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**

**«**Линейный алгоритм. Ветвление»

**ПО ДИСЦИПЛИНЕ**

**«**Основы алгоритмизации и программирования**»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Ильин Тимофей Анатольевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

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

**Задание 1 (Вариант 6)**

Написать программу, вычисляющую значение функции:

56 + x^(0,1 \* x), если х < -8;

x^2 + -x, если -8 <= x < 0;

38 / x^3 + tg(x), если 0 <= x.

**Описание алгоритма задания 1**

1. Ввод значение х;
2. Если х меньше -8, то к переменной у присваивается значение выражения 56 + x^(0,1 \* x), в противном случае переходим к следующему условию;
3. Если х больше или равен -8 и меньше 0, то к переменной у присваивается значение выражения x^2 + -x, в противном случае переходим к следующему условию;
4. В остальных случаях, то есть когда х больше или равен 0, к переменной у присваивается значение выражения 38 / x^3 + tg(x);
5. Проверка условием y=y переменной у на наличие невозможных значений: если условие верно то выводим y, если нет то выводим “нет решений”.

**Схема алгоритма с комментариями задания 1**

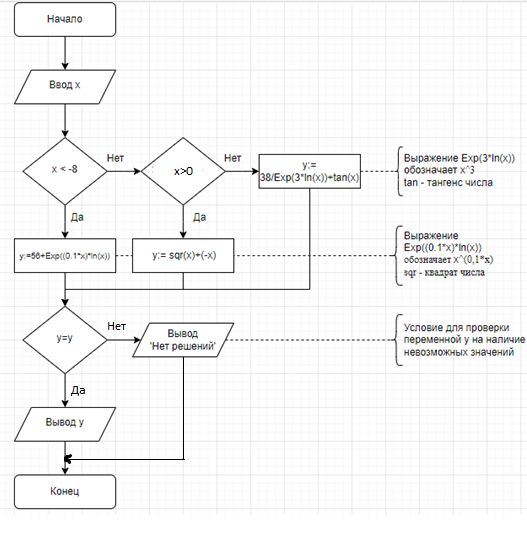


Рис.1 Схема алгоритма первой задачи

**Код программы задания 1**

**program** zad1;

**var**

x, y: real;

a: integer;

**begin**

write('Введите x, x = ');

readln(x);

**if** x < -8 **then**

y := 56 + Exp((0.1 \* x) \* ln(x))

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

y := sqr(x) + (-x)

**else**

y := 38 / Exp(3 \* ln(x)) + tan(x);

**if** y = y **then** a := 1

**else** a := 0;

**if** a = 1 **then**

writeln('x = ', x:0:1, ' y = ', y:0:2)

**else** writeln('x = ', x, ' Нет решений');

**end**.

**Результат выполнения программы задания 1**

1. Результат программы при условии х < -8. Вводим -9.

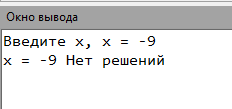


Рис. 2 Результат программы (х < -8)

1. Результат программы при условии -8 <= x < 0. Вводим -8.

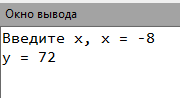


Рис. 3 Результат программы (-8 <= x < 0)

1. Результат программы при условии 0 <= x. Вводим 1.

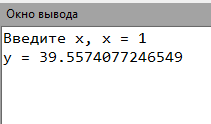


Рис. 4 Результат программы (0 <= x)

**Задание 2 (Вариант 6)**

Вычислить значение функции (задание 1) на интервале [-10;2] с шагом 0,2

**Описание алгоритма задания 2**

1. Ввод значение х;
2. Присвоение к переменной х значения -10, для того чтобы интервал, вычисляемых значений функции, начинался с -10
3. Ввод цикла c предусловием, который будет выполняет действия тогда и только тогда, когда условие х <= 2 верно.
4. Если х меньше -8, то к переменной у присваивается значение выражения 56 + x^(0,1 \* x), в противном случае переходим к следующему условию;
5. Если х больше или равен -8 и меньше 0, то к переменной у присваивается значение выражения x^2 + -x, в противном случае переходим к следующему условию;
6. В остальных случаях, то есть когда х больше или равен 0, к переменной у присваивается значение выражения 38 / x^3 + tg(x);
7. Проверка условием y=y переменной у на наличие невозможных значений: если условие верно то выводим y, если нет то выводим “нет решений”.
8. Присвоение к переменной х выражения х+0,2, для того чтобы задать шаг интервала, вычисляемых на нем значений функции.

