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 программированния»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Сибейкин Иван Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Цель работы

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

2.Формулировка задания (с вариантом)

Вариант 15

Написать программу, вычисляющую значение функции.

72/x^ (1/3)- x^ (0,1 \* x) / x. если x <-9;

X ^ (1/3) / cos(x) + e ^x, если -9 <=x<0;

90- cos(x), если 0 <=x.

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

3.Описание алгоритма и ответы на вопросы

Подключите модуль Math для доступа к математическим функциям, таким как Power для возведения в степень, Cos для косинуса и Ln для натурального логарифма.

Определение функции f(x), которая вычисляет значение функции в зависимости от значения x:

- Если x < -9, вычисляет 72/x^ (1/3)- x^ (0,1 \* x) / x

- Если -9 <= x < 0, вычисляет X ^ (1/3) / cos(x) + e ^x

- Если 0 <= x, вычисляет 90- cos(x)

Объявление переменной x :

Основной блок программы:

Присвойте переменной x начальное значение -11.

Используйте цикл while, чтобы вычислить значение функции для всех значений x от -11 до 2 с шагом 0.3.

Вычислите значение функции f(x) для текущего значения x, используя определённые условия для диапазонов значений.

Выведите на экран значение x и соответствующее значение функции f(x)f(x)f(x), используя Format для форматированного вывода.

Увеличьте значение x на 0.3 для перехода к следующему значению

Таким образом, программа вычисляет значение функции на заданном интервале [-11; 4] с шагом 0.3 и выводит результаты на экран.

Ответы на вопросы:

1.Алгоритм — это четкая и понятная последовательность действий, которая решает определенную задачу. Он описывает, что нужно делать, в каком порядке и как именно.

2. Виды алгоритмов:

- Линейные алгоритмы - выполняются строго последовательно, без ветвлений и циклов.

- Условные алгоритмы - содержат ветвления, где выбор действия зависит от выполнения определенного условия.

- Циклические алгоритмы - включают повторяющиеся действия, выполняемые определенное количество раз или пока не будет выполнено некоторое условие.

3. Условный алгоритм — это алгоритм, в котором есть ветвления, то есть выбор действия зависит от выполнения определенного условия.

4. Виды условных алгоритмов:

Ветвление — это выбор между двумя вариантами действий.

Множественное ветвление switch-case - выбор между несколькими вариантами действий.

5. Циклический алгоритм — это алгоритм, который содержит повторяющиеся действия, выполняемые определенное количество раз или пока не будет выполнено некоторое условие.

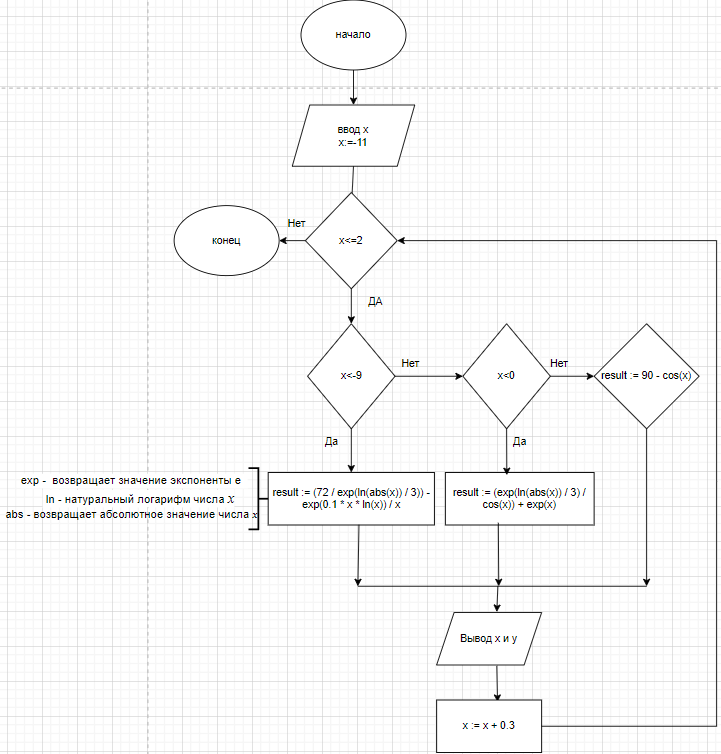
6. Виды циклических алгоритмов:

- Цикл for - выполняется фиксированное количество раз.

- Цикл while - выполняется, пока условие истинно.

- Цикл do-while - выполняется как минимум один раз, затем пока условие истинно.

Схема алгоритма с комментариями



Resuit :=90-cosx

Код для программы в Pascal.

**program** dmvirme;

**var**

x, result, cubeRoot: Real;

**begin**

x := -11;

**while** x <= 2 **do**

**begin**

**if** x < -9 **then**

result := (72 / exp(ln(abs(x)) / 3)) - exp(0.1 \* x \* ln(x)) / x

**else if** x < 0 **then**

result := (exp(ln(abs(x)) / 3) / cos(x)) + exp(x)

**else**

result := 90 - cos(x);

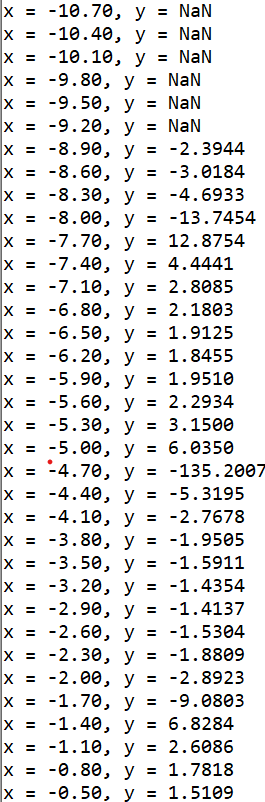
writeln('x = ', x:0:2, ', y = ', result:0:4);

x := x + 0.3;

**end**;

**end**.

6.Результат выполнения программы



Вывод:

Изучение языка программирования Pascal позволило глубже понять принципы построения программ и эффективного применения основных языковых конструкций для решения различных задач. В ходе освоения Pascal были усвоены не только особенности синтаксиса, но и ключевые принципы программирования, что важно для развития логического мышления.

Также в процессе работы была изучена методология создания алгоритмических схем, что способствует упорядоченной организации решений и помогает лучше понять взаимодействие компонентов программы. Владение алгоритмическими методами стало основой для будущих программных разработок и создания более сложных решений.

Таким образом, поставленные цели были успешно выполнены, и приобретенные навыки будут активно применяться в последующих проектах. Базовое понимание Pascal укрепило уверенность в собственных силах и повысило мотивацию для дальнейшего изучения информатики и программирования.