваиваем значение (summ + mas[i])	2	
4		Возвращение значения summ	∅	

Класс объекта: class_3

Модификатор доступа: public

Метод: in_out

Функционал: Выводит размерность, значение, разность и сумму элементов массива

Параметры: нет

Возвращаемое значение: void, ничего

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода in_out класса class_3

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод "Array dimension", n, вывод на новую строку	2	
2		Вывод "The original array:"	3	
3	i < n	Инкремент i	4	
			5	
4		Вывод манипулятора setw(5), значение элемента массива mas[i]	3	
5		Вывод перехода на новую	6	

		строку		
6		Вывод "Min: " значение метода func класса class_1, переход на новую строку	7	
7		Вывод "Sum: " значение метода func класса class_2	Ø	

Функция: main

Функционал: Главная функция программы

Параметры: нет

Возвращаемое значение: int, Признак успеха программы

Алгоритм функции представлен в таблице 5.

Таблица 5. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Указатель obj класса class_3 с динамической памятью	2	
2		Вызов метода in_out класса base объекта obj	3	
3		Вызов метода func класса class_1 объекта obj	4	
4		Вызов метода func класса class_2 объекта obj	5	
5		Вызов метода in_out класса class_3 объекта obj	Ø	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

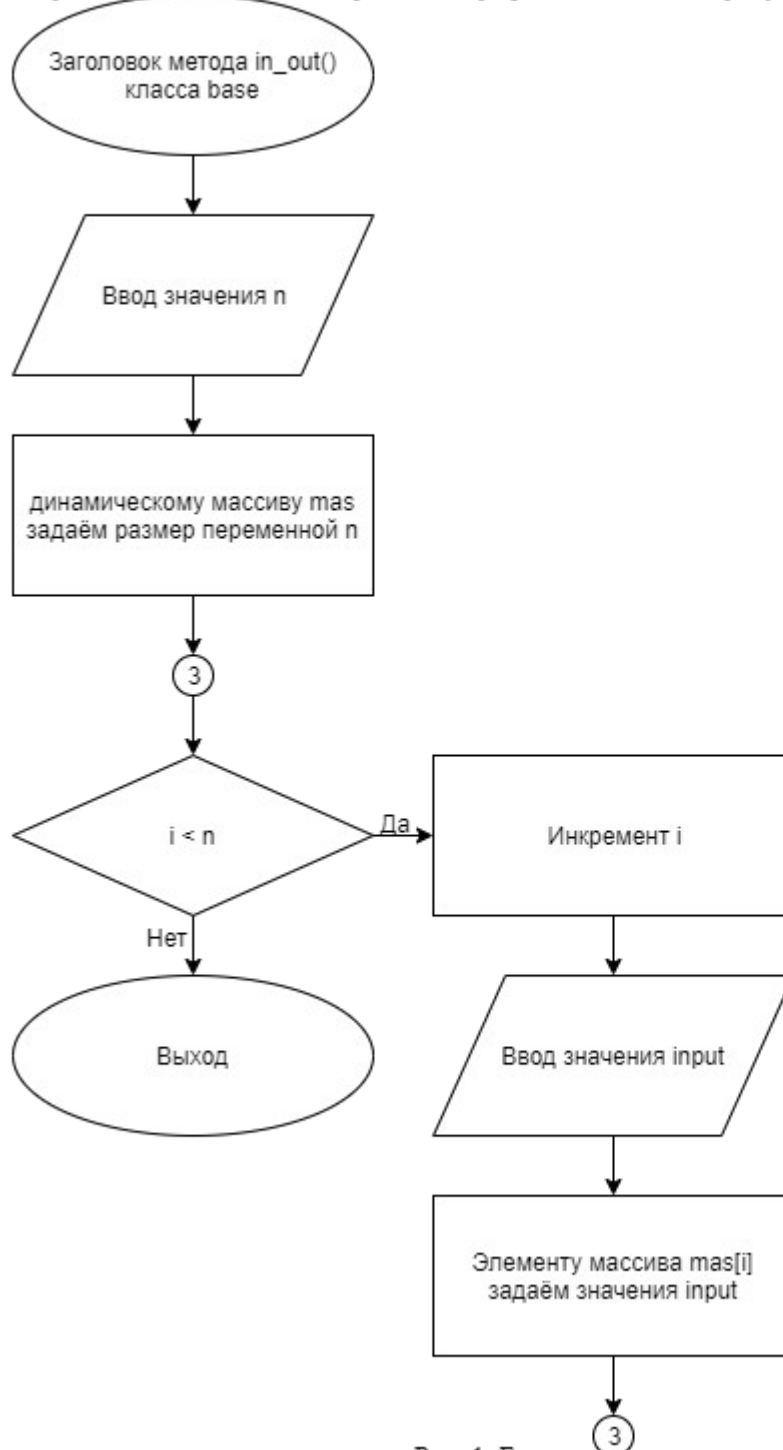


Рис. 1. Блок-схема алгоритма.

