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**

**«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

203-52-00

ИСПк-\_\_\_\_\_\_\_\_\_

Куншин Кирилл Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

Цель работы: изучить базовую структуру организации программы и основные конструкции языка программирования Pascal.

Задание 1. Вариант 9

1. Написать программу, вычисляющую значение функции:

In(x) / -х, если х < - 9;

sin(x) \* In(x) + x^ 3 /x^3, если -9 <= x < - 4;

e ^ x \* - x - cos(2 \* x), если -4 <= x \* 2;

x^ 2 / sin(x), если 2 == х.

Описание алгоритма:

1. Начало программы
2. Ввод значения переменной x
3. Последовательная проверка условий:

* Если x < -9: вычислить y = ln|x| / (-x)
* Иначе если -9 ≤ x < -4: вычислить y = sin(x) × ln|x| + 1
* Иначе если -4 ≤ x < 2: вычислить y = eˣ × (-x) - cos(2x)
* Иначе если x = 2: вычислить y = x² / sin(x)
* Иначе: вывести сообщение об ошибке и завершить программу

1. Вывод результата вычислений
2. Конец программы

Ответы на вопросы:

1. Алгоритм - это точная, конечная последовательность шагов (инструкций), описывающая процесс решения задачи за конечное число действий.

2. Виды алгоритмов:

* Линейные - последовательное выполнение операторов
* Условные - выполнение разных действий в зависимости от условий
* Циклические - многократное повторение действий
* Рекурсивные - вызов алгоритмом самого себя
* Комбинированные - сочетание различных видов

3. Определение условного алгоритма

Условный алгоритм - это алгоритм, в котором выполнение тех или иных действий зависит от выполнения определенных условий. Он позволяет реализовать ветвление процесса решения задачи.

4. Виды условных алгоритмов и использованные в задаче:

Виды:

* Простая условная конструкция (if-then)
* Полная условная конструкция (if-then-else)
* Множественное ветвление (if-then-else if-...-else)
* Выбор (case-of)

В данной задаче использовано: Множественное ветвление с помощью конструкции if-else if-else if-else

5. Определение циклического алгоритма

Циклический алгоритм - это алгоритм, в котором некоторая последовательность действий выполняется многократно до выполнения определенного условия. Циклы используются для обработки множества данных или многократного выполнения одинаковых операций.

6. Виды циклических алгоритмов и использованные в задаче:

* Виды:
* Цикл с предусловием (while-do) - проверка условия перед выполнением тела цикла
* Цикл с постусловием (repeat-until) - проверка условия после выполнения тела цикла
* Цикл с параметром (for-to/downto-do) - выполнение фиксированного числа раз
* Итерационный цикл - пока не достигнута требуемая точность

В данной задаче циклические алгоритмы не использовались.

Python

import math

x = float(input('Введите значение x: '))

if x < -9:

y = math.log(abs(x)) / -x

elif -9 <= x < -4:

y = math.sin(x) \* math.log(abs(x)) + (x\*\*3 / x\*\*3)

elif -4 <= x < 2:

y = math.exp(x) \* -x - math.cos(2 \* x)

elif x == 2:

y = x\*\*2 / math.sin(x)

else:

print('Введенное значение x не попадает в область определения функции.')

exit()

print(f'Значение функции y при x = {x:.2f} равно {y:.4f}')

input()

Pascal

**uses** Math;

**var** x, y: real;

**begin**

writeln('Введите значение x ');

readln(x);

**if** x < -9

**then**

y := Ln(Abs(x)) / -x

**else if** (x >= -9) **and** (x < -4)

**then**

y := Sin(x) \* Ln(Abs(x)) + (Power(x, 3) / Power(x, 3))

**else if** (x >= -4) **and** (x < 2)

**then**

y := Exp(x) \* -x - Cos(2 \* x)

**else if** x = 2

**then**

y := Power(x, 2) / Sin(x)

**else**

**begin**

writeln('Введенное значение x не попадает в область определения функции.');

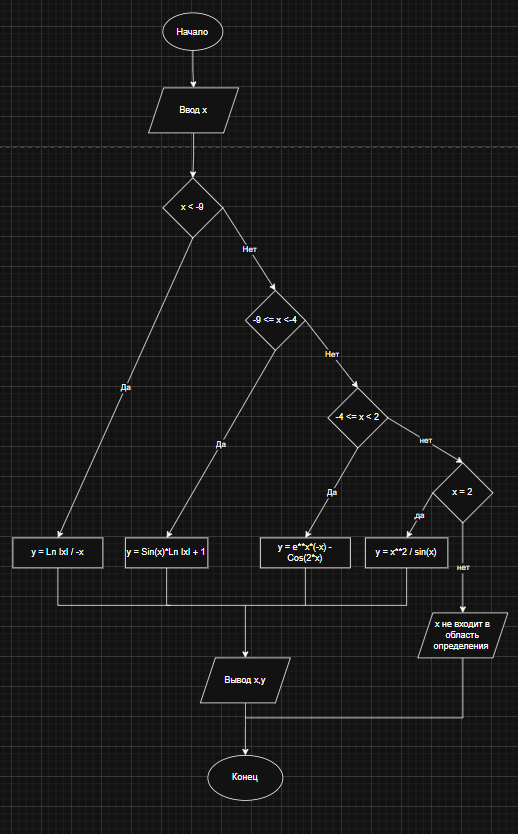
halt;

