

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 метода print класса triangle.....	9
3.3 Алгоритм функции operator+.....	9
3.4 Алгоритм функции operator-.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	15
5.1 Файл main.cpp.....	15
5.2 Файл triangle.cpp.....	16
5.3 Файл triangle.h.....	17
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

1 ПОСТАНОВКА ЗАДАЧИ

Перегрузка арифметических операций.

Перезагрузка операции для объекта треугольник.

У треугольника есть стороны a , b , c и они принимают только натуральные значения. Определяем операцию сложения и вычитания для треугольников.

+ сложить значения сторон, если допустимо.

- вычесть значения сторон, если допустимо.

Складываются и вычитаются соответствующие стороны треугольников. Т.е. $a_1 + a_2$, $b_1 + b_2$, $c_1 + c_2$. Если после выполнения операции получается недопустимый треугольник, то результатом операции берется первый аргумент.

Написать программу, которая выполняет операции над треугольниками.

В основной программе реализовать алгоритм:

1. Ввод количества треугольников n .
2. В цикле для каждого треугольника вводятся исходные длины сторон. Далее создается объект, в конструктор которого передаются значения длин сторон. Каждый объект треугольника получает свой номер от 1 до n .
3. В цикле, последовательно, построчно вводится «номер первого треугольника» «символ арифметической операции $+$ или $-$ » «номер второго треугольника»
4. После каждого ввода выполняется операция, результат присваивается первому аргументу (объекту треугольника).
5. Цикл завершается по завершению данных.
6. Выводится результат последней операции.

Гарантируется:

- Количество треугольников больше или равно 2;

- Значения исходных длин сторон треугольников задаются корректно.

Реализовать перегрузку арифметических операции «+» и «-» для объектов треугольника посредством самостоятельных не дружественных функций.

1.1 Описание входных данных

Первая строка содержит значение количества треугольников n :

«Натуральное значение»

Далее n строк содержат

«Натуральное значение» «Натуральное значение» «Натуральное значение»

Начиная с $n + 2$ строки:

«Натуральное значение» «Знак операции» «Натуральное значение»

1.2 Описание выходных данных

a = «Натуральное значение»; b = «Натуральное значение»; c = «Натуральное значение».

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект `triangle` класса `triangles` предназначен для ;
- функция `operator+` для функция для перегрузки оператора сложения;
- функция `operator-` для функция для перегрузки оператора вычитания;
- `for` - оператор цикла со счетчиком;
- `while` - оператор цикла с предусловием;
- `if...else` - условный оператор.

Класс `triangle`:

- свойства/поля:
 - поле первая сторона:
 - наименование — `a`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле вторая сторона:
 - наименование — `b`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле третья сторона:
 - наименование — `c`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `print` — вывод значений треугольника.

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: запуск программы.

Параметры: none.

Возвращаемое значение: код ошибки - int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменных n, a, b, c типа int	2
2		ввод значения переменной n	3
3		созданиее пустого массива triangles типа triangle	4
4	i<n		5
			8
5		ввод значений переменных a, b, c	6
6		создание объекта obj класса triangle с параметрами a, b, c	7
7		добавление объекта obj в массив triangles в 4 позиции i, i++	4
8		объявление переменных triangle1, triangle2 типа int	9
9		объявление переменной operation типа char	10
10	(cin >> triangle1, cin >> operation, cin >> triangle2)		13
			11

№	Предикат	Действия	№ перехода
11	(operation=='+')	массиву triangles в позиции (triangles1-1) присваивается сумма значений массива triangles в позиции (triangle1-1) и массива triangles в позиции (triangle2-1)	12
			12
12	(operation=='-')	массиву triangles в позиции (triangles1-1) присваивается разность значений массива triangles в позиции (triangle1-1) и массива triangles в позиции (triangle2-1)	10
			10
13		вызов метода print у объекта obj из массива triangles в позиции (triangle1-1)	14
14		возврат значения 0	∅

3.2 Алгоритм метода print класса triangle

Функционал: вывод значений треугольника.

