

## 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 (значение можно получить после выполнения трех операций).

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.

## 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

## Описание выходных данных

**Первая строка,** с первой позиции:  
«значение выражения»

**Последующие строки,** с первой позиции:  
«значение выражения»

## Метод решения

потоки ввода и вывода cin/cout

Объект calc класса Calculator

методы класса Calculator:

- void calculation(int, char, int) - выполнение первой операции
- void calculate(char, int) - выполнение второй и далее операции
- int get\_res() - метод получения результата вычислений

свойства класса:

int result - результата вычислений, модификатор доступа - private

## Описание алгоритма

Класс объекта: Calculator

Модификатор доступа: public

Метод: calculation

Функционал: вычисление первой операции

Параметры: int, char, int - два целых числа и символ операции

Возвращаемое значение: -

№	Предикат	Действия	№ перехода	Комментарий
1	знак операции "+"	подсчет $x + y$ и присваивание их значения полю result	Ø	
			2	
2	знак операции "-"	подсчет $x - y$ и присваивание их значения полю result	Ø	
			3	
3	знак операции "*"	подсчет $x * y$ и присваивание их значения полю result	Ø	
			4	
4	знак операции "%"	подсчет $x * y$ и присваивание их значения полю result	Ø	
			Ø	

Класс объекта: Calculator

Модификатор доступа: public

Метод: calculate

Функционал: вычисление второй и далее операций

Параметры: char, int

Возвращаемое значение: -

№	Предикат	Действия	№ перехода	Комментарий
1	знак операции "+"	прибавление к полю result значения поля y	Ø	
			2	
2	знак операции "-"	вычитание из поля result значения поля y	Ø	
			3	
3	знак операции "*"	умножение поля result на значения поля y	Ø	
			4	
4	знак операции "%"	целочисленное деление поля result на значения поля y	Ø	
			Ø	