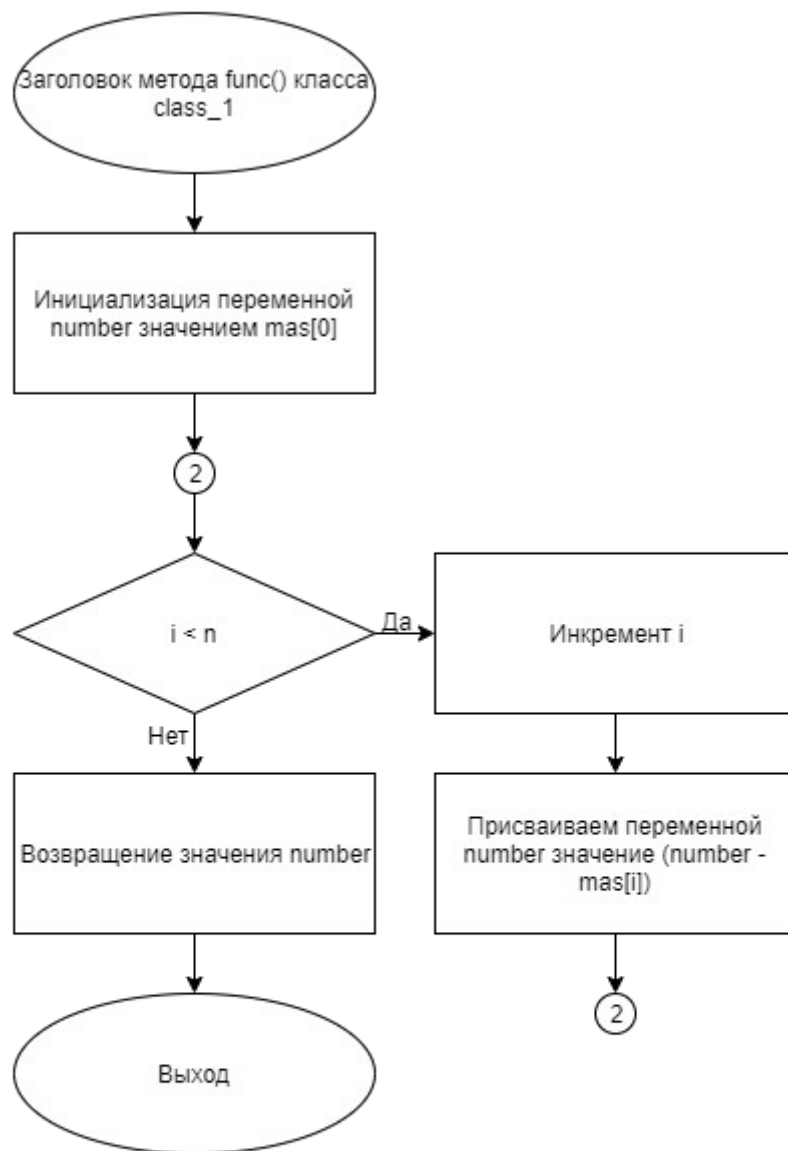


Рис. 2. Блок-схема алгоритма.

