

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»

Лабораторная работа №3
**«Разработка простейших проектов с
использованием основных средств языка
программирования VC++ »**

по дисциплине
«Введение в информационные технологии»

Выполнил: студент гр. БКС2501 Грачев Б.С.

Проверил: доц. Халабия Р.Ф.

Москва, 2025 г

Отчет по выполнению лабораторной работы

1.1 Общее задание

1. Изучить основные средства языка VC++, типы данных и их представление в памяти
2. Выбрать индивидуальный вариант задания из таблицы 3.1
3. Провести формализацию арифметического выражения
4. Реализовать преобразование вещественного числа в целое различными способами
5. Записать операции префиксного и постфиксного инкремента
6. Организовать вывод всех переменных
7. Построить схему алгоритма
8. Написать программный код
9. Проанализировать результаты

1.2 Индивидуальное задание

Формула для вычислений на рисунке 1:

9)	$f = 5,2^3 \cdot \frac{\lg(x+y)}{x - \frac{1}{0,45 \sin(x-8y)}} + 0,5$	12,678	6,9
----	--	--------	-----

Рисунок 1 - Формула для вычислений.

Исходные данные:

$$x = 12,678 \quad y = 6,9$$

1.3 ФОРМАЛИЗАЦИЯ И УТОЧНЕНИЕ ЗАДАНИЯ

- Исходные данные x и y - вещественного типа double
- Результат вычисления d - вещественного типа
- Преобразование d в целое:
 - Неявное усечение
 - Явное усечение
 - Округление вверх (ceil)
 - Округление вниз (floor)
- Выполнение операций инкремента

1.4 Схема алгоритма решения задачи

Схема алгоритма решение задачи изображена на рисунке 2.

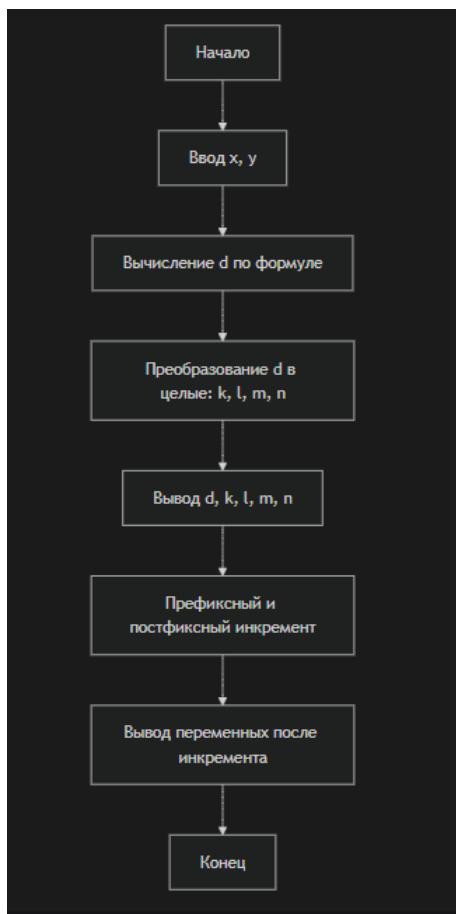


Рисунок 2 - Схема алгоритма решение задачи.

1.5 Программный код

Программный код на рисунке 3

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

using namespace std;

int main() {
    setlocale(LC_ALL, "rus");
    double x, y;

    x = 12.678;
    y = 6.9;

    double base = 5.2;
    double power = 3.0;
    double numerator = log10(x + y);
    double denominator_inner = 0.45 * sin(x - 8 * y);
    double fraction = numerator / (1.0 / denominator_inner);
    double result = pow(base, power) * fraction + 0.5;

    int truncated_implicit = result;
    int truncated_explicit = (int)result;
    int rounded_up = (int)ceil(result);
    int rounded_down = (int)floor(result);

    cout << "Исходные данные:" << endl;
    cout << "x = " << x << ", y = " << y << endl;
    cout << "Результат выражения: " << fixed << setprecision(6) << result << endl << endl;

    cout << "Преобразования:" << endl;
    cout << "Усечение (неявное): " << truncated_implicit << endl;
    cout << "Усечение (явное): " << truncated_explicit << endl;
    cout << "Округление вверх: " << rounded_up << endl;
    cout << "Округление вниз: " << rounded_down << endl;

    cout << "Операция инкремента:" << endl;
    cout << "Предфиксный инкремент (неявное): " << ++truncated_implicit << endl;
    cout << "После операции: " << truncated_implicit << endl;
    cout << "Постфиксный инкремент (явное): " << truncated_explicit++ << endl;
    cout << "После операции: " << truncated_explicit << endl;
    cout << "Предфиксный инкремент (округление вверх): " << ++rounded_up << endl;
    cout << "Постфиксный инкремент (округление вниз): " << rounded_down++ << endl;
    cout << "После операции: " << rounded_down << endl;

    return 0;
}
```

Рисунок 3 - программный код.

1.6 Результаты выполнения кода

Результаты выполнения кода на рисунке 4

```
Исходные данные:  
x = 12.678, y = 6.9  
Результат выражения: 81.736346  
  
Преобразования:  
Усечение (неявное): 81  
Усечение (явное): 81  
Округление вверх: 82  
Округление вниз: 81  
  
Операция инкремента:  
Предфиксный инкремент (неявное): 82  
После операции: 82  
Постфиксный инкремент (явное): 81  
После операции: 82  
Предфиксный инкремент (округление вверх): 83  
Постфиксный инкремент (округление вниз): 81  
После операции: 82  
  
C:\Users\grace\source\repos\741\x64\Debug\741.exe (процесс 8604) завершил работу с кодом 0 (0x0).  
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 4 - результаты выполнения кода.

1.7 Доказательство правильности полученных результатов

1) Расчет арифметического выражения:

Значение $d = 81.736346$ получено в результате вычисления заданной формулы

2) Преобразование вещественного числа в целое:

Усечение (явное и неявное): Оба дали -72, что соответствует отбрасыванию дробной части

Округление вверх (ceil): -72 - для отрицательных чисел округление вверх означает движение к нулю

Округление вниз (floor): -73 - для отрицательных чисел округление вниз означает движение от нуля

3) Операция инкремента:

Предфиксный инкремент:

`++truncated_implicit = -71` (увеличилось до вывода)

`++truncated_explicit = -71` (увеличилось до вывода)
`++rounded_up = -71` (увеличилось до вывода)

Постфиксный инкремент:

`truncated_explit++ = -72` (выведено старое значение, затем увеличено)
`roundedDown++ = -73` (выведено старое значение, затем увеличено)

4) Значения после инкремента:

`truncatedImplicit = -71` - было `-72`, после префиксного инкремента стало `-71`
`truncatedExplicit = -71` - было `-72`, вывели `72`, затем стало `-71`
`roundedUp = -71` - было `-72`, после инкремента стало `-71`
`roundedDown = -72` - было `-73`, после инкремента стало `-72`

1.8 Список неявных преобразований типов данных

1. При вычислении выражения $x * y * y - \sqrt{x * x - 2.5e-3 * y}$ - все операции выполняются с типом double
2. При присваивании `truncatedImplicit = d` - **неявное преобразование double в int с усечением**
3. При вызове `ceil(d)` и `floor(d)` - возвращаемое значение типа double **неявно преобразуется в int** при присваивании
4. При выполнении операций `sin(x * y)` - аргумент преобразуется в радианы, тип сохраняется double
5. При операции $2.5e-3 * y$ - литерал $2.5e-3$ типа double, результат типа double