Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«Вычисление значения функции»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирования»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Клепикова Екатерина Ивановна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

**Цель работы**

Изучить базовую структуру организации программы и основные конструкции языка программирования Pascal.

**Формулировка задания**

**Вариант: 5**

Написать программу, вычисляющую значение функции:

* cos(x) \* lg(x), если x < -9
* lg(x) / sin(x) + tg(x), если -9 <= x < 0
* ln(x) + x², если 0 <= x

Вычислить значение функции на интервале [-11; 2] с шагом 0,3

**Описание алгоритма и ответы на вопросы**

**Алгоритм:**

1. Начало программы
2. Инициализация переменной x = -11.0
3. Цикл while пока x ≤ 2.0
4. Проверка условий:
   * Если x < -9: y = cos(x) \* lg(-x)
   * Если -9 ≤ x < 0: y = lg(-x)/sin(x) + tg(x)
   * Если x ≥ 0: y = ln(x) + x²
5. Вывод результатов
6. Увеличение x на 0.3
7. Конец цикла
8. Завершение программы

**Особенности реализации:**

* Для отрицательных x используется lg(-x) вместо lg(x)
* При x = 0 функция не определена (ln(0))
* При sin(x) = 0 или cos(x) = 0 во втором случае - деление на ноль

**Алгоритм** — это точная конечная последовательность шагов, направленных на решение конкретной задачи или достижение определенной цели. Алгоритм должен обладать свойствами:

* **Дискретность** — разбиение на отдельные шаги
* **Определенность** — однозначность выполнения каждого шага
* **Результативность** — получение результата за конечное число шагов
* **Массовость** — применимость к разным исходным данным

Основные виды алгоритмов:

* **Линейные** — последовательное выполнение операций без ветвлений
* **Разветвляющиеся** — содержащие условия и выбор путей выполнения
* **Циклические** — с повторяющимися блоками операций
* **Рекурсивные** — вызывающие сами себя
* **Вспомогательные** (подпрограммы) — выделенные блоки для многократного использования

**Условный алгоритм** — это алгоритм, в котором выполнение тех или иных операций зависит от выполнения определенных условий. Выбор пути вычислений определяется проверкой логического выражения.

Виды условных алгоритмов:

* **Простая ветвь** — if...then
* **Полная ветвь** — if...then...else
* **Множественное ветвление** — case...of или вложенные if...else

**В данной задаче использовались:**

* **Множественное ветвление** — конструкция if...else if...else для выбора одной из трех формул в зависимости от значения x
* **Полная ветвь** — внутри второго условия дополнительная проверка if...then...else для избежания деления на ноль

**Циклический алгоритм** — это алгоритм, в котором некоторая последовательность действий выполняется многократно до выполнения определенного условия. Циклы используются для обработки множества данных или повторяющихся вычислений.

Виды циклических алгоритмов:

* **Цикл с предусловием** — while...do (проверка условия перед выполнением)
* **Цикл с постусловием** — repeat...until (проверка условия после выполнения)
* **Цикл с параметром** — for...to/downto...do (фиксированное число повторений)

