

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

Воспользуемся объектом `q` класса `number`

- класс `number`
 - Поля, отвечающие за вычисление целочисленного арифметического выражения
 - Наименования - `number`
 - Тип - целочисленные
 - Модификатор доступа - `private`

Методы

- Метод `functionFirstResult()` класса `number()` - выполняет первую операцию выражения
- Метод `functionSecondResult()` класса `number()` - выполняет вторую операцию выражения
- Метод `functionFirstResultCout()` класса `number()` - возвращает значение вычисленного выражения
- Метод `functionSecondResultCout()` класса `number()` - возвращает значение вычисленного выражения

Для решения этой задачи воспользуемся объектами стандартных потоков ввода и вывода `cin`, `cout`, условным оператором `if`, оператором цикла с предусловием `while` из библиотеки `<iostream>`

Описание алгоритма

Функция: `main`

Функционал: точка входа в программу

Параметры: -

Возвращаемое значение: int, 0

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление firstNumber	2	Целочисленная переменная
2		Объявление secondNumber	3	Целочисленная переменная
3		Объявление i	4	Переменная i служит счетчиком в цикле
4		Объявление oper	5	Строковый литерал
5		Создание объекта q класса number	6	
6		Ввод целочисленных переменных firstNumber, secondNumber и строкового литерала oper	7	
7		Вызов метода fuctionFirstResult	8	Выполнение первой операции
8		Ввод целочисленной переменной secondNumber и строкового литерала oper	9	
9	oper == 'C'	Завершение программы	∅	Выполнение функции exit(0)
			10	
10		Вызов метода fuctionSecondResult	11	Выполнение второй операции
11		Ввод целочисленной переменной secondNumber и строкового литерала oper	12	
12	oper == 'C'	Завершение программы	∅	
			13	
13		Вызов метода functionSecondResult	14	Выполнение второй операции
14		Вызов метода functionFirstResultCout	15	Вывод первой операции
15	true	Увеличение i на 1	16	Бесконечный цикл while, условие true - из-за того, что не знаем точного количества выражения
			∅	
16		Ввод целочисленной переменной secondNumber и строкового литерала oper	17	
17	oper == 'C'	Завершение программы	∅	
			18	
18		Вызов метода functionSecondResult	19	
19	(i%3 == 0)	Вызов метода	15	

	functionSecondResultCout	15	
--	--------------------------	----	--

Класс объекта: number

Модификатор доступа: public

Метод: fuctionFirstResult

Функционал: Выполнение первой операции

Параметры: целочисленные переменные firstNumber,SecondNumber и символьная переменная oper

Возвращаемое значение: loveAvrora

№	Предикат	Действия	№ перехода	Комментарий
1	oper == '+'	Выполнение арифметической операции сложения	1	
	oper == '-'	Выполнение арифметической операции вычитания	1	
	oper == '*'	Выполнение арифметической операции умножения	1	
	oper == '%'	Выполнение арифметической операции деления с остатком на 2	1	

Класс объекта: number

Модификатор доступа: public

Метод: functionSecondResult

Функционал: Выполнение второй операции

Параметры: целочисленная переменная secondNumber и символьная переменная oper

Возвращаемое значение: loveAvrora

№	Предикат	Действия	№ перехода	Комментарий
1	oper == '+'	Выполнение арифметической операции сложения	1	
	oper == '-'	Выполнение арифметической операции вычитания	1	
	oper == '*'	Выполнение арифметической операции умножения	1	

oper == '%'	Выполнение арифметической операции деления остатка на 2	1	
-------------	---	---	--

Класс объекта: number

Модификатор доступа: public

Метод: functionSecondResultCout

Функционал: Возвращает значение целочисленного выражения

Параметры: -

Возвращаемое значение: int

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод строки	2	
2		Вывод значения n	∅	

