МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

Московский технический университет связи и информатики

Кафедра «Информатика»

Лабораторный практикум по дисциплине

Введение в информационные технологии

Лабораторная работа № 3

«Разработка простейших проектов с использованием основных средств языка программирования C++»

Выполнил:

студент группы БПС2601

Миронов М.А.

Проверил:

доц. Рюмин Ю.А.

Москва 2026

СОДЕРЖАНИЕ

1 Цель работы 3

2 Общее задание 4

3 Индивидуальное задание 5

4 Формализация и уточнение задания 6

4.1 Математическая модель и программное выражение 6

4.2 Типы данных и область допустимых значений 6

4.3 Преобразования и инкременты 7

5 Схема алгоритма решения задачи 8

6 Код программного проекта 9

7 Результаты выполнения проекта 11

8 Доказательство правильности и объяснение результатов 12

9 Неявные преобразования типов данных 13

10 Вывод 14

1 Цель работы

Изучить основные средства языка C++, типы данных и правила преобразования вещественных значений в целые. Реализовать вычисление арифметического выражения для индивидуального варианта, освоить явное и неявное преобразование типов, округление с помощью функций ceil и floor, а также префиксный и постфиксный инкременты.

2 Общее задание

1. Изучить основные средства языка программирования C++, типы данных и их представление в оперативной памяти компьютера.
2. Провести формализацию заданного арифметического выражения. Записать выражение по правилам языка программирования с учётом приоритета операций.
3. Реализовать преобразование вещественного числа в целое с усечением (явное и неявное), с округлением в большую сторону и с округлением в меньшую сторону. Записать четыре оператора присваивания результатов четырём переменным целого типа.
4. Записать операции префиксного и постфиксного инкремента для переменных, получивших значения усечением в результате неявного и явного преобразования вещественного числа в целый тип.
5. Записать операции вывода всех используемых переменных и инкрементных выражений. Повторить вывод переменных после вывода инкрементных выражений.
6. Создать схему алгоритма решения задачи (функции main) в виде SVG-рисунка с помощью редактора в Linux.
7. Создать программный код проекта с функцией main решения задачи средствами C++.
8. Получить результаты не менее чем с тремя наборами исходных данных, проанализировать, объяснить и доказать их правильность. Перечислить все неявные преобразования типов данных при выполнении программного проекта.

3 Индивидуальное задание

Вариант 16. Вычислить значение выражения

$$b = \frac{\log_{10} x - \sin^2(xy)}{0.8 \cdot (\ln(1 - x))^2} \tag{1}$$

при исходных данных $x = 0.501$ и $y = 1.7$. Реализовать четыре способа преобразования результата b к целому типу, выполнить префиксный инкремент переменной, полученной неявным усечением, и постфиксный инкремент переменной, полученной явным усечением. Вывести исходные данные, значения всех переменных и инкрементных выражений. Выполнить программу ещё на двух наборах исходных данных и проверить результаты.

4 Формализация и уточнение задания

4.1 Математическая модель и программное выражение

В числителе используется десятичный логарифм $\log_{10} x$ и квадрат синуса произведения xy. В знаменателе вычисляется квадрат натурального логарифма $\ln(1 - x)$. Тригонометрические функции языка C++ принимают аргумент в радианах, поэтому дополнительное преобразование угла не требуется.

Для проверки вычисление удобно представить через промежуточные значения:

$$\text{numerator} = \log_{10} x - \sin^2(xy), \tag{2}$$

$$\text{denominator} = 0.8 \cdot (\ln(1 - x))^2, \tag{3}$$

$$b = \frac{\text{numerator}}{\text{denominator}}. \tag{4}$$

В программе всё выражение записывается одним оператором присваивания. Для вычислений используются функции log10, sin, log и pow.

4.2 Типы данных и область допустимых значений

Типы переменных приведены в таблице 1.

Таблица 1 — Типы данных программного проекта

Переменные	Тип	Назначение
x, y	double	Исходные вещественные данные
b	double	Вещественный результат вычисления
k, l, m, n	int	Результаты преобразования к целому типу

Для существования $\log_{10} x$ необходимо условие $x > 0$, а для существования $\ln(1 - x)$ — условие $x < 1$. При $0 < x < 1$ значение $\ln(1 - x)$ отлично от нуля, поэтому знаменатель положителен и деление на ноль исключено. Таким образом, область допустимых значений:

$$0 < x < 1, \quad y \in \mathbb{R}. \tag{5}$$

4.3 Преобразования и инкременты

Для изучения преобразований используются следующие операторы:

```
k = b;
l = int(b);
m = ceil(b);
n = floor(b);
```

В первом операторе выполняется неявное преобразование double в int, а во втором — явное преобразование. Оба преобразования усекают дробную часть и приводят значение к нулю. Функции ceil и floor возвращают double, после чего результат неявно преобразуется в int.