Класс объекта: Calculator

Модификатор доступа: public

Метод: get\_res

Функционал: получение результата вычислений

Параметры: -

Возвращаемое значение: int - результата вычислений

№	Предикат	Действия	№ перехода	Комментарий
1		возврат поля result	Ø	

Функция: main

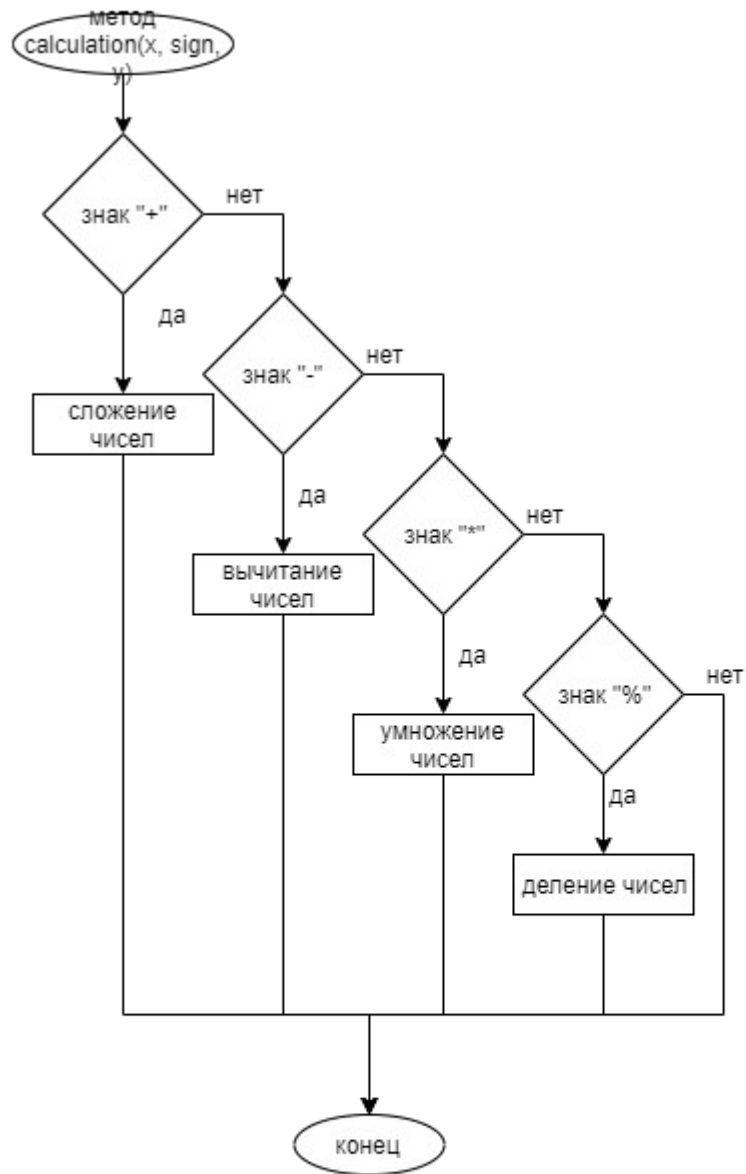
Функционал: основная программа

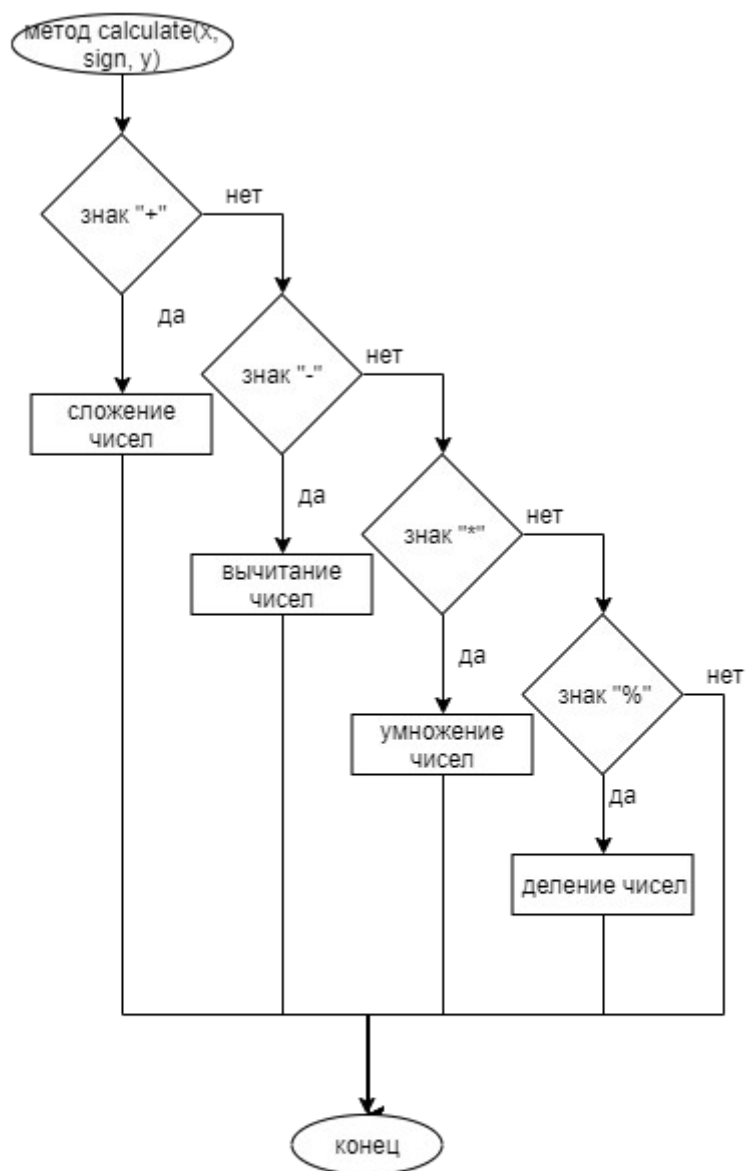
Параметры: -

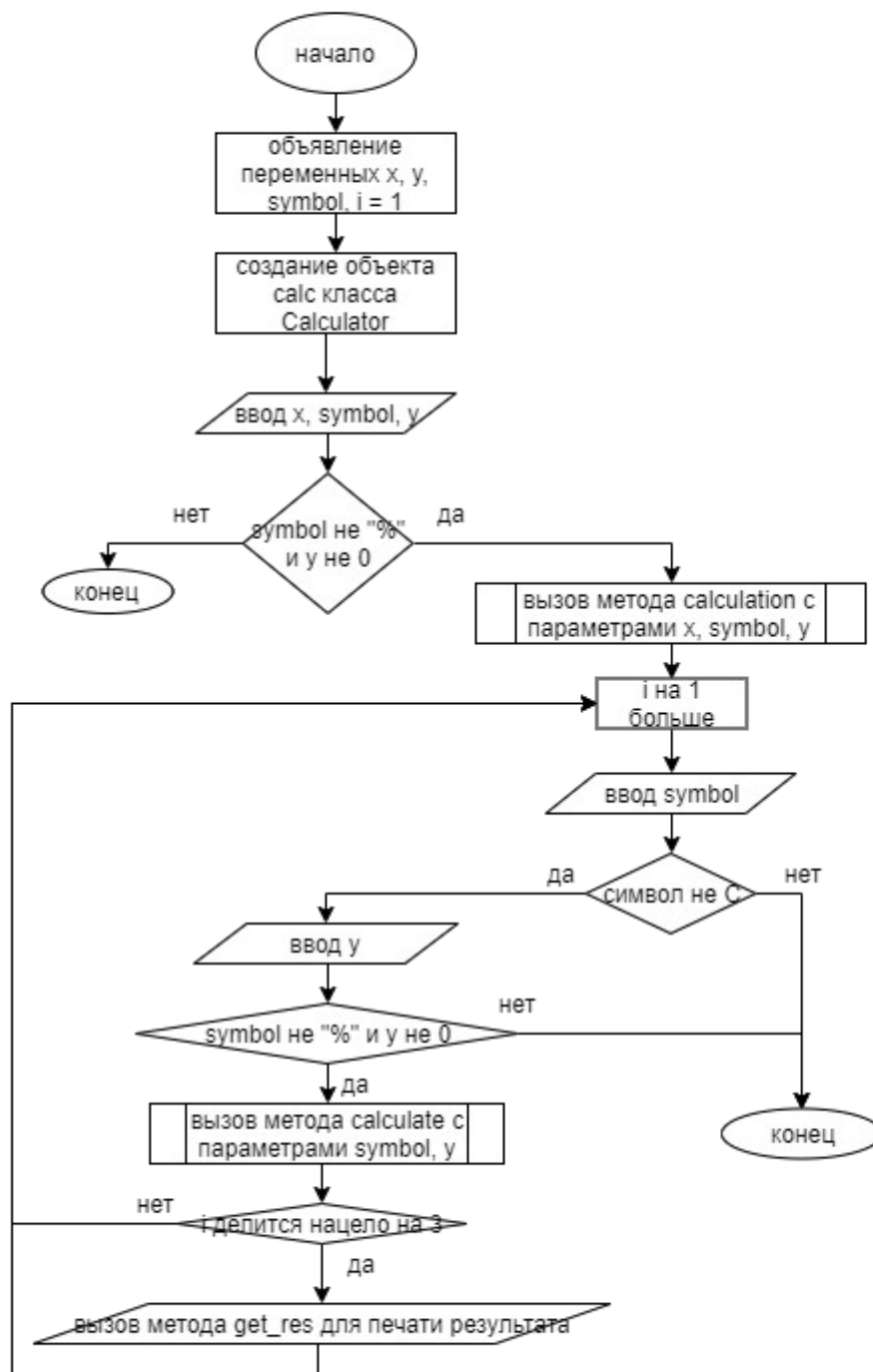
Возвращаемое значение: int - код выполнения программы

№	Предикат	Действия	№ перехода	Комментарий
1		объявление переменных x, y, symbol	2	
2		создание объекта calc класса Calculator	3	
3		ввод x, symbol, y	4	
4	symbol не "%" и y не 0		5	
			Ø	
5		вызов метода calculation с параметрами x, symbol, y	6	
6		ввод symbol	7	
7	symbol не "C"	ввод y	8	
			Ø	
8	symbol не "%" и y не 0	вызов метода calculate с параметрами symbol, y	9	
			Ø	
9	операция - каждая третья	вывод результата методом get_res()	6	
			6	

### Блок-схема алгоритма







**Код программы**

**Файл Calculator.cpp**

```

#include "Calculator.h"
#include <iostream>
void Calculator::calculation(int x, char sign, int y)
{
    switch(sign)
    {
        case('+'):
            result = x + y;
            break;
        case('-'):
            result = x - y;
            break;
        case('*'):
            result = x * y;
            break;
        case('%'):
            result = x % y;
            break;
    }
}
void Calculator::calculate(char sign, int y)
{
    switch(sign)
    {
        case('+'):
            result += y;
            break;
        case('-'):
            result -= y;
            break;
        case('*'):