Класс объекта: number

Модификатор доступа: public

Метод: functionFirstResultCout

Функционал: Возвращает значение целочисленного выражения

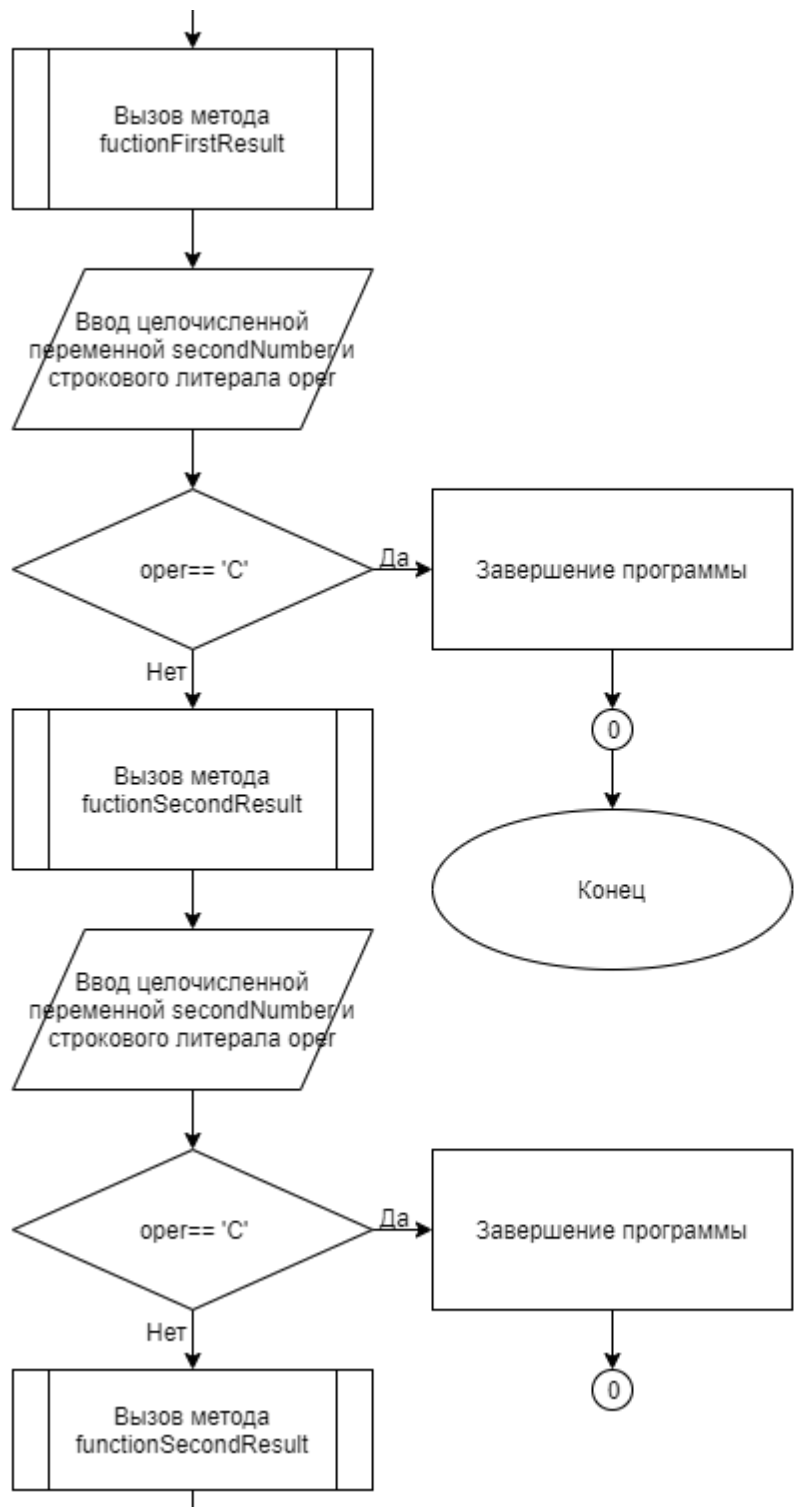
Параметры: -

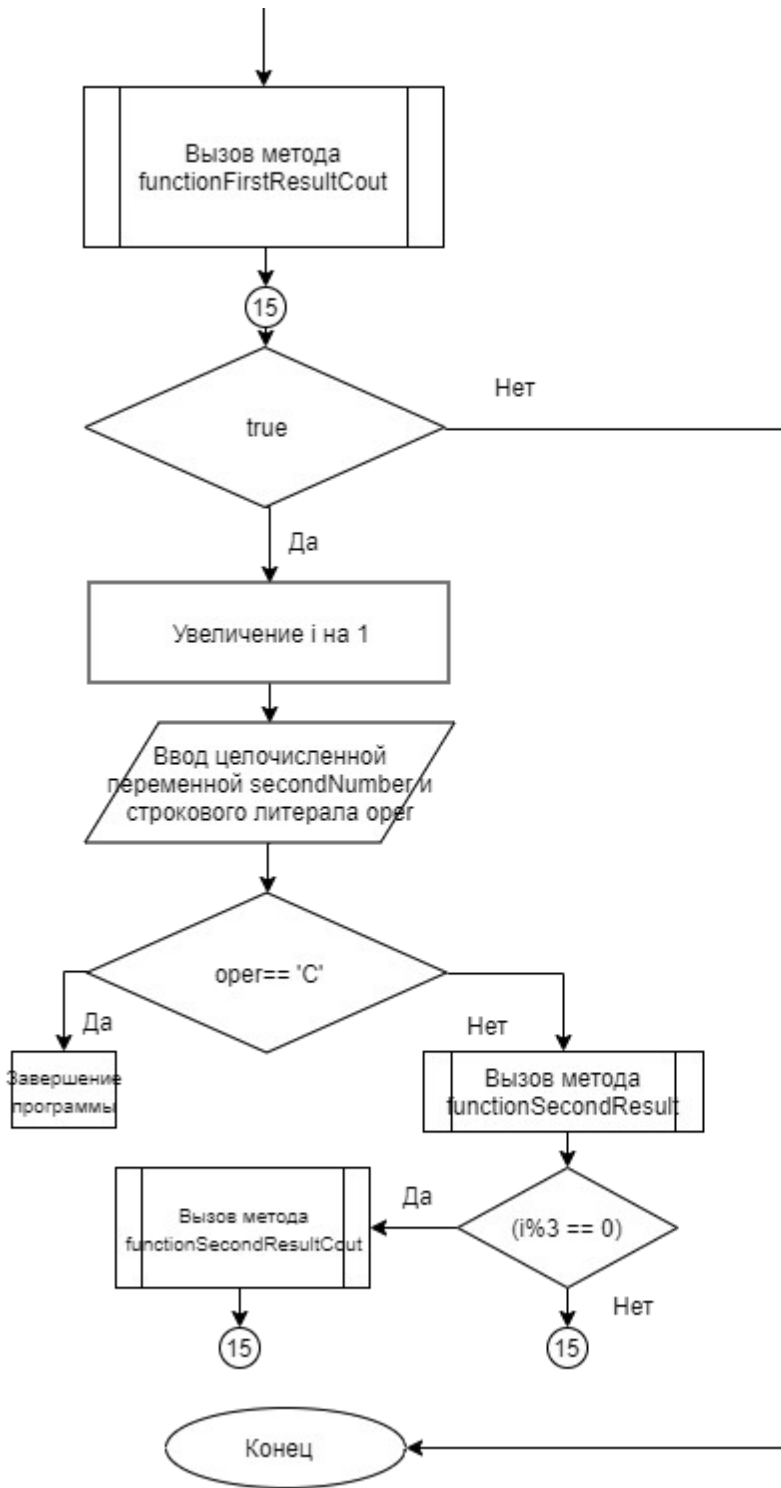
Возвращаемое значение: int

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод значения n	∅	

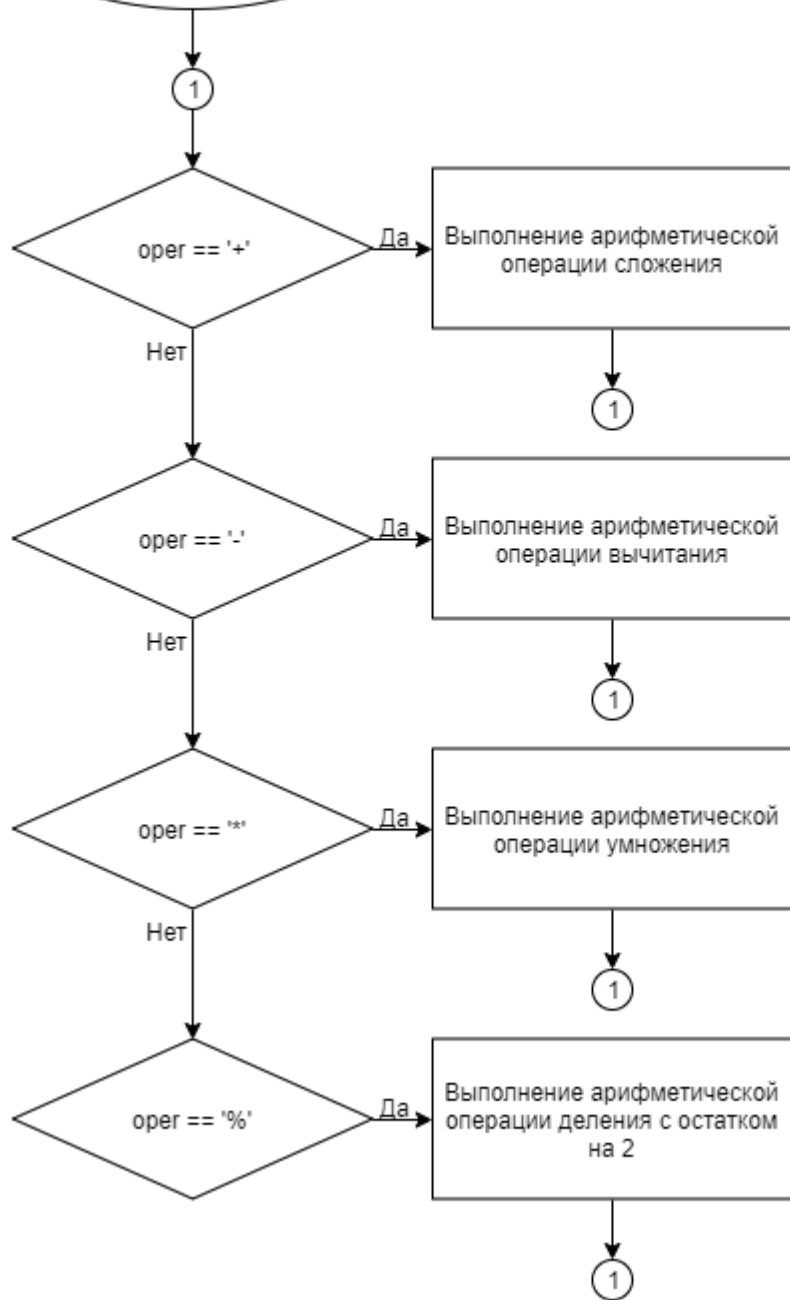
Блок-схема алгоритма



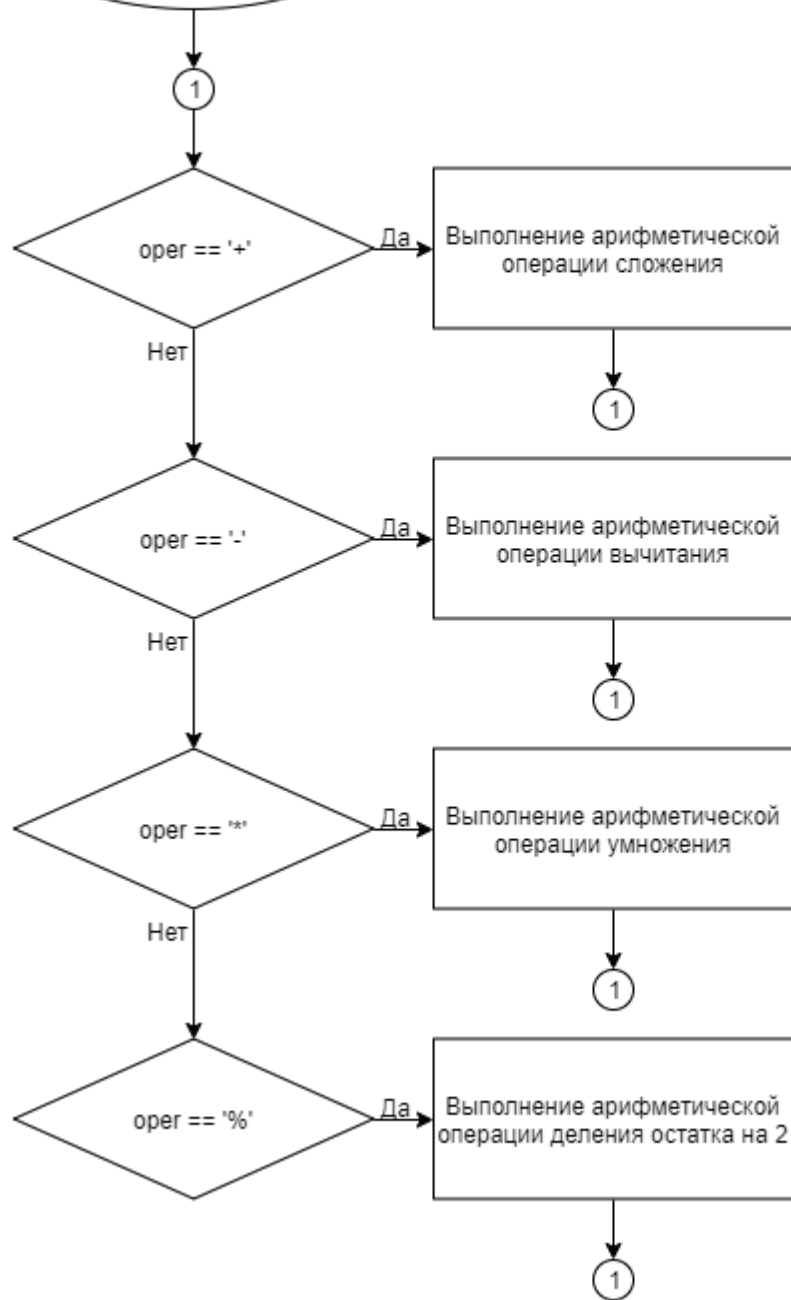


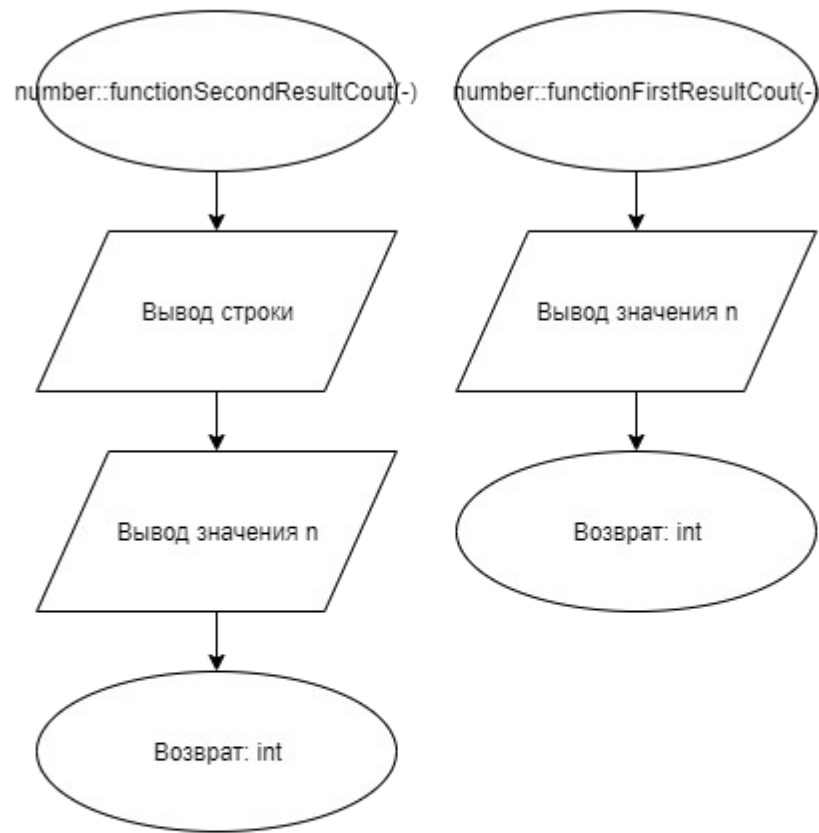


number::fuctionFirstResult(целочисленные переменные firstNumber,SecondNumber и символьная переменная oper)