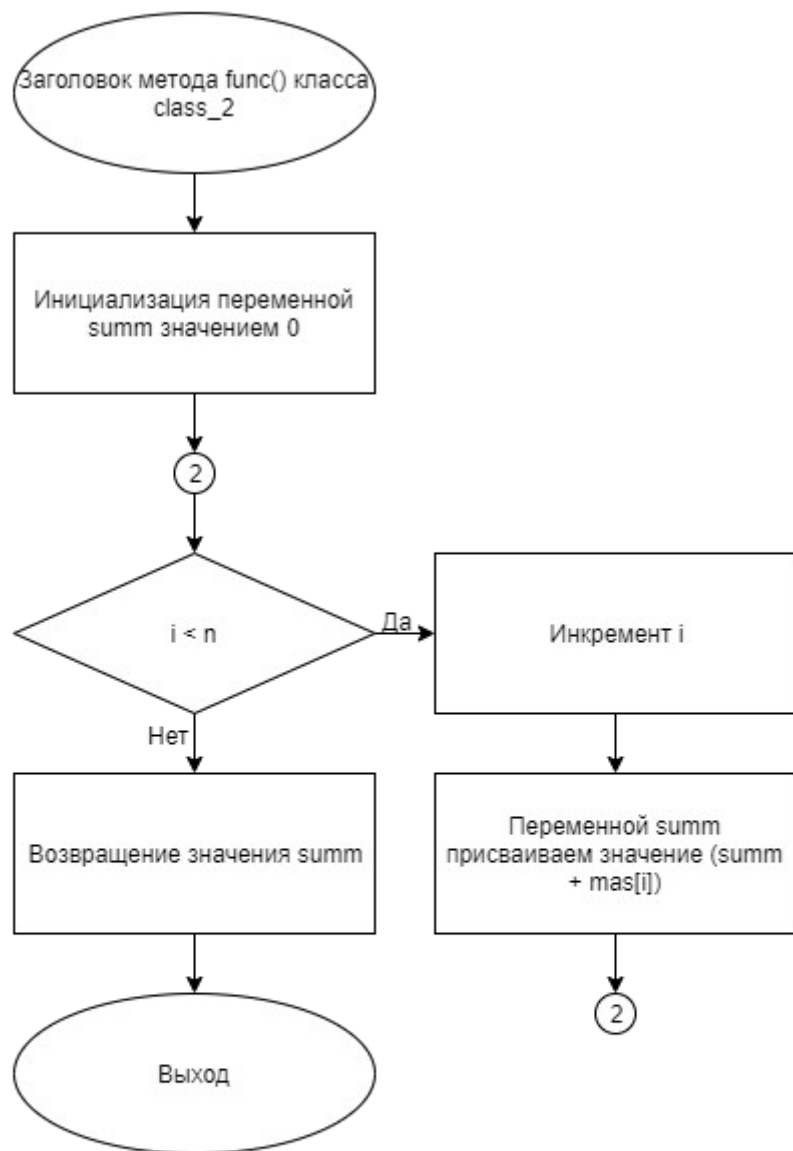


Рис. 3. Блок-схема алгоритма.

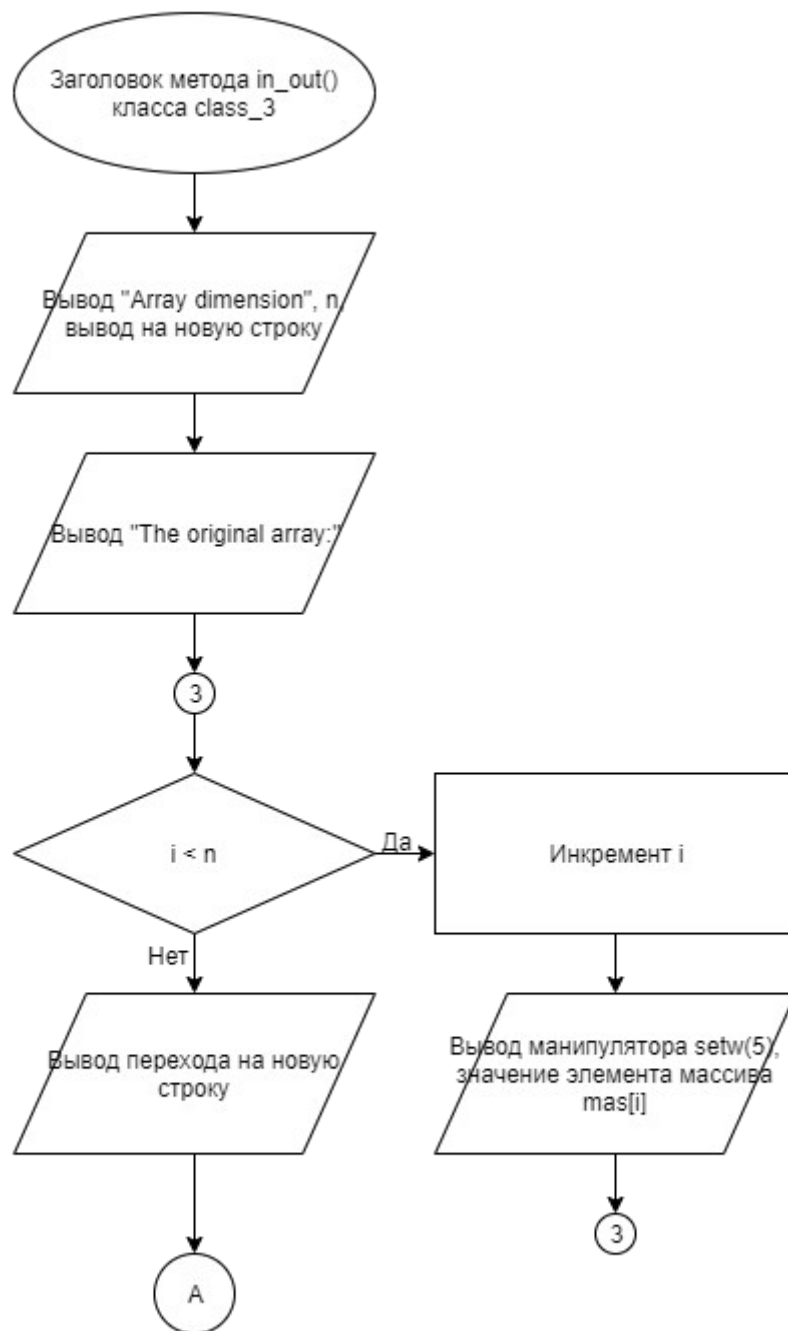


Рис. 4. Блок-схема алгоритма.