**В данной задаче использовался:**

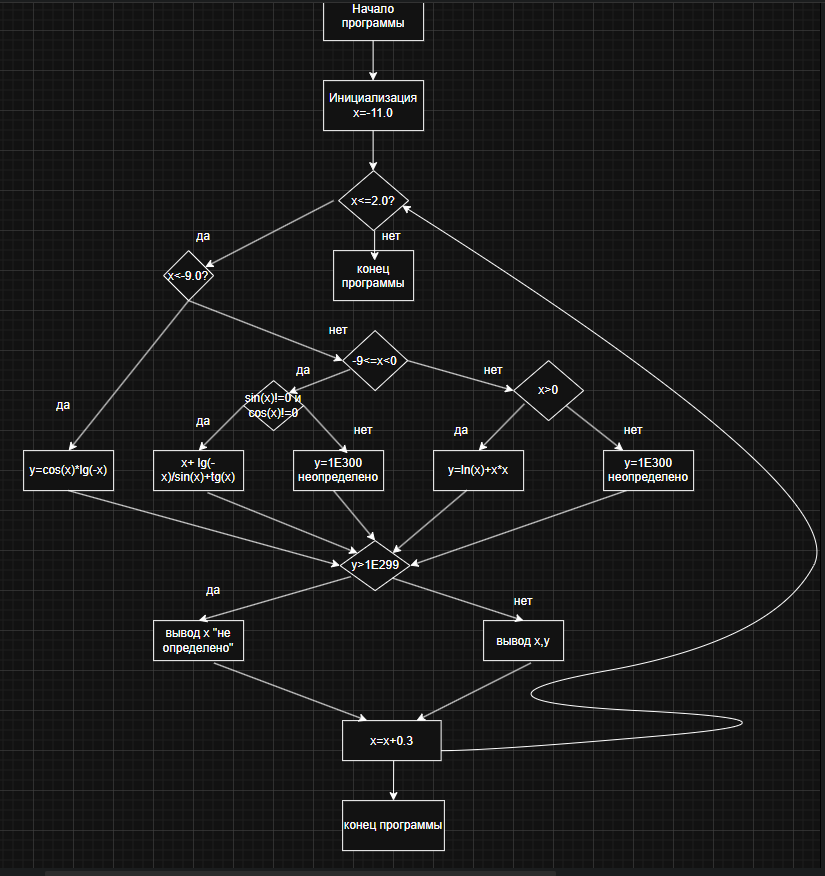
* **Цикл с предусловием** — while x <= 2.0 do для перебора значений x от -11 до 2 с шагом 0.3

Код программы:

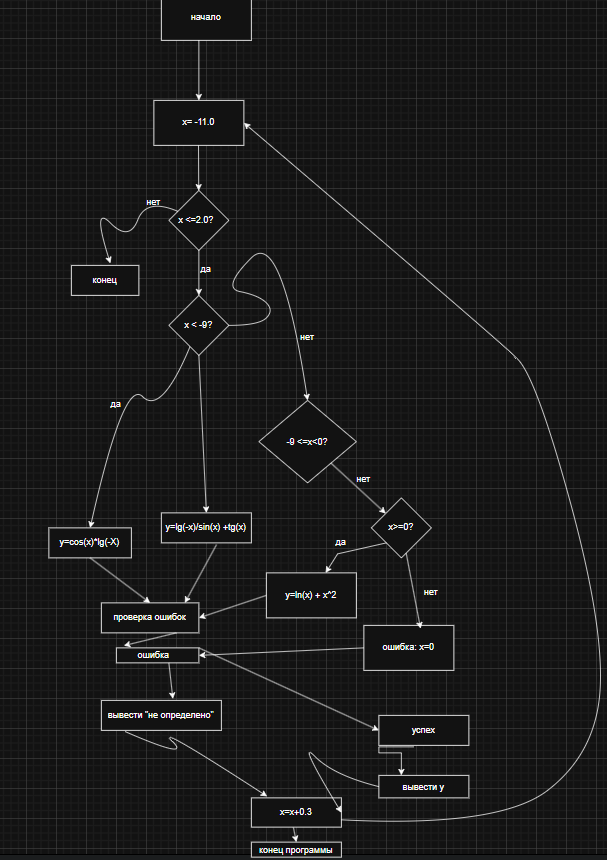
program Lab1;

**Блок схема с пояснениями:**

**Pascal**



**Python**



**Код программы:**

**Pascal:**

uses

Math;

var

x, y: Real;

begin

x := -11.0;

while x <= 2.0 do

begin

if x < -9 then

y := cos(x) \* log10(-x)

else if x < 0 then

begin

if (sin(x) <> 0) and (cos(x) <> 0) then

y := log10(-x) / sin(x) + tan(x)

else

y := 1E300;

end

else if x > 0 then

y := ln(x) + x \* x

else

y := 1E300;

if y > 1E299 then

writeln('x = ', x:5:2, ' y = не определено')

else

writeln('x = ', x:5:2, ' y = ', y:8:4);

x := x + 0.3;

end;

end.