number.functionSecondResult(целочисленная переменная secondNumber и символьная переменная oper)





Код программы

Файл `header.cpp`

```

#include "header.h"
int number::functionFirstResult(int firstNumber, char oper, int
secondNumber) {
    if(oper == '+') {
        loveAvrora = firstNumber + secondNumber;
    }
    if(oper == '-') {
        loveAvrora = firstNumber - secondNumber;
    }
    if( oper == '*') {
        loveAvrora = firstNumber * secondNumber;
    }
    if(oper == '%') {
        loveAvrora = firstNumber % secondNumber;
    }
}
int number::functionSecondResult(int secondNumber, char oper) { if(
oper == '+') {
    loveAvrora = loveAvrora + secondNumber;
}
if(oper == '-') {
    loveAvrora = loveAvrora - secondNumber;
}
if( oper == '*') {
    loveAvrora = loveAvrora * secondNumber;
}
if(oper == '%') {
    loveAvrora = loveAvrora % secondNumber;
}
}
int number::functionFirstResultCout() {
    cout << loveAvrora;
}
int number::functionSecondResultCout() {
    cout << endl << loveAvrora;
}
}

```

Файл header.h

```

#ifndef number_h
#define number_h
#include <iostream>
using namespace std;
class number {
    int loveAvrora;
public:
    int fuctionFirstResult(int firstNumber,