Параметры: none.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода print класса triangle

№	Предикат	Действия	№ перехода
1		вывод на экран сообщение "a = Натуральное значение; b = Натуральное значение; c = Натуральное значение."	∅

3.3 Алгоритм функции operator+

Функционал: оператор сложения.

Параметры: triangle t1, triangle t2.

Возвращаемое значение: triangle.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции operator+

№	Предикат	Действия	№ перехода
1		переменной a объекта t1 присваивается сумма значений свойств a объекта t1 и a объекта t2	2
2		переменной b объекта t1 присваивается сумма значений свойств b объекта t1 и b объекта t2	3
3		переменной c объекта t1 присваивается сумма значений свойств c объекта t1 и c объекта t2	4
4	$(a+b>c)\&(a+c>b)\&(b+c>a)$	свойству a объекта t1 присваивается значение переменной a свойству b объекта t1 присваивается значение переменной b свойству c объекта t1 присваивается значение переменной c	5
			5
5		возврат значения объекта t1	∅

3.4 Алгоритм функции operator-

Функционал: оператор вычитания.

Параметры: triangle t1, triangle t2.

Возвращаемое значение: triangle.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции operator-

№	Предикат	Действия	№ перехода
1		переменной a объекта t1 присваивается разность значений свойств a объекта t1 и a объекта t2	2
2		переменной b объекта t1 присваивается разность значений свойств a объекта t1 и b объекта t2	3
3		переменной b объекта t1 присваивается разность значений свойств b объекта t1 и a объекта t2	4
4	$(a+b>c)\&(a+c>b)\&(b+c>a))$	свойству a объекта t1 присваивается значение переменной a	5
		свойству b объекта t1 присваивается значение переменной b	
		свойству c объекта t1 присваивается значение переменной c	
			5
5		возврат значения объекта t1	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

