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

Барбарина Полина Игоревна

Преподаватель:

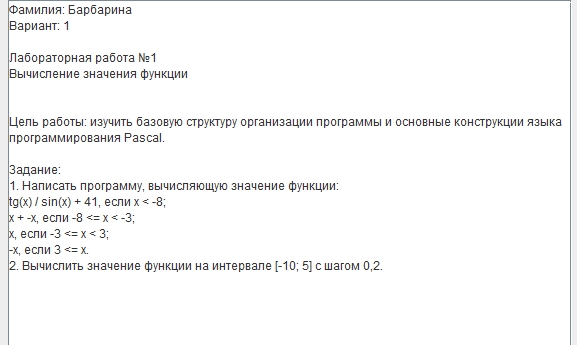
Бычкова Дарья Владимировна

Киров

2025

Ход работы

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



Алгоритм программы для задания №1:

1. Начало программы: объявление переменных: x, y (вещественные числа).
2. Инициализация начального значения: ввести значение x.
3. Вывод заголовка таблицы: создать шапку таблицы значений.
4. Определение функции: сравнить х с пределом допустимого значения:
   * Если x < -8: вычислить y = tg(x) / sin(x) + 41;
   * Иначе если x < -3: вычислить y = x + (-x) = 0;
   * Иначе если x <3: установить y = x;
   * Иначе (x ≥ 3): установить y = -x.
5. Вывод результата: форматированный вывод пары (x, y) в таблицу.
6. Завершение программы: конец программы.

Алгоритм программы для задания №2:

1. Начало программы: объявление переменных: x, y (вещественные числа).
2. Инициализация начального значения: установить x = -10.0 (начало интервала).
3. Вывод заголовка таблицы: создать шапку таблицы значений.
4. Цикл вычислений **(**пока x ≤ 5.0)
   * Если x < -8: вычислить y = tg(x) / sin(x) + 41;
   * Иначе если x < -3: вычислить y = x + (-x) = 0;
   * Иначе если x <3: установить y = x;
   * Иначе (x ≥ 3): установить y = -x.
5. Вывод результата: форматированный вывод пары (x, y) в таблицу.
6. Изменение шага: увеличить x на 0.2 (x := x + 0.2).
7. Повторение цикла: вернуться к шагу 5, пока x не превысит 5.0.
8. Завершение программы: конец программы.

Алгоритм — это точная, конечная последовательность шагов (инструкций), описывающая процесс решения задачи за конечное время.

Виды алгоритмов:

* Линейные;
* Условные (разветвляющиеся);
* Циклические;
* Вспомогательные.

Условный алгоритм (разветвляющиеся)— это алгоритм, в котором выбор следующей команды зависит от истинности или ложности некоторого условия.

Виды условных алгоритмов:

* Полная форма (if-else) — если условие истинно, выполняется одна группа команд, если ложно — другая;
* Неполная форма (if) — команды выполняются только при истинности условия;
* Многозначное ветвление (switch-case) — выбор из нескольких вариантов в зависимости от значения переменной.

В обоих заданиях мы использовали только полную форму условного алгоритма для определения вида функции, на основе сравнения числа с пределом допустимого значения

Циклический алгоритм — это алгоритм, в котором некоторая последовательность команд (тело цикла) выполняется многократно до тех пор, пока выполняется заданное условие.

