

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 метода calculation класса Arifmetik.....	8
3.2 Алгоритм метода calculate класса Arifmetik.....	9
3.3 Алгоритм метода get_result класса Arifmetik.....	9
3.4 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	15
5.1 Файл Arifmetik.cpp.....	15
5.2 Файл Arifmetik.h.....	16
5.3 Файл main.cpp.....	16
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

# 1 ПОСТАНОВКА ЗАДАЧИ

Создать объект, который вычисляет значение целочисленного арифметического выражения.

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.

Написать программу, которая обязательно вводит значения и выполняет первую операцию.

Далее, в цикле осуществляет ввод очередной операции и значения второго аргумента.

Если на месте операции введен символ «С», то программа завершает работу, иначе выполняет очередную операцию и выводит результат каждой третьей операции.

## 1.1 Описание входных данных

**Первая строка:**

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

**Последующие строки:**

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

**В последней строке:**

С

## **1.2 Описание выходных данных**

**Первая строка**, с первой позиции:

«значение выражения»

**Последующие строки**, с первой позиции:

«значение выражения»

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj класса Arifmetik предназначен для ;
- функция main для Основная функция;
- Объект стандартного ввода/вывода cin/cout;
- Оператор множественного выбора;
- Цикл с предусловием;
- Условный оператор.

Класс Arifmetik:

- свойства/поля:
  - поле Полученный результат:
    - наименование — result;
    - тип — int;
    - модификатор доступа — private;
- функционал:
  - метод calculation — Вычисление первой операции между целочисленными переменными;
  - метод calculate — Вычисление последующих операций между целочисленными переменными;
  - метод get\_result — Вывод полученного результата после выполнения всех операций.

## 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 метода calculation класса Arifmetik

Функционал: Вычисление первой операции между целочисленными переменными.

Параметры: нет.

Возвращаемое значение: Нет.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода calculation класса Arifmetik

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

### 3.2 Алгоритм метода calculate класса Arifmetik

Функционал: Вычисление последующих операций между целочисленными переменными.

Параметры: нет.

Возвращаемое значение: Нет.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода calculate класса Arifmetik

№	Предикат	Действия	№ перехода
1	Знак операции "+"	Прибавления значения result к переменной y	Ø
			2
2	Знак операции "-"	Вычитание значение y из переменной result	Ø
			3
3	Знак операции "*"	Умножение result на y	Ø
			4
4	Знак операции "%"	Целочисленное деление переменной result на y	Ø
			Ø

### 3.3 Алгоритм метода get\_result класса Arifmetik

Функционал: Вывод полученного результата после выполнения всех операций.

Параметры: нет.

Возвращаемое значение: Целое.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода get\_result класса Arifmetik

№	Предикат	Действия	№ перехода
1		Возвращает значение целочисленной переменной result	Ø

### 3.4 Алгоритм функции main

Функционал: Основная функция.

Параметры: Отсутствуют.

Возвращаемое значение: Целое.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных x, y	2
2		Объявление символьной переменной symbol	3
3		Объявление и инициализация целочисленной переменной count = 1	4
4		Создание объекта obj класса Arifmetik	5
5		Ввод значения x, symbol, y	6
6		Выполнение метода calculation(int x, char symbol, int y)	7
7	True	Инкрементация переменной count	8
		Возвращение значения result	∅
8		Ввод значения переменной symbol, y	9
9		Выполнение метода calculate	10
10			11
11	symbol = 'C'	Завершение цикла	∅
			12
12	count % 3 == 0	Вывод result	∅
			7