**Схема алгоритма с комментариями задания 2**

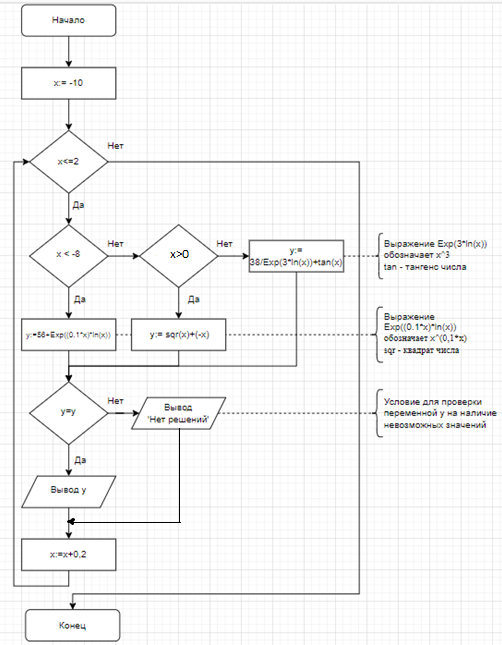


Рис. 5 Схема алгоритма задачи 2

**Код программы задания 2**

**program** zad2;

**var**

x, y, a: real;

**begin**

x := -10;

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

**begin**

**if** x < -8 **then**

y := 56 + Exp((0.1 \* x) \* ln(x))

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

y := sqr(x) + (-x)

**else**

y := 38 / Exp(3 \* ln(x)) + tan(x);

**if** y = y **then** a := 1

**else** a := 0;

**if** a = 1 **then**

writeln('x = ', x:0:1, ' y = ', y:0:2)

**else** writeln('x = ', x:0:1, ' Нет решений');

x := x + 0.2;

**end**;

**end**.

**Результат выполнения программы задания 2**

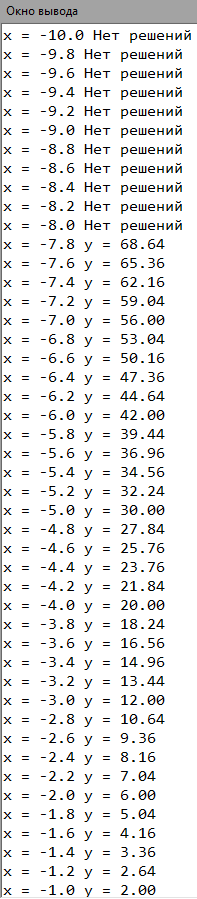


Рис. 6 Результат программы

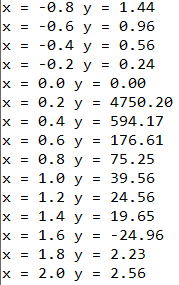


Рис. 7 Результат программы (продолжение)

Вывод: в результате выполнения домашней контрольной работы было изучено базовая структура организации программы и основные конструкции языка программирования Pascal.

Так же столкнулся с такой проблемой как **NaN** - числовой тип данных, значение которого является неопределенным или непредставимым. Из-за того что степень числа получалась отрицательной программа выводила NaN.

NaN не равен ни одному другому значению (даже самому себе); соответственно, самый простой метод проверки результата на NaN — это сравнение полученной величины с самой собой.

Отсюда мы получаем простое условие y=y, которое будет проверять полученные значения переменных на наличие невозможных ответов.