

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	5
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.1 Алгоритм функции main.....	8
3.2 Алгоритм метода метод met1 класса calc.....	8
3.3 Алгоритм метода метод met2 класса calc.....	9
3.4 Алгоритм метода метод met3 класса calc.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	15
5.1 Файл calc.cpp.....	15
5.2 Файл calc.h.....	15
5.3 Файл main.cpp.....	16
6 ТЕСТИРОВАНИЕ.....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18

# 1 ПОСТАНОВКА ЗАДАЧИ

Сконструировать систему, которая моделирует работу простого арифметического калькулятора.

Спроектировать объект, который вычисляет значение целочисленного арифметического выражения, состоящего из трех последовательных операций. Операция деления заменена на операцию вычисления целочисленного остатка.

Объект обладает закрытым свойством и следующей функциональностью:

- выполняет первую операцию выражения, в качестве параметров передается первый целочисленный параметр, символ операции (+, -, \*, %), второй целочисленный параметр;
- вычисляет вторую и далее операцию, в качестве параметров передается символ операции (+, -, \*, %), второй целочисленный параметр;
- возвращает значение вычисленного выражения.

Алгоритм отработки системы:

1. Создает объект.
2. Вводит значения аргументов для первой операции.
3. Выполняет первую операцию.
4. Вводит значение аргументов для второй операции.
5. Выполняет вторую операцию.
6. Вводит значение аргументов для третьей операции.
7. Выполняет третью операцию.
8. Выводит результат.

## 1.1 Описание входных данных

Первая строка:

«целое число в десятичном формате»\_«символ операции»\_«целое число в десятичном формате»

Вторая строка:

«символ операции»\_«целое число в десятичном формате»

Третья строка:

«символ операции»\_«целое число в десятичном формате»

## **1.2 Описание выходных данных**

Первая строка, с первой позиции:

«значение выражения»

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект с класса `calc` предназначен для проверки работы простого арифметического калькулятора;
- библиотека `iostream`;
- библиотека `string`;
- объект стандартного потока ввода `cin`;
- объект стандартного потока вывода `cout`;
- операторы выбора `switch-case`.

Класс `calc`:

- свойства/поля:
  - поле скрытое свойство объекта:
    - наименование — `res`;
    - тип — целочисленный;
    - модификатор доступа — `private`;
- функционал:
  - метод `метод met1` — вычисляет первую операцию выражения;
  - метод `метод met2` — вычисляет вторую и далее операции;
  - метод `метод met3` — возвращает значение вычисленного выражения.

## 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: Главная функция программы.

Параметры: нет.

Возвращаемое значение: Целое число, идентификатор работоспособности программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции `main`

№	Предикат	Действия	№ перехода
1		объявление целочисленных переменных <code>x1</code> , <code>x2</code> , <code>x3</code> , <code>x4</code>	2
2		объявление символьных переменных <code>щз1б</code> <code>щз2б</code> <code>щз3</code>	3
3		создание объекта с класса <code>calc</code>	4
4		ввод значений переменных <code>x1</code> , <code>op1</code> , <code>x2</code>	5
5		вызов метода <code>met1</code> объекта с параметрами <code>x1</code> , <code>op1</code> , <code>x2</code>	6
6		ввод значений <code>op2</code> , <code>x3</code>	7
7		вызов метода <code>met2</code> объекта с параметрами <code>op2</code> , <code>x3</code>	8
8		ввод значений <code>op3</code> , <code>x4</code>	9
9		вызов метода <code>met2</code> объекта с параметрами <code>op3</code> , <code>x4</code>	10
10		вывод вызванного метода <code>met3</code> объекта с	Ø

### 3.2 Алгоритм метода `метод met1` класса `calc`

Функционал: вычисляет первую операцию выражения.

Параметры: первый целочисленный параметр, символ операции (+, -, \*, %).

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода метод *met1* класса *calc*

№	Предикат	Действия	№ перехода
1		switch op1	2
2	op1 == '+'?	присвоение значения выражения x1+x2 скрытому свойству res класса calc	6
			3
3	op1 == '-'?	присвоение значения выражения x1-x2 скрытому свойству res класса calc	6
			4
4	op1 == '*'?	присвоение значения выражения x1*x2 скрытому свойству res класса calc	6
			5
5	op1 == '%'?	присвоение значения выражения x1%x2 скрытому свойству res класса calc	6
			6
6		прерывание выполнения оператора выбора	Ø

### 3.3 Алгоритм метода метод *met2* класса *calc*

Функционал: вычисляет вторую и далее операции.