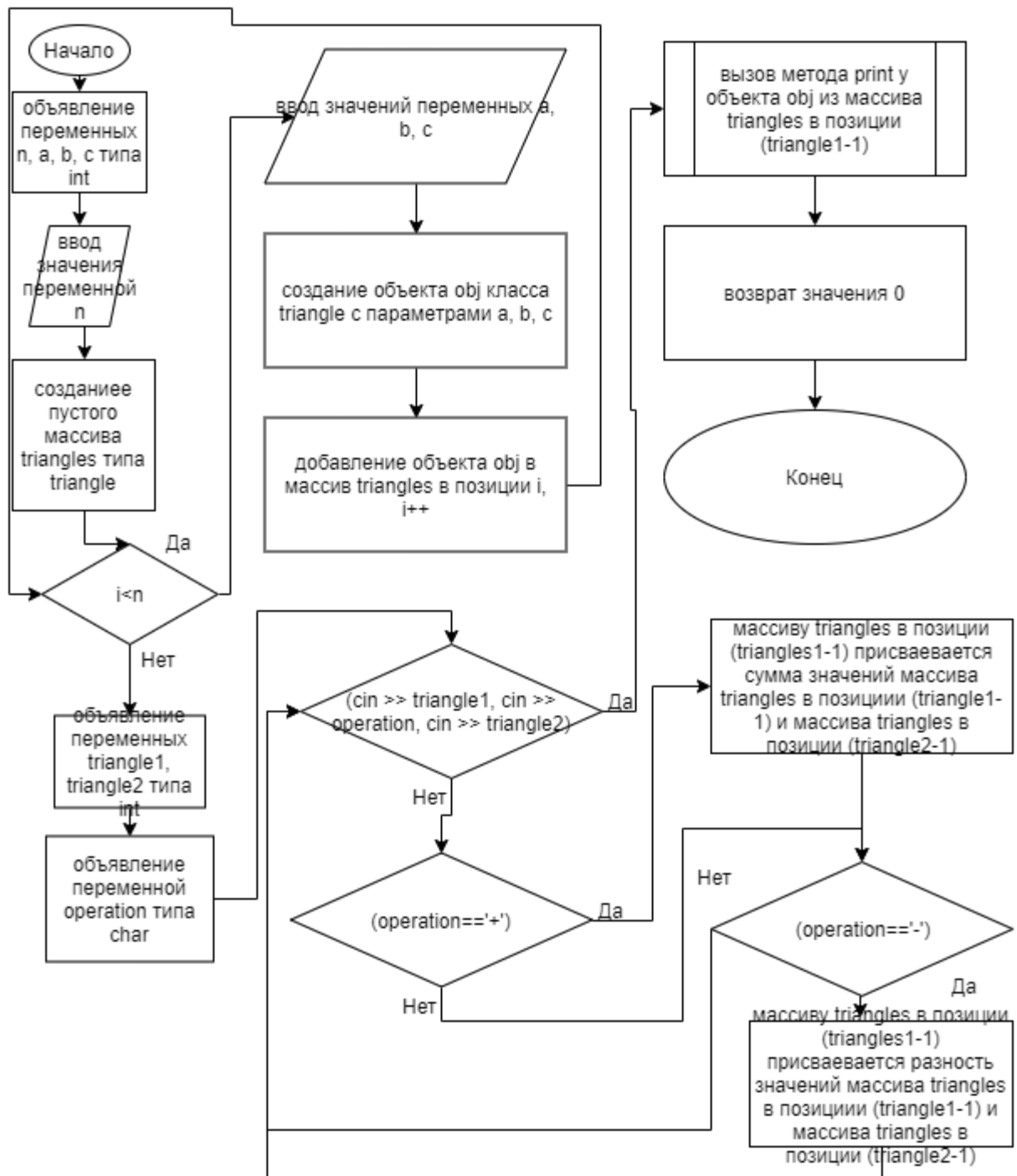


Рисунок 1 – Блок-схема алгоритма

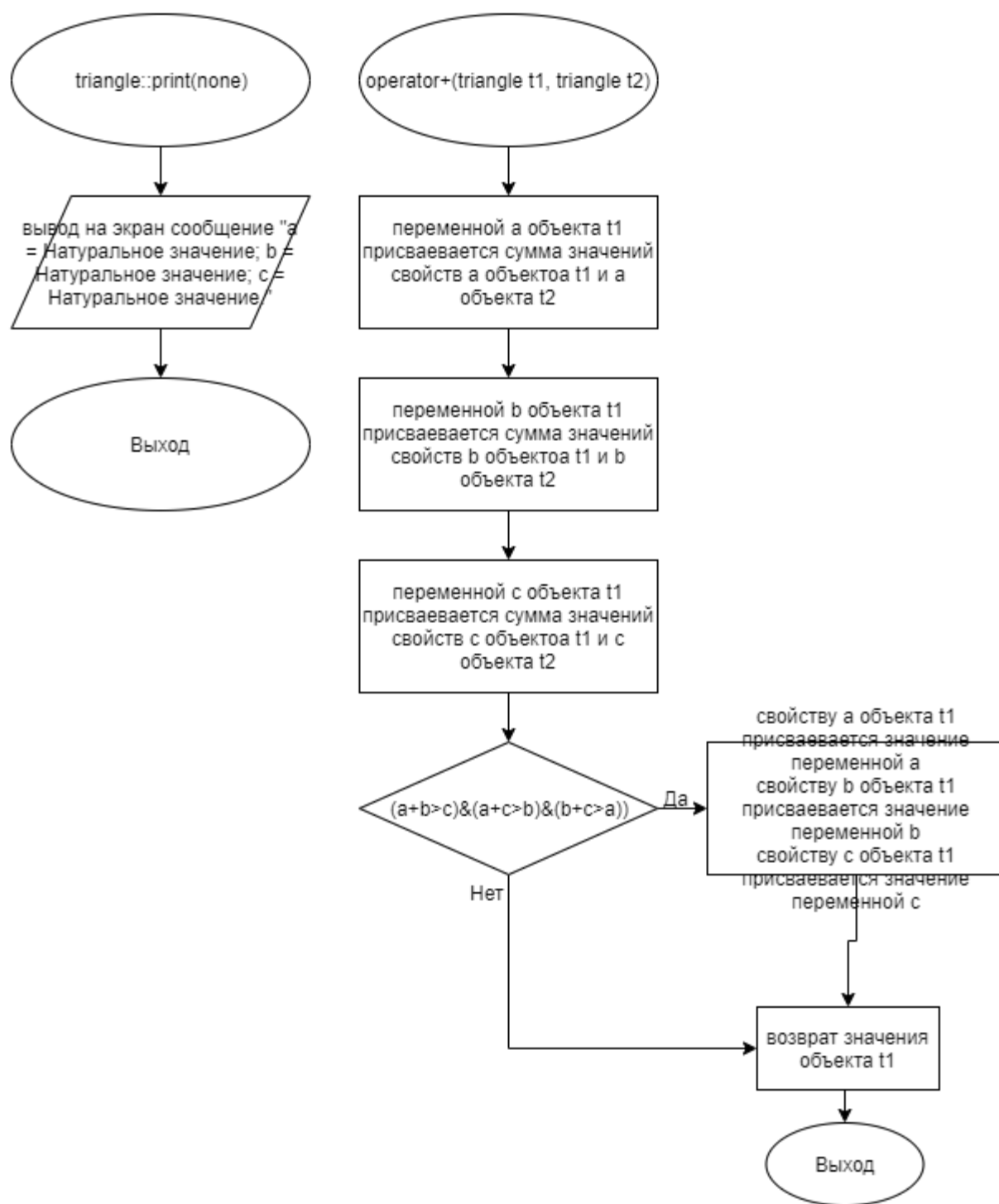


Рисунок 2 – Блок-схема алгоритма

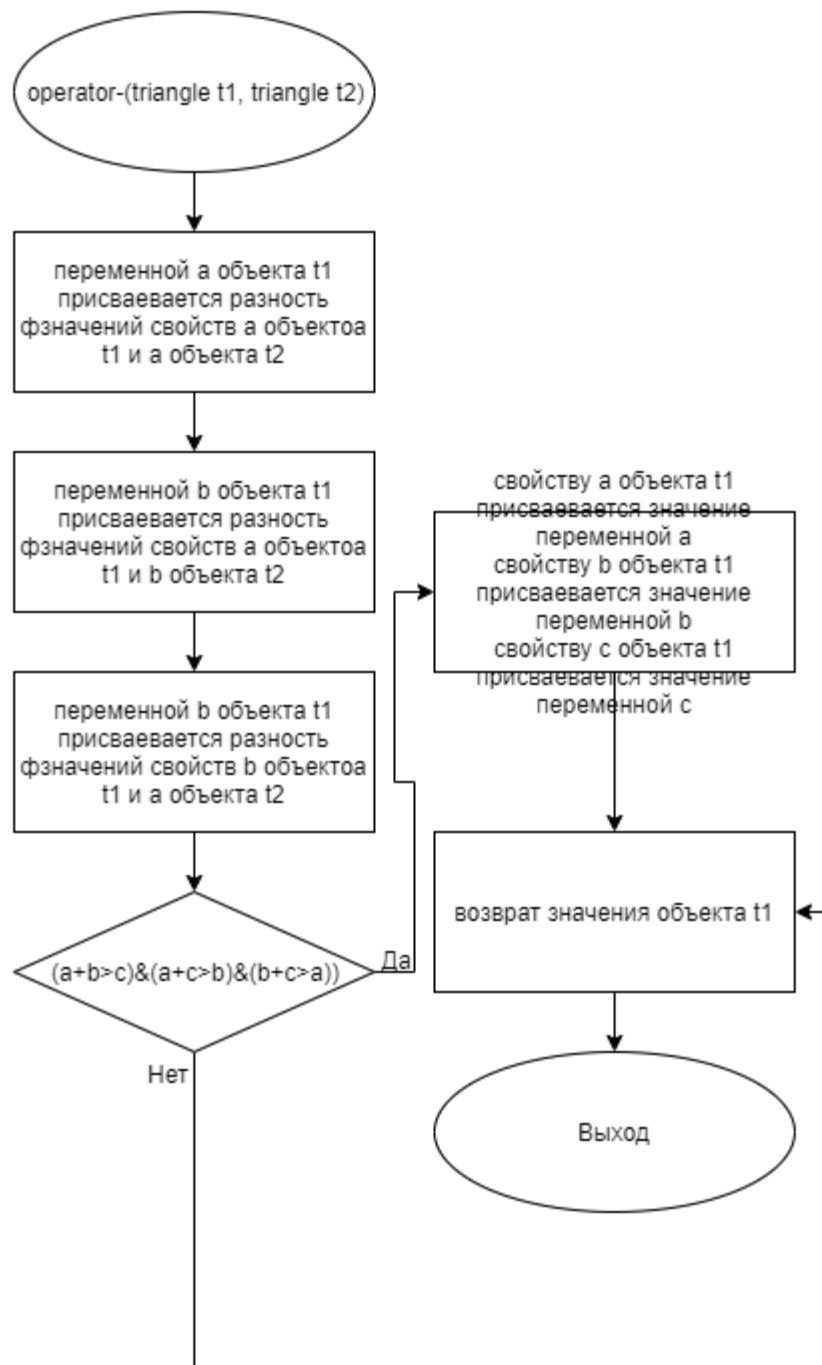


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <vector>
#include "triangle.h"

using namespace std;

triangle operator +(triangle t1, triangle t2)
{
    int a = t1.a+t2.a;
    int b = t1.b+t2.b;
    int c = t1.c+t2.c;
    if ((a+b>c)&(a+c>b)&(b+c>a))
    {
        t1.a=a;
        t1.b=b;
        t1.c=c;
    }
    return t1;
}

triangle operator -(triangle t1, triangle t2)
{
    int a = t1.a-t2.a;
    int b = t1.b-t2.b;
    int c = t1.c-t2.c;
    if ((a+b>c)&(a+c>b)&(b+c>a))
    {
        t1.a=a;
        t1.b=b;
        t1.c=c;
    }
    return t1;
}

int main()
{
    int n,a,b,c;