Python

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Контент, сгенерированный ИИ, может содержать ошибки.

**Результат выполнения программы:**

x = -11.00 y = 0.3583

x = -10.70 y = 0.3544

x = -10.40 y = 0.3496

x = -10.10 y = 0.3437

x = -9.80 y = 0.3365

x = -9.50 y = 0.3278

x = -9.20 y = 0.3173

x = -8.90 y = не определено

x = -8.60 y = 2.1125

x = -8.30 y = 1.7363

x = -8.00 y = 1.4941

...

x = -0.30 y = -3.3093

x = 0.00 y = не определено

x = 0.30 y = -1.9186

x = 0.60 y = -0.6321

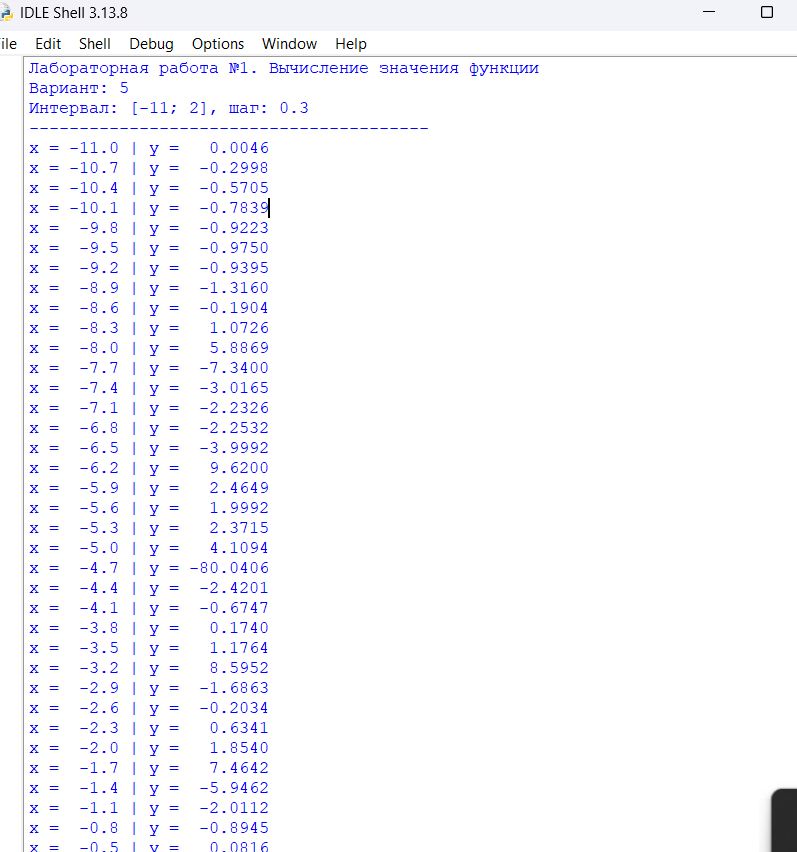
x = 0.90 y = 0.6955

x = 1.20 y = 2.1783

x = 1.50 y = 3.7081

x = 1.80 y = 5.2849

Python



**Вывод**

В ходе лабораторной работы была изучена базовая структура программы на Pascal. Реализована программа, вычисляющая значение кусочно-заданной функции на интервале [-11; 2] с шагом 0,3. Программа корректно обрабатывает особые случаи (деление на ноль, логарифм от неположительного числа) и выводит соответствующие сообщения. Освоены основные конструкции языка: условные операторы, циклы, математические функции.