Для переменной k применяется префиксный инкремент ++k: значение увеличивается до вычисления выражения вывода. Для переменной l применяется постфиксный инкремент l++: в выражении вывода используется прежнее значение, после чего переменная увеличивается на единицу.

5 Схема алгоритма решения задачи

Алгоритм функции main включает ввод исходных данных, вычисление выражения, преобразование результата к целому типу, вывод переменных и выполнение инкрементов. Основная последовательность действий представлена на рисунке 1.

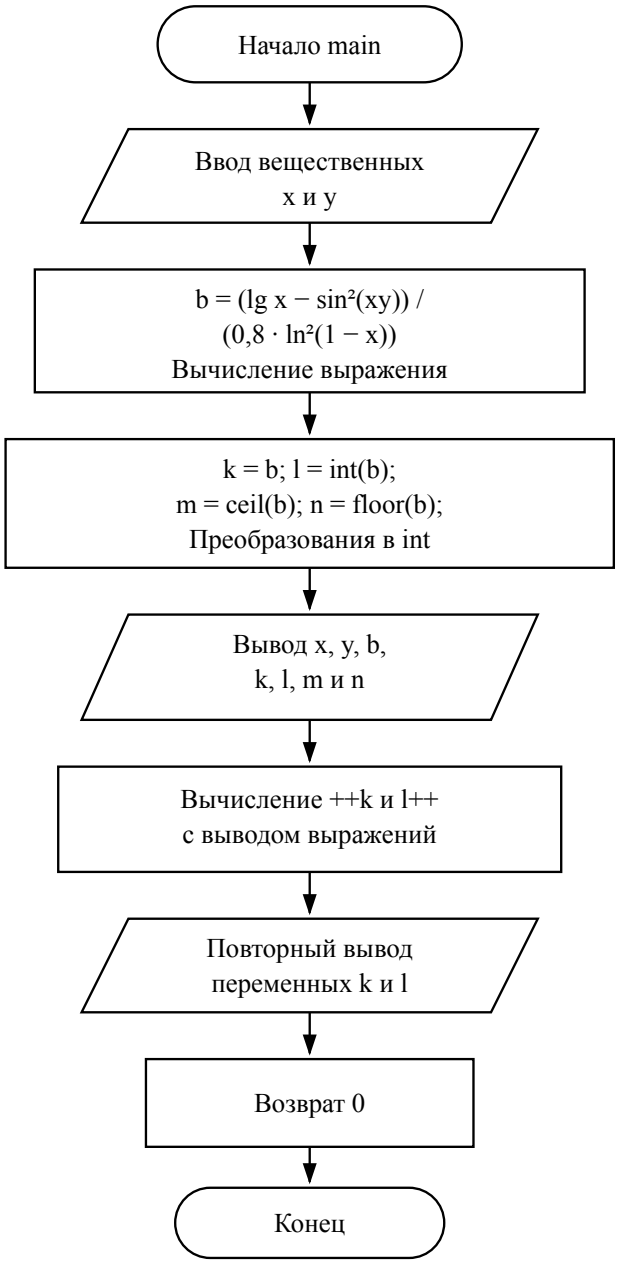


Рисунок 1 — Блок-схема алгоритма функции main

Программа рассчитана на ввод значений из указанной области допустимых значений. После успешного выполнения возвращается код 0.

6 Код программного проекта

Программа реализована на C++20 в файле main.cpp. Первая часть кода, содержащая ввод данных и вычисление выражения, приведена в листинге 1. Преобразования и операции вывода представлены в листинге 2.

```
#include <clocale>
#include <cmath>
#include <iomanip>
#include <iostream>

using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    double x, y, b;
    int k, l, m, n;

    cout << "Введите x: ";
    cin >> x;
    cout << "Введите y: ";
    cin >> y;

    b = (log10(x) - pow(sin(x * y), 2)) / (0.8 * pow(log(1 - x), 2));
```

Листинг 1 — Код программы: ввод данных и вычисление выражения

```
    k = b;
    l = int(b);
    m = ceil(b);
    n = floor(b);

    cout << fixed << setprecision(10);
    cout << "x = " << x << ", y = " << y << endl;
    cout << "Вещественное b = " << b << endl;
    cout << "Неявное усечение k = " << k << endl;
    cout << "Явное усечение l = " << l << endl;
    cout << "Округление в большую сторону m = " << m << endl;
    cout << "Округление в меньшую сторону n = " << n << endl;

    cout << "Значение префиксного инкремента ++k = " << ++k << endl;
    cout << "Значение постфиксного инкремента l++ = " << l++ << endl;
    cout << "Значение k после приращения = " << k << endl;
    cout << "Значение l после приращения = " << l << endl;

    return 0;
}
```