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

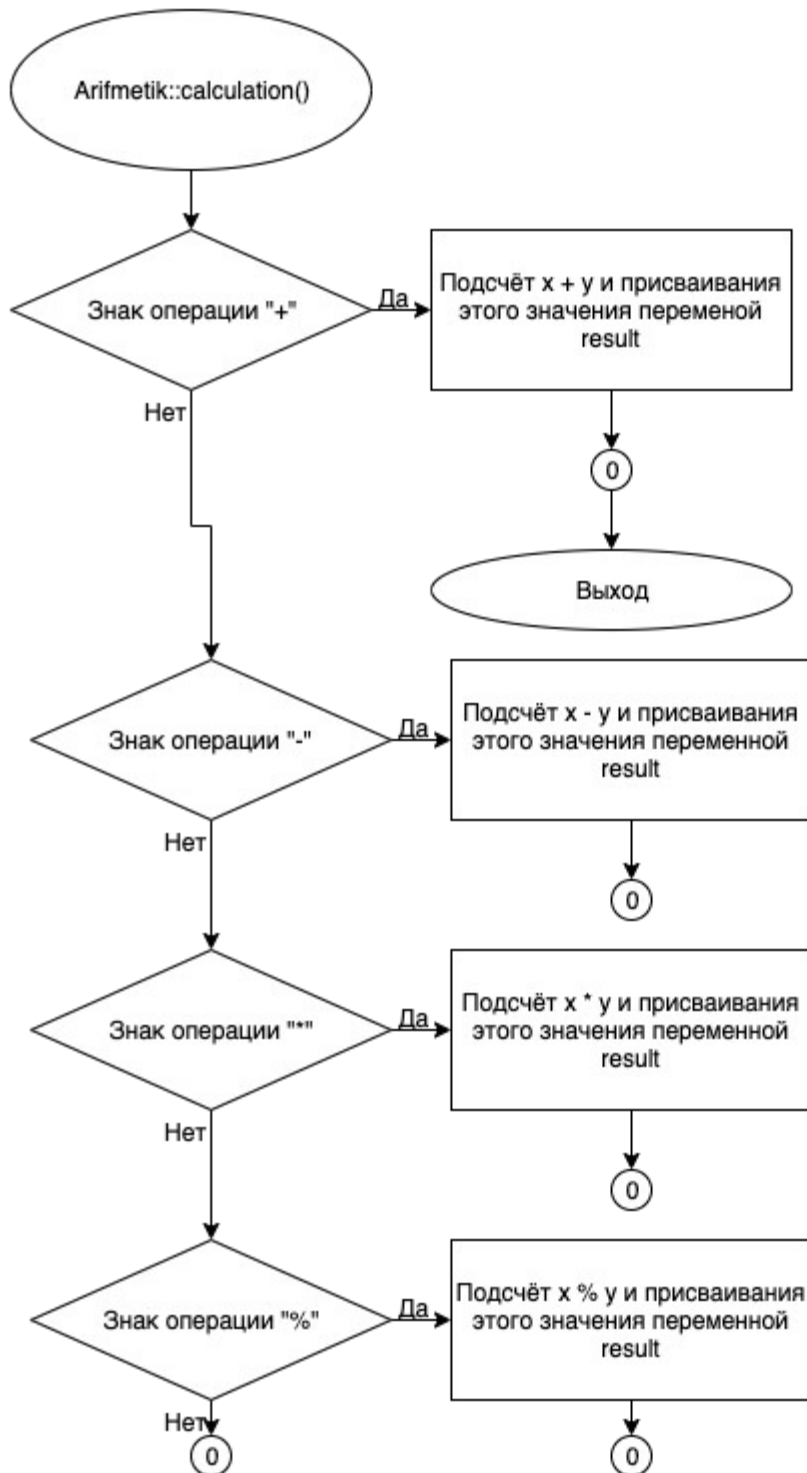


Рисунок 1 – Блок-схема алгоритма

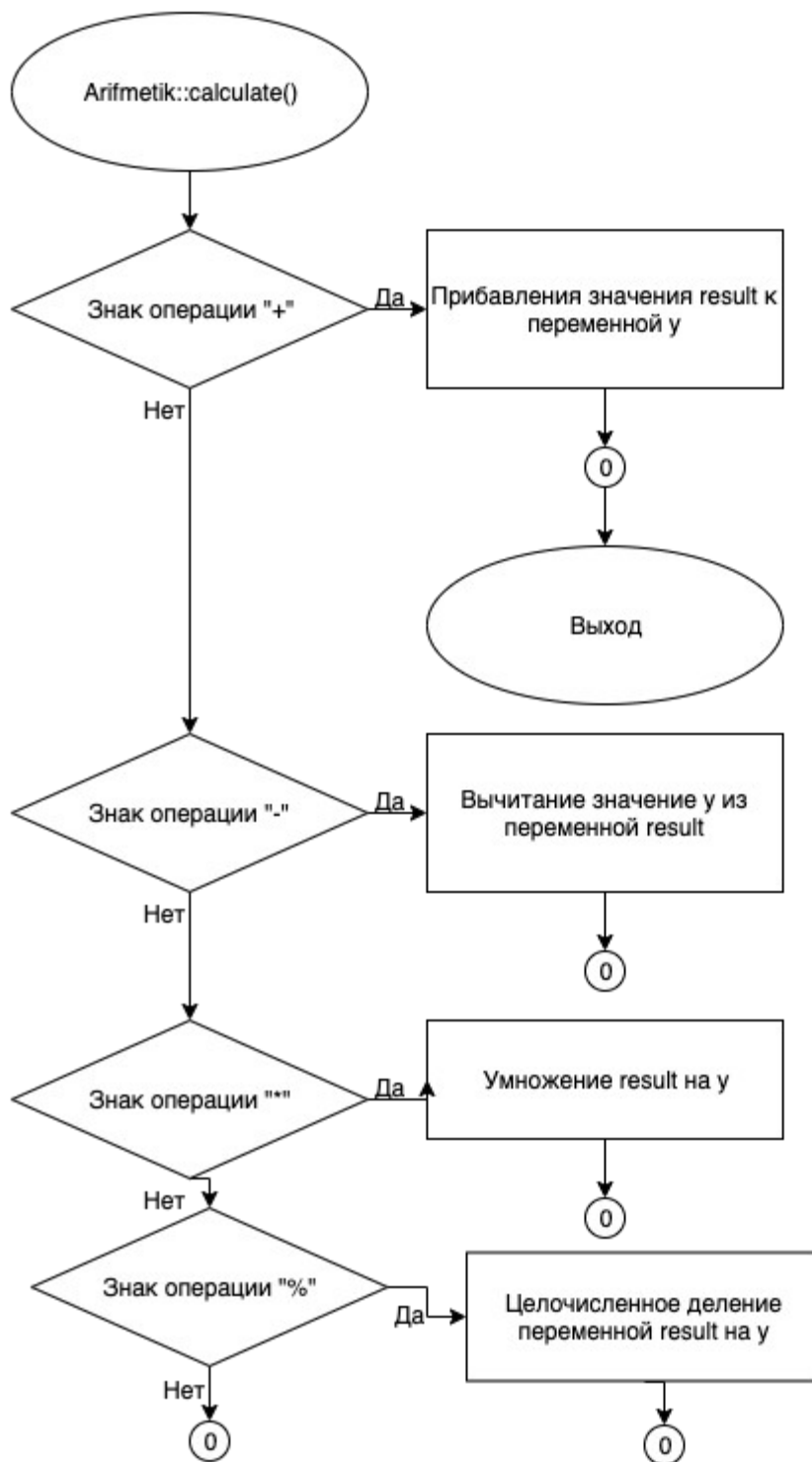


Рисунок 2 – Блок-схема алгоритма

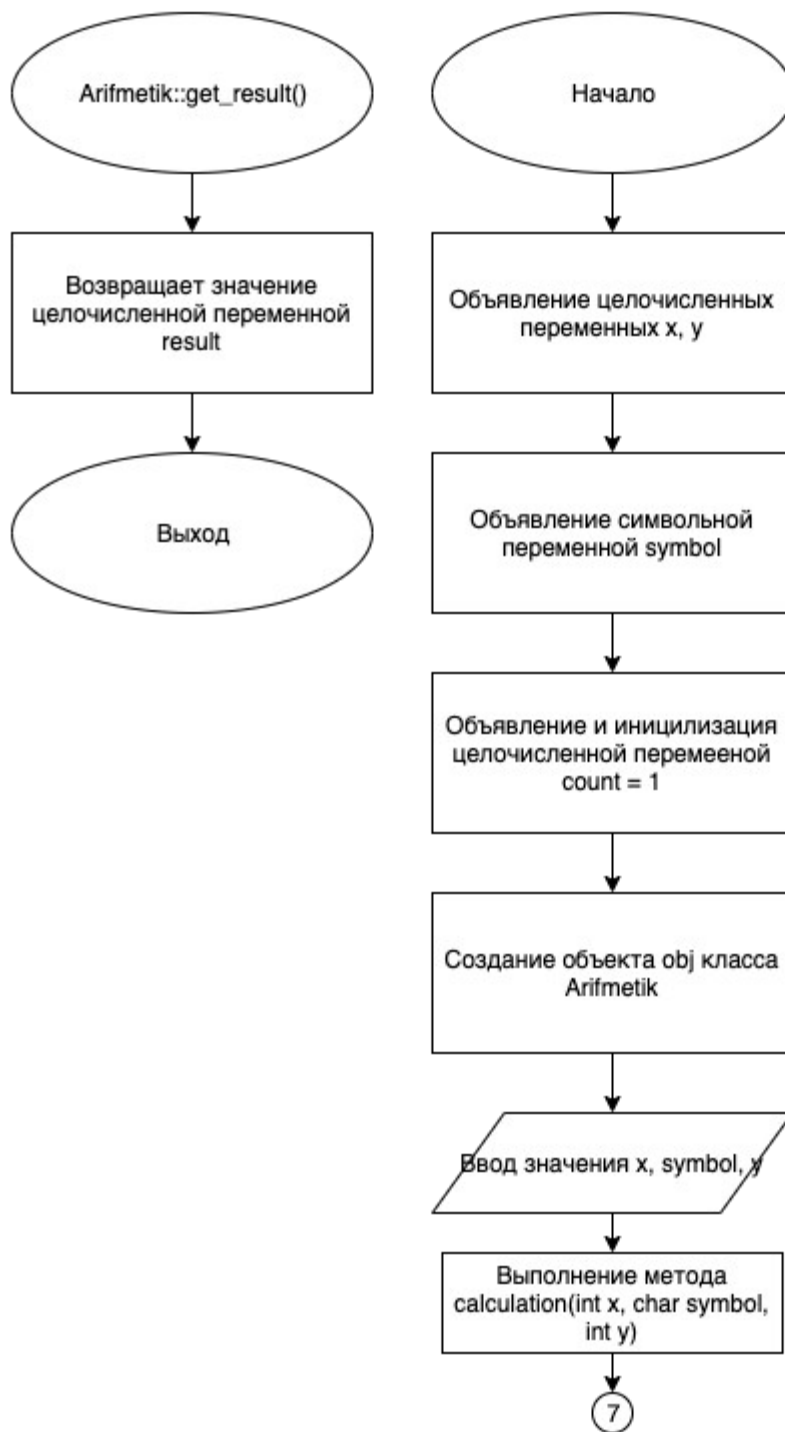


Рисунок 3 – Блок-схема алгоритма

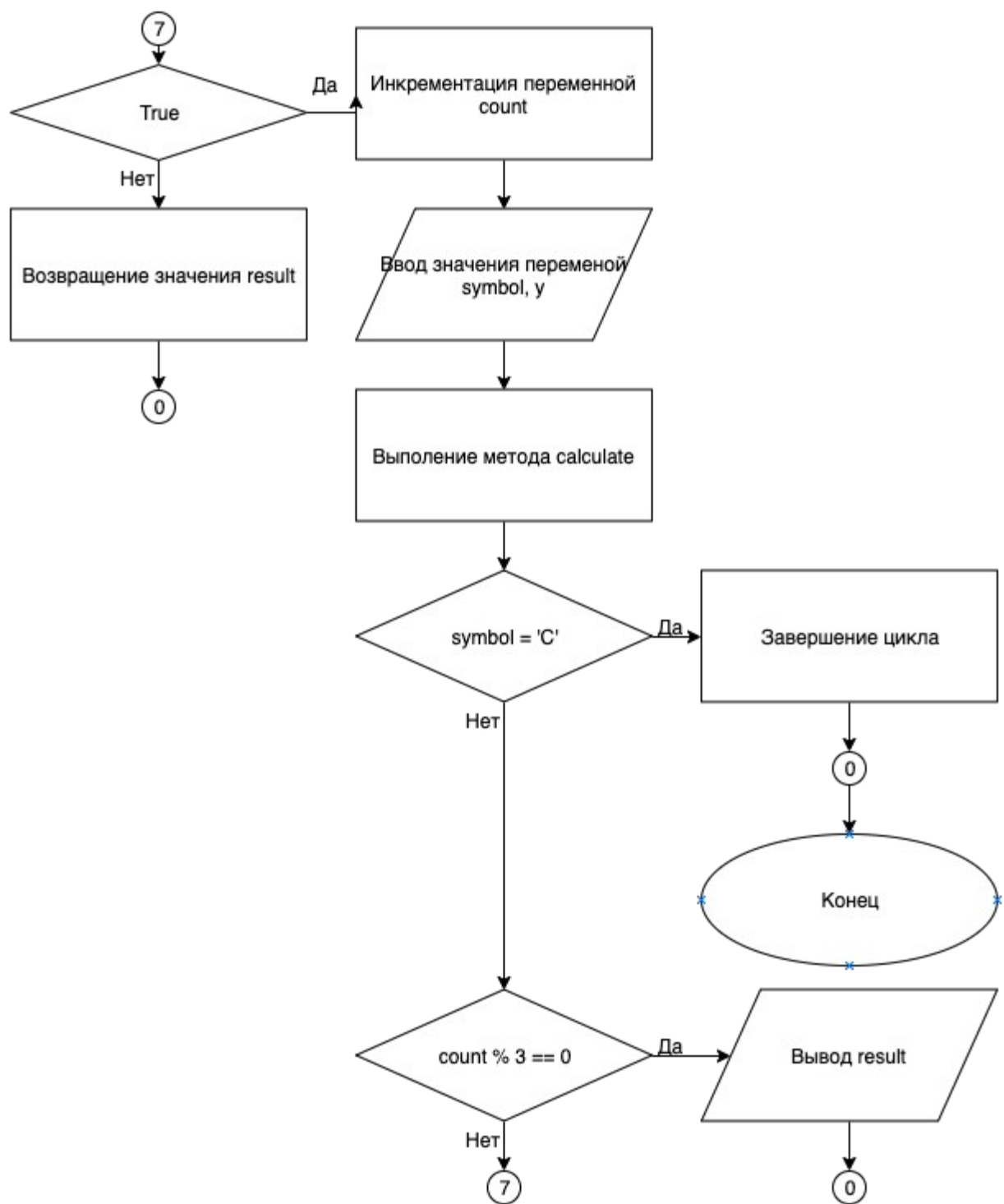


Рисунок 4 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл 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;
        default:
            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;
    }
}
```

```

        default:
            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 = 0;
public:
    void calculation(int x, char symbol, int y);
    void calculate(char symbol, 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;
    int count = 1;

```

```
Arifmetik obj;  
cin >> x >> symbol >> y;  
obj.calculation(x, symbol, y);  
while(true)  
{  
    count++;  
    cin >> symbol >> y;  
    if(symbol == 'C')  
    {  
        break;  
    }  
    obj.calculate(symbol, y);  
    if(count%3==0)  
    {  
        cout << obj.get_result()<< endl;  
    }  
}  
return(0);  
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 - 3 -3 -3 -3 С	-7	-7
1 + 1 +1 +1 +1 С	4	4
2 - 3 -3 -3 -3 -3 -3 -3 -3 С	-7 -16	-7 -16



## 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).