char oper, int secondNumber); int
functionSecondResult(int secondNumber,
char oper);
int functionFirstResultCout();

```

```
        int functionSecondResultCout();
};
#endif
```

Файл main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "header.h"
int main() {
    int firstNumber, secondNumber;
    int i=0;
    char oper;
    number q;
    cin >> firstNumber >> oper >> secondNumber;
    q.functionFirstResult(firstNumber, oper, secondNumber);
    cin >> oper >> secondNumber;
    if( oper == 'C') {
        exit(0); // break?
    }
    //2
    q.functionSecondResult(secondNumber, oper);
    cin >> oper >> secondNumber;
    if(oper == 'C') {
        exit(0); // break?
    }
    q.functionSecondResult(secondNumber, oper); q.functionFirstResultCout();
    while(true){
        i++;
        cin >> oper>> secondNumber;
        if( oper == 'C') {
            exit(0); //break?
        }
        q.functionSecondResult(secondNumber, oper);
        if(i % 3 == 0) {
            q.functionSecondResultCout();
        }
    }
}
```

Тестирование

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
$1+1 +1 +1 +1 +1 +1C$	4 7	4 7
$-1-1-1-1C$	-4	-4
$-1+1 +1 +1C$	2	2
$1+3 +1 +2C$	7	7
$1+1 +1 +1C$	4	4
$2+2+2+2C$	8	8