Листинг 2 — Код программы: преобразования, инкременты и вывод

Для сборки проекта используется файл CMakeLists.txt. Конфигурация, сборка и запуск в терминале Linux выполняются командами:

```
cmake -S . -B build
cmake --build build
./build/lab3
```

7 Результаты выполнения проекта

Программа была собрана компилятором Clang 21.1.8 без ошибок и предупреждений и выполнена на трёх наборах исходных данных. Первый набор соответствует индивидуальному заданию, остальные используются для дополнительной проверки. Результаты вычислений и преобразований приведены в таблице 2.

Таблица 2 — Результаты вычисления выражения и преобразования типов

№	x	y	b	k	l	m	n
1	0,501	1,7	−2,2408213917	−2	−2	−2	−3
2	0,25	0,5	−9,3281250899	−9	−9	−9	−10
3	0,75	−1,2	−0,4803668117	0	0	0	−1

Значения инкрементных выражений и переменных после инкрементов приведены в таблице 3.

Таблица 3 — Результаты префиксного и постфиксного инкрементов

Набор	++k	l++	k после	l после
1	−1	−2	−1	−1
2	−8	−9	−8	−8
3	1	0	1	1

8 Доказательство правильности и объяснение результатов

Правильность вычислений проверена независимым расчётом с использованием математических функций log10, sin, log и pow. Промежуточные значения числителя и знаменателя приведены в таблице 4.

Таблица 4 — Независимая проверка вычислений

Набор	Числитель	Знаменатель	Контрольное b	Расхождение
1	−0,8662699759	0,3865859096	−2,2408213917	0
2	−0,6176037805	0,0662087798	−9,3281250899	0
3	−0,7385397840	1,5374496445	−0,4803668117	0

Контрольные значения полностью совпали с результатами программы с точностью вывода до десяти знаков после запятой. Это подтверждает правильность реализации арифметического выражения.

Для первого набора $b = -2.2408213917$. Неявное и явное усечение дают -2, поскольку преобразование к целому типу отбрасывает дробную часть и приводит число к нулю. Округление в большую сторону также даёт -2, а в меньшую — -3, поскольку $-3 < -2.2408213917 < -2$.

Аналогично, для второго набора усечение и округление вверх дают -9, а округление вниз — -10. Для третьего набора результат находится между -1 и 0: усечение и округление вверх дают 0, округление вниз — -1. Эти результаты соответствуют определениям функций ceil и floor.

Префиксный инкремент сначала увеличивает k, поэтому выражение ++k и последующее значение k совпадают. Постфиксный инкремент возвращает прежнее значение l, после чего переменная увеличивается на единицу. Например, в первом наборе l++ выводит -2, а следующее значение l равно -1. Результаты программы и независимой проверки совпали.

9 Неявные преобразования типов данных

При выполнении программы осуществляются следующие неявные преобразования:

1. В выражении 1 - x целая константа 1 типа int преобразуется к типу double перед вычитанием.
2. В вызове pow(sin(x * y), 2) целый аргумент 2 преобразуется к типу double.
3. В вызове pow(log(1 - x), 2) целый аргумент 2 также преобразуется к типу double.
4. В операторе k = b значение b типа double преобразуется к типу int переменной k с усечением дробной части.
5. В операторе m = ceil(b) вещественное значение, возвращённое функцией ceil, преобразуется к типу int.
6. В операторе n = floor(b) вещественное значение, возвращённое функцией floor, преобразуется к типу int.

Преобразование в операторе l = int(b) является явным и в этот список не входит. Константа 0.8 и переменные x и y уже имеют тип double. Операции инкремента выполняются над переменными типа int без изменения их типа.

10 Вывод

В ходе выполнения лабораторной работы были изучены основные средства C++, вещественный и целый типы данных, способы преобразования double в int и операции инкремента. Для варианта 16 было формализовано заданное арифметическое выражение, определена область допустимых значений и построена блок-схема алгоритма функции main.

Был написан и собран программный проект, реализующий ввод исходных данных, вычисление выражения, явное и неявное усечение, округление в большую и меньшую стороны, а также префиксный и постфиксный инкременты. Программа была выполнена на трёх наборах данных, включая исходные данные варианта.

Правильность результатов подтверждена независимым расчётом числителя, знаменателя и итогового значения. Результаты программы и проверки совпали с точностью до десяти знаков после запятой. Дополнительно были объяснены различия между усечением и округлением отрицательных чисел, а также между префиксным и постфиксным инкрементами.