**end**;

writeln('Значение функции y при x = ', x:0:2, ' равно ', y:0:4);

readln;

**end**.



Задание 2. Вариант 9

Вычислить значение функции на интервале [-11; 4] с шагом 0,1.

Описание алгоритма:

1. Начало программы
2. Задаем значение переменной x: = -11
3. Цикл While-do выполняется пока x <= 4
4. Проверка условий

* Если x < -9: y = ln(-x) / -x
* Если -9 ≤ x < -4: y = sin(x) \* ln(-x) + 1
* Если -4 ≤ x < 2: y = eˣ \* (-x) - cos(2x)
* Если x = 2: y = x² / sin(x)
* Если x > 2: y = 0

1. Вывод x
2. Увеличение x на 0,1
3. Конец цикла
4. Конец программы
5. В данной задаче был использован циклический алгоритм While-do

Python

import math

x = -11.0

while x <= 4:

if x < -9:

y = math.log(-x) / -x

elif x < -4:

y = math.sin(x) \* math.log(-x) + 1

elif x < 2:

y = math.exp(x) \* -x - math.cos(2 \* x)

elif x == 2:

y = x \* x / math.sin(x)

else:

y = 0

print(f'x = {x:6.2f} y = {y:8.4f}')

x += 0.1

Pascal

**var** x, y: real;

**begin**

x := -11;

**while** x <= 4 **do**

**begin**

**if** x < -9

**then**

y := ln(-x) / -x

**else if** x < -4

**then**

y := sin(x) \* ln(-x) + 1

**else if** x < 2

**then**

y := exp(x) \* -x - cos(2 \* x)

**else if** x = 2

**then**

y := x \* x / sin(x)

**else**

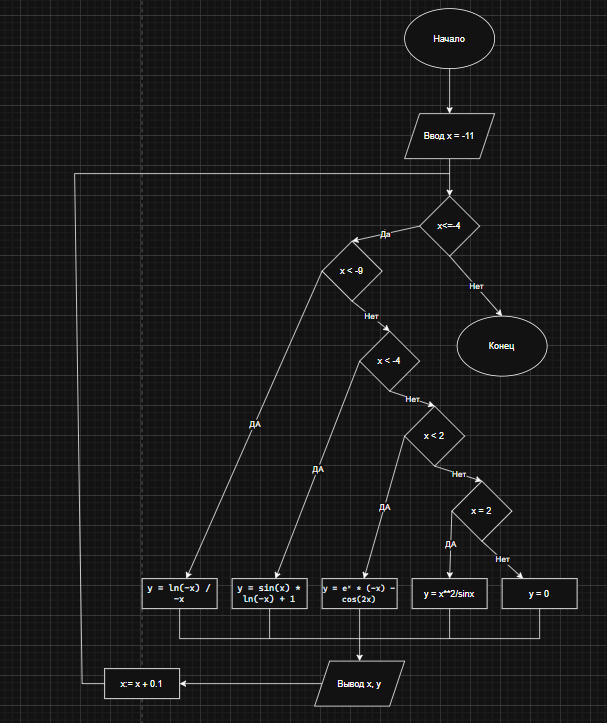
y := 0;

writeln('x = ', x:6:2, ' y = ', y:8:4);

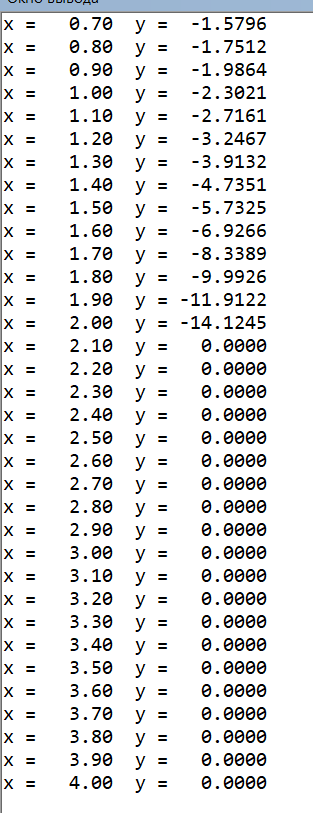
x := x + 0.1;

**end**;

**end**.

****

**Результат выполнения:**

****

**Вывод**

В ходе лабораторной работы была изучена базовая структура программы на Pascal. 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.