

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 calculation класса Arifmetik.....	8
3.3 Алгоритм метода calculate класса Arifmetik.....	9
3.4 Алгоритм метода get_result класса Arifmetik.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	13
5.1 Файл Arifmetik.cpp.....	13
5.2 Файл Arifmetik.h.....	14
5.3 Файл main.cpp.....	14
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

# 1 ПОСТАНОВКА ЗАДАЧИ

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

Объект обладает 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 (значение можно получить после выполнения трех операций).

Написать программу, которая:

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 obj класса Arifmetik;
- функция main для Основная функция;
- Объект стандартного потока ввода/вывода cin/ cout;
- Оператор множественного выбора.

Класс Arifmetik:

- свойства/поля:
  - поле Результат выполнения операций:
    - наименование — result;
    - тип — int;
    - модификатор доступа — private;
- функционал:
  - метод calculation — вычисление первой операции;
  - метод calculate — вычисление второй и последующих операций;
  - метод get\_result — Возвращение результата подсчёта.

## 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: Основная функция.

Параметры: нет.

Возвращаемое значение: `int`.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции `main`

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных <code>x</code> и <code>y</code>	2
2		Объявление символьной переменной <code>symbol</code>	3
3		Создание объекта <code>obj</code> класса <code>Arifmetik</code>	4
4		Ввод значений переменных <code>x, symbol, y</code>	5
5		Выполнение метода <code>calculation()</code> с параметрами <code>x, symbol, y</code>	6
6		Ввод значений переменных <code>symbol, y</code>	7
7		Выполнение метода <code>calculate()</code> с параметрами <code>symbol, y</code>	8
8		Ввод значений переменных <code>symbol, y</code>	9
9		Выполнение метода <code>calculate()</code> с параметрами <code>symbol, y</code>	10
10		Вывод результата методом <code>get_result()</code>	Ø

### 3.2 Алгоритм метода `calculation` класса `Arifmetik`

Функционал: Выполняет первую операцию между двумя целочисленными переменными.

Параметры: Целочисленные x,y и символьный symbol.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода calculation класса Arifmetik

№	Предикат	Действия	№ перехода
1	Знак операции "+"	Подсчёт $x + y$ и присваивание это значения переменной result	Ø
			2
2	Знак операции "-"	Подсчёт $x - y$ и присваивание это значения переменной result	Ø
			3
3	Знак операции "*" "	Подсчёт $x * y$ и присваивание это значения переменной result	Ø
			4
4	Знак операции "%"	Подсчёт $x \% y$ и присваивание это значения переменной result	Ø
			Ø

### 3.3 Алгоритм метода calculate класса Arifmetik

Функционал: Последующие выполнения операций над целыми числами.

Параметры: целочисленный y и символьный symbol.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода calculate класса Arifmetik

№	Предикат	Действия	№ перехода
1	Знак операции "+"	Прибавление значение переменной result к переменной y	Ø

№	Предикат	Действия	№ перехода
			2
2	Знак операции "-"	Вычитание значения переменной y от переменной result	∅
			3
3	Знак операции "*" "	Умножение значения переменной result на значение y	∅
			4
4	Знак операции "%"	Целочисленное деление result на y	∅
			∅

### 3.4 Алгоритм метода get\_result класса Arifmetik

Функционал: Возвращение результата подсчёта.

Параметры: нет.

Возвращаемое значение: Целое.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода get\_result класса Arifmetik