```

```

        cin>>n;
        vector <triangle> triangles;
        for(int i = 0; i<n; i++){
            cin >> a >> b >> c;
            triangle obj(a,b,c);
            triangles.push_back(obj);
        }
        int triangle1, triangle2;
        char operation;
        while(cin >> triangle1, cin >> operation, cin >> triangle2){
            if(operation == '+'){
                triangles[triangle1-1] = triangles[triangle1-1] +
triangles[triangle2-1];
            }
            else if (operation == '-'){
                triangles[triangle1-1] = triangles[triangle1-1] -
triangles[triangle2-1];
            }
        }
        triangles[triangle1-1].print();
        return 0;
    }

```

5.2 Файл triangle.cpp

Листинг 2 – triangle.cpp

```

#include "triangle.h"
#include <iostream>
#include <cmath>

using namespace std;

triangle::triangle(int a, int b, int c)
{
    this->a=a;
    this->b=b;
    this->c=c;
}

void triangle :: print(){
    cout << "a = " << a << "; " << "b = " << b << "; " << "c = " << c << "."
<< endl;
}

```

5.3 Файл triangle.h

Листинг 3 – triangle.h

```
#ifndef __TRIANGLE__H
#define __TRIANGLE__H

class triangle
{
    public:
        int a;
        int b;
        int c;
        triangle(int a, int b, int c);
        void print();
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 3 4 5 5 12 13 1 + 2	a = 8; b = 16; c = 18.	a = 8; b = 16; c = 18.

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 (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 (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).