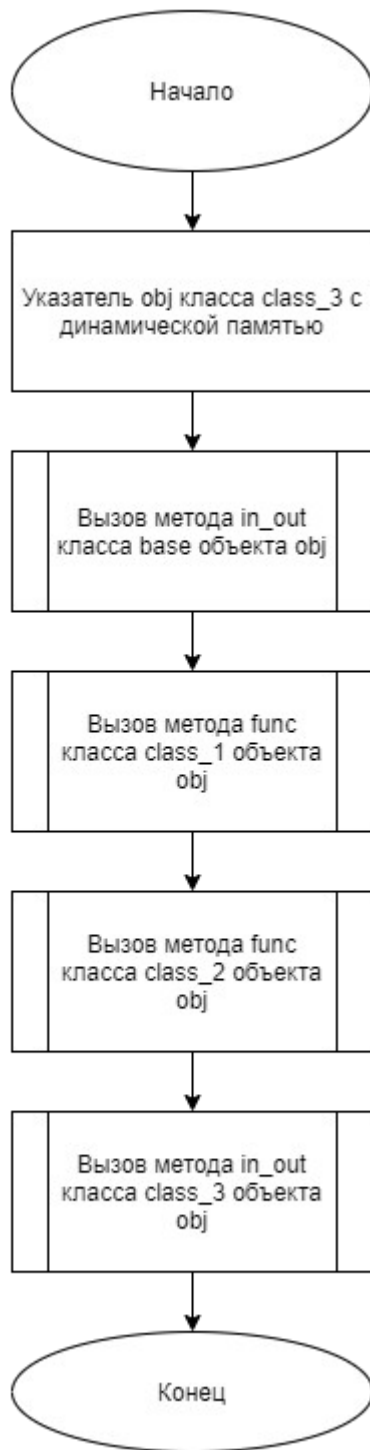


Рис. 5. Блок-схема алгоритма.



Рис. 6. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл base.cpp

```
#include "base.h"

void base::in_out()
{
    int input;
    cin >> n;
    mas = new int[n];

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

Файл base.h

```
#ifndef BASE_H
#define BASE_H
#include <iostream>
#include <iomanip>

using namespace std;

class base
{
protected:
    int n;
    int* mas;
public:
    void in_out();
};
#endif
```

Файл class_1.cpp

```
#include "class_1.h"

int class_1::func()
```

```

{
int number = mas[0];
for(int i = 1; i < n; i++)
{
    number -= mas[i];
}
return number;
}

```

Файл class_1.h

```

#ifndef CLASS_1_H
#define CLASS_1_H
#include "base.h"

class class_1 : public base
{
public:
int func();
};
#endif

```

Файл class_2.cpp

```

#include "class_2.h"

int class_2::func()
{
int summ = 0;
for(int i = 0; i < n; i++)
{
    summ += mas[i];
}
return summ;
}

```

Файл class_2.h

```

#ifndef CLASS_2_H
#define CLASS_2_H
#include "class_1.h"

class class_2 : public class_1

```

```

{
public:
int func();
};
#endif

```

Файл class_3.cpp

```

#include "class_3.h"

void class_3::in_out()
{
cout << "Array dimension: " << n << endl;
cout << "The original array:";
    for(int i = 0; i < n; i++)
    {
        cout << setw(5) << mas[i];
    }
    cout << endl;
    cout << "Min: " << class_1::func() << endl;
    cout << "Sum: " << class_2::func();
}

```

Файл class_3.h

```

#ifndef CLASS_3_H
#define CLASS_3_H
#include "class_2.h"

class class_3 : public class_2
{
public:
void in_out();
};
#endif

```

Файл main.cpp

```

#include <stdlib.h>
#include <stdio.h>
#include "class_3.h"

int main()
{
    class_3* obj = new class_3;
    obj->base::in_out();
    obj->class_1::func();
}

```

```
    obj->class_2::func();  
    obj->class_3::in_out();  
    return(0);  
}
```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 22 12	Array dimension: 2 The original array: 22 12 Min: 10 Sum: 34	Array dimension: 2 The original array: 22 12 Min: 10 Sum: 34
3 3 2 1	Array dimension: 3 The original array: 3 2 1 Min: 0 Sum: 6	Array dimension: 3 The original array: 3 2 1 Min: 0 Sum: 6
4 1 2 3 4	Array dimension: 4 The original array: 1 2 3 4 Min: -8 Sum: 10	Array dimension: 4 The original array: 1 2 3 4 Min: -8 Sum: 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).