№	Предикат	Действия	№ перехода
1		Возвращает значение result	∅



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

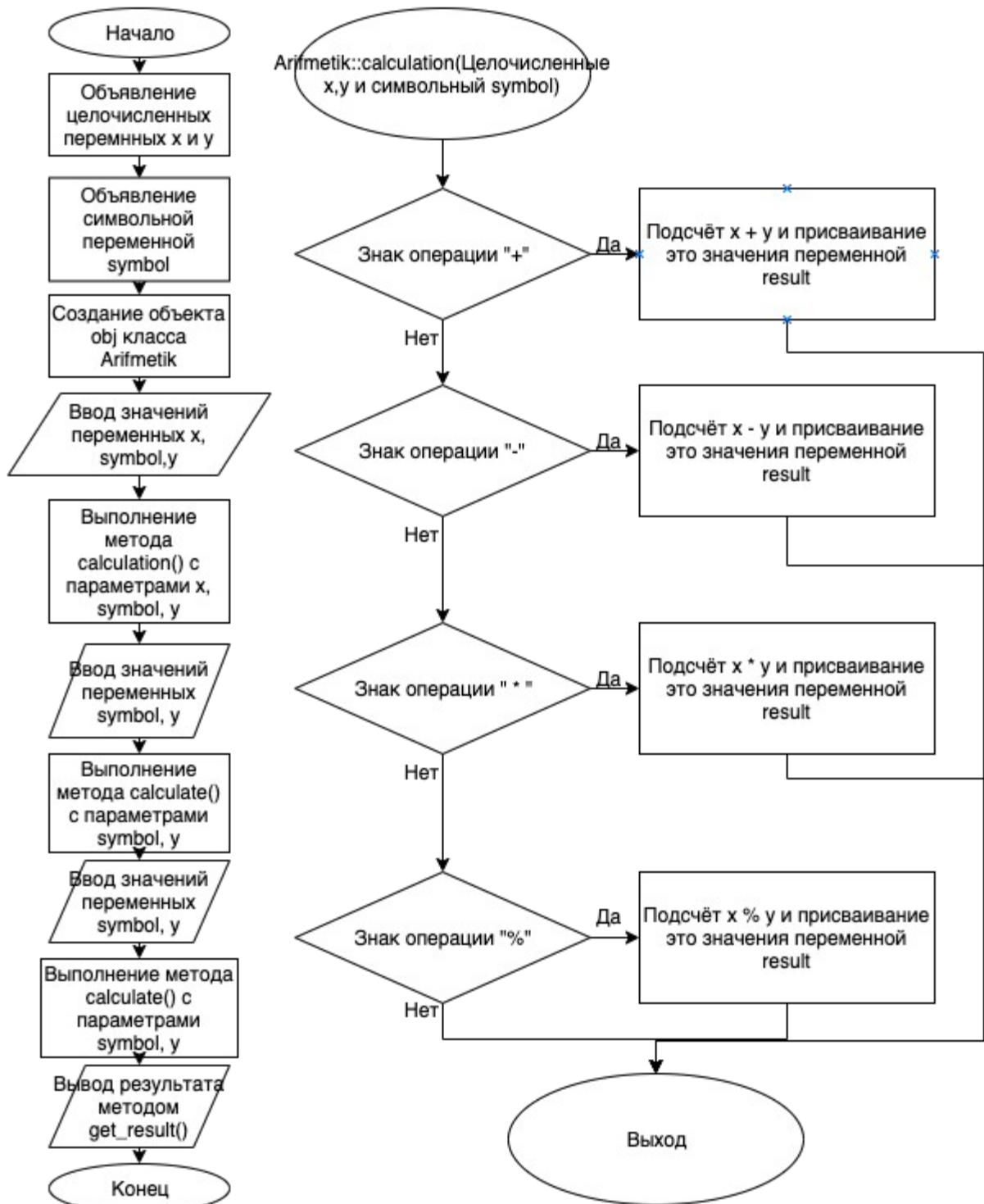


Рисунок 1 – Блок-схема алгоритма

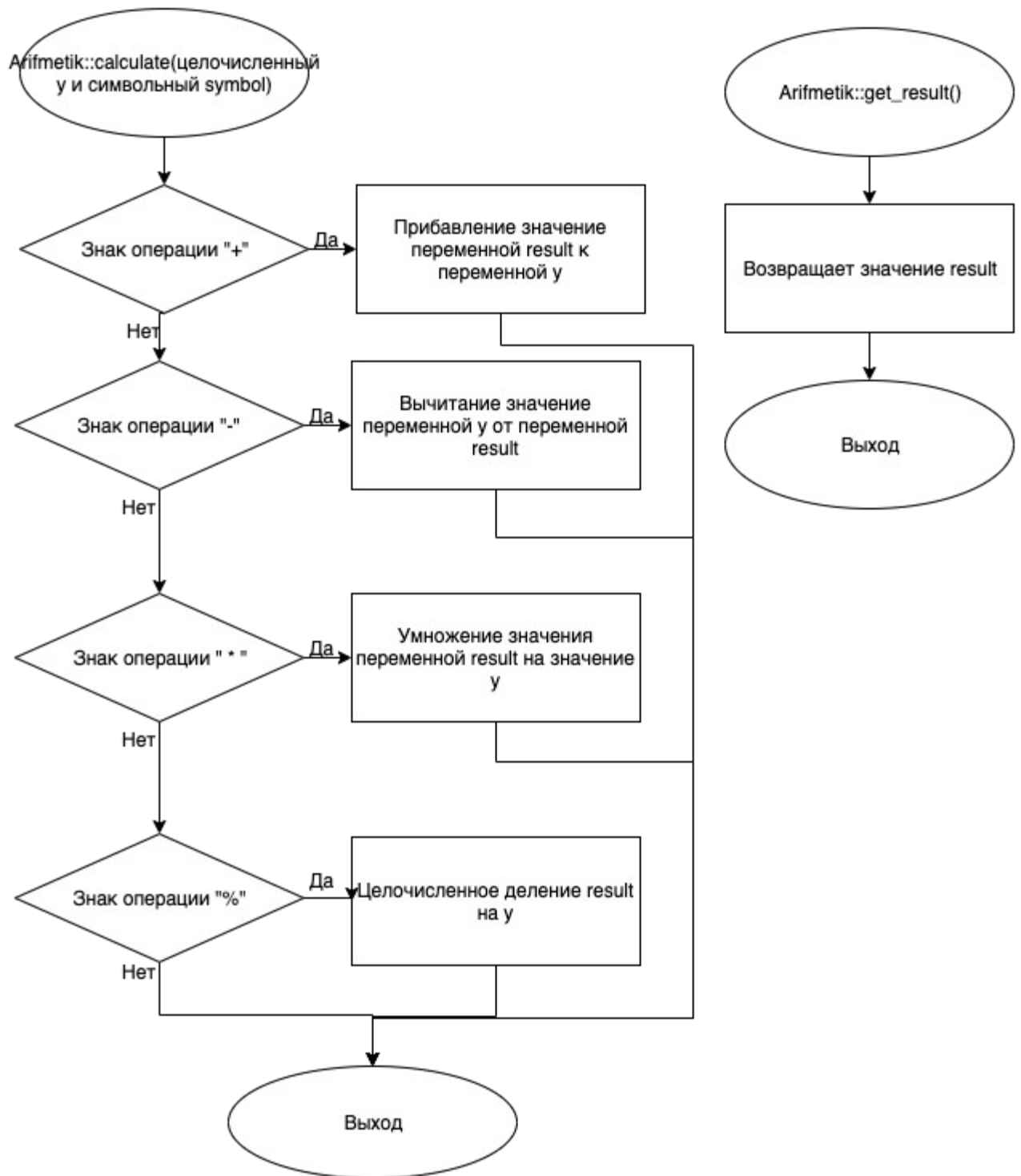


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл Arifmetik.cpp

*Листинг 1 – Arifmetik.cpp*

```
#include "Arifmetik.h"
void Arifmetik::calculation(int x, char symbol, int y)
{
    switch(symbol)
    {
        case('+'):
            result = x + y;
            break;
        case('-'):
            result = x - y;
            break;
        case('*'):
            result = x * y;
            break;
        case('%'):
            result = x % y;
            break;
    }
}
void Arifmetik::calculate(char symbol, int y)
{
    switch(symbol)
    {
        case('+'):
            result += y;
            break;
        case('-'):
            result -= y;
            break;
        case('*'):
            result *= y;
            break;
        case('%'):
            result %= y;
            break;
    }
}
int Arifmetik::get_result()
{

```

```
    return result;  
}
```

## 5.2 Файл Arifmetik.h

*Листинг 2 – Arifmetik.h*

```
#ifndef __ARIFMETIK__H  
#define __ARIFMETIK__H  
#include <iostream>  
using namespace std;  
  
class Arifmetik  
{  
private:  
    int result;  
public:  
    void calculation(int x, char sign, int y);  
    void calculate(char sign, int y);  
    int get_result();  
};  
#endif
```

## 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```
#include <stdlib.h>  
#include <stdio.h>  
#include "Arifmetik.h"  
int main()  
{  
    int x,y;  
    char symbol;  
    Arifmetik obj;  
    cin >> x >> symbol >> y;  
    obj.calculation(x,symbol,y);  
  
    cin >> symbol >> y;  
    obj.calculate(symbol,y);  
  
    cin >> symbol >> y;  
    obj.calculate(symbol,y);  
  
    cout <<obj.get_result();  
}
```

```
    return(0);  
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

*Таблица 5 – Результат тестирования программы*

<b>Входные данные</b>	<b>Ожидаемые выходные данные</b>	<b>Фактические выходные данные</b>
1+2+3+4	10	10
10%2+5-2	3	3
14*2-18-1	9	9

## 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).