Параметры: символ операции (+, -, \*, %), второй целочисленный параметр.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода метод *met2* класса *calc*

№	Предикат	Действия	№ перехода
1		switch op2	2
2	op2 == "?"	прибавить скрытому свойству res класса calc значение переменной x3	6
			3
3	op2 == "?"	вычесть из скрытого свойства res класса calc значение переменной x3	6
			4
4	op2 == "?"	умножить скрытое свойство res класса calc значение переменной x3	6
			5
5	op2 == "?"	присвоить значение скрытому свойству res класса calc остатка от деления на значение переменной x3	6
			6
6		прерывание выполнения оператора выбора	∅

### 3.4 Алгоритм метода метод *met3* класса *calc*

Функционал: возвращает значение вычисленного выражения.

Параметры: нет.

Возвращаемое значение: целое, значение вычисленного выражения.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода метод *met3* класса *calc*

№	Предикат	Действия	№ перехода
1		возвращение значения вычисленного выражения	∅



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

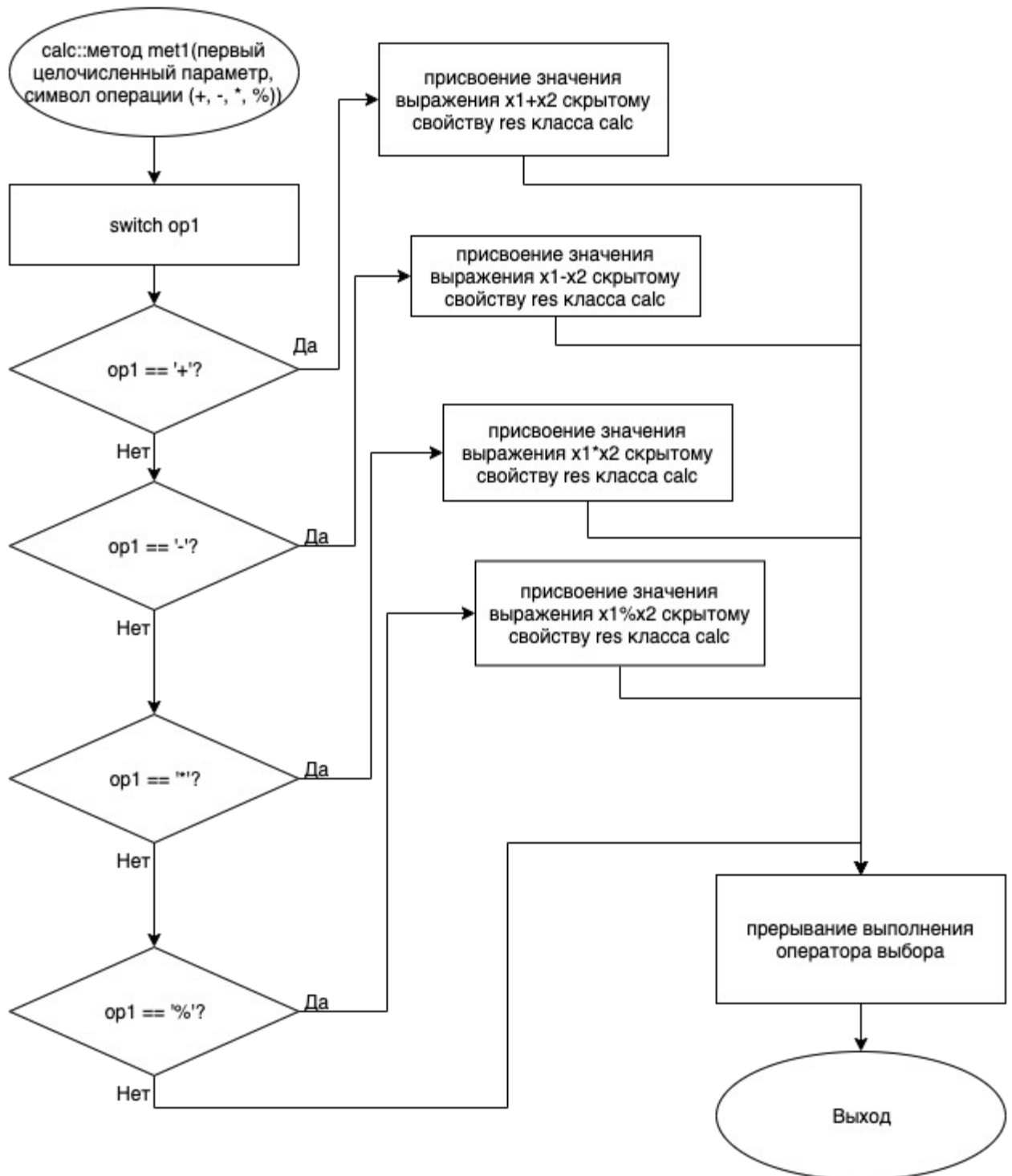


Рисунок 1 – Блок-схема алгоритма



**Рисунок 2 – Блок-схема алгоритма**

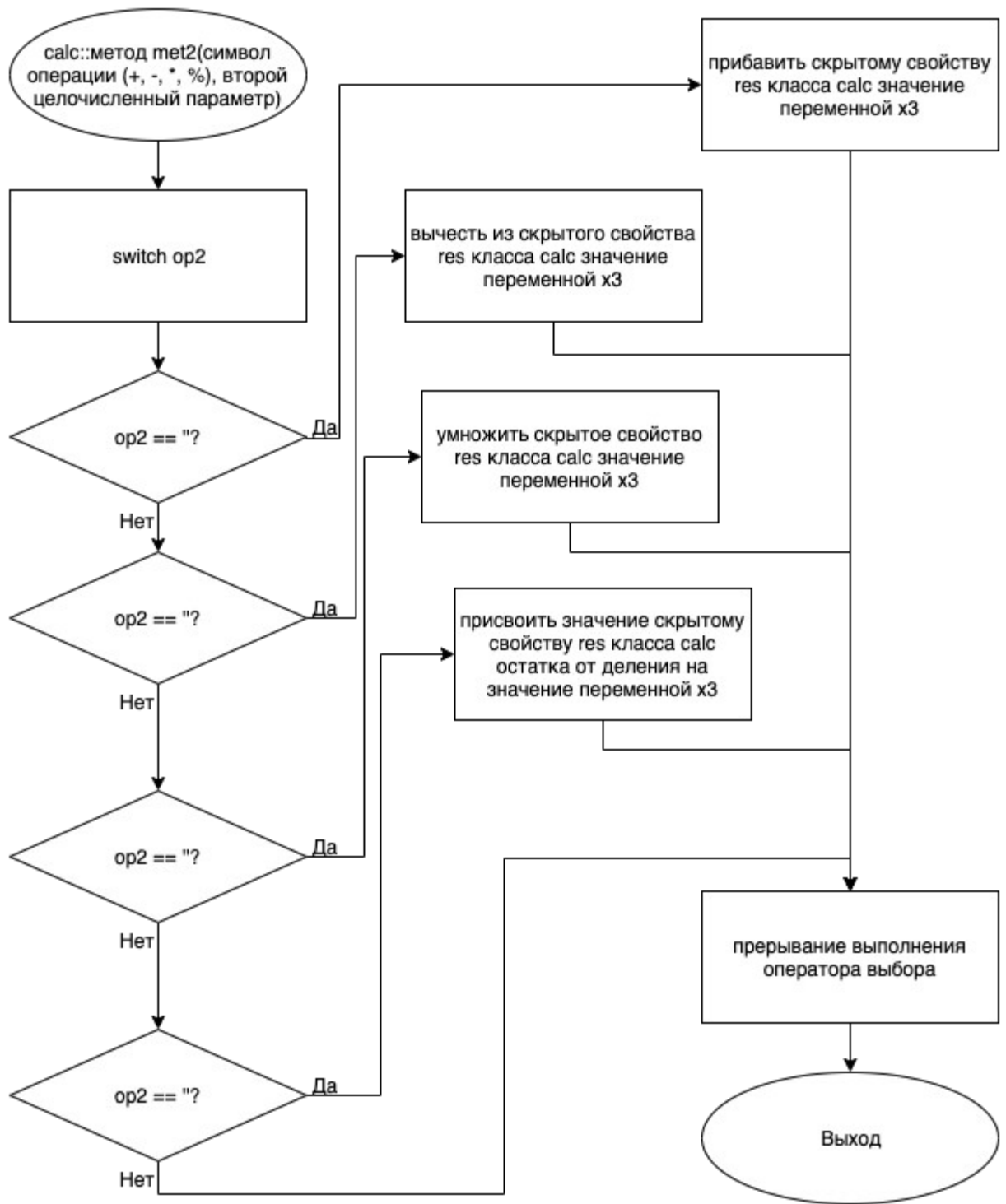


Рисунок 3 – Блок-схема алгоритма

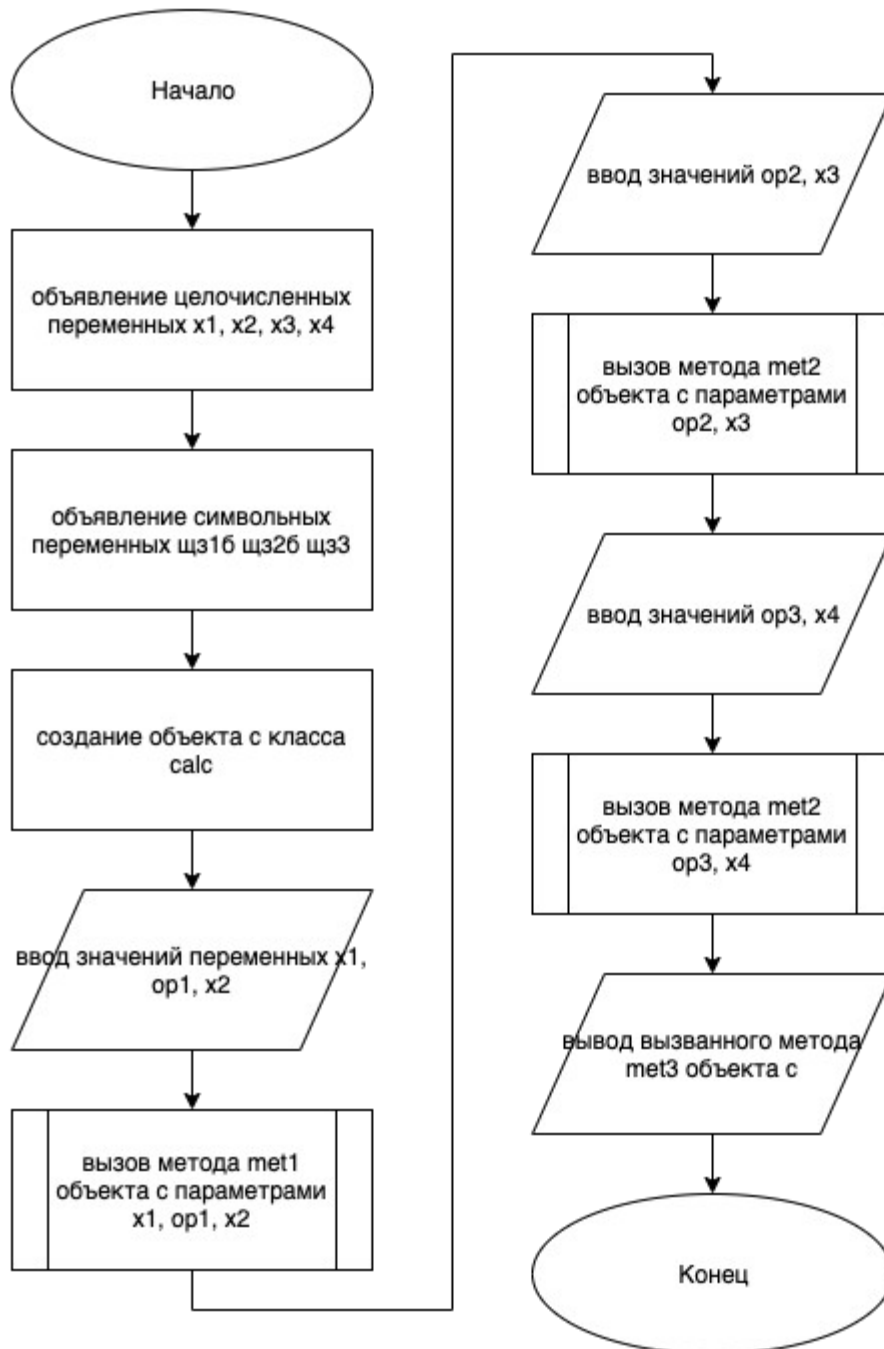


Рисунок 4 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл calc.cpp

*Листинг 1 – calc.cpp*

```
#include "calc.h"
using namespace std;

void calc::met1(int x1, char op1, int x2)
{
    switch(op1){
        case '+': res = x1 + x2; break;
        case '-': res = x1 - x2; break;
        case '*': res = x1 * x2; break;
        case '%': res = x1 % x2; break;
    }
};

void calc::met2(char op2, int x3)
{
    switch(op2){
        case '+': res += x3; break;
        case '-': res -= x3; break;
        case '*': res *= x3; break;
        case '%': res %= x3; break;
    }
};

int calc::met3()
{
    return res;
};
```

### 5.2 Файл calc.h

*Листинг 2 – calc.h*

```
#ifndef __CALC__H
#define __CALC__H
```

```

class calc{
private:
    int res;
public:
    void met1(int x1, char op1, int x2);
    void met2(char op2, int x3);
    int met3();
};

#endif

```

### 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```

#include <iostream>
#include "calc.h"
using namespace std;

int main()
{
    int x1, x2, x3, x4;
    char op1, op2, op3;
    calc c;

    cin >> x1 >> op1 >> x2;
    c.met1(x1, op1, x2);

    cin >> op2 >> x3;
    c.met2(op2, x3);

    cin >> op3 >> x4;
    c.met2(op3, x4);

    cout << c.met3();

    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 + 3 * 4 - 2	30	30
0 % 2 * 5 + 3	3	3
-200 * 48 % 27 + 4	-11	-11

## 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).