            result *= y;
            break;
        case('%'):
            result %= y;
            break;
    }
}
int Calculator::get_res()
{
    return result;
}

```

## Файл Calculator.h

```

#ifndef CALCULATOR_H
#define CALCULATOR_H
class Calculator
{
private:

```



```

        int result = 0;
public:
    void calculation(int x, char sign, int y);
    void calculate(char sign, int y);
    int get_res();
};
#endif

```

## Файл main.cpp

```

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

int main()
{
    int x, y;
    char symbol;
    Calculator calc;
    cin >> x >> symbol >> y;
    if (y != 0 or symbol != '%')
    {
        calc.calculation(x, symbol, y);
        for (int i = 2; symbol != 'C' and (symbol == '+' or symbol ==
        '-' or symbol == '*' or symbol == '%'); i++)
        {
            cin >> symbol;
            if (symbol != 'C')
            {
                cin >> y;
                if (y != 0 or symbol != '%')
                {
                    calc.calculate(symbol, y);
                    if (i % 3 == 0)
                    {
                        if (i != 3) cout << endl;
                        cout << calc.get_res();
                    }
                }
            }
            else
            {
                return 0;
            }
        }
        else return 0;
    }
    else cout << "NaN";
    return 0;
}

```

## Тестирование

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
$343 + 32 - 6 + 5 \text{ C}$	374	374
$132 - 4 * 6 * 1 + 3 + 5 - 4 \text{ C}$	768 772	768 772
$100 + 5 - 23 * 6 \text{ C}$	492	492
$1 + 1 - 1 * 1 \% 1 \text{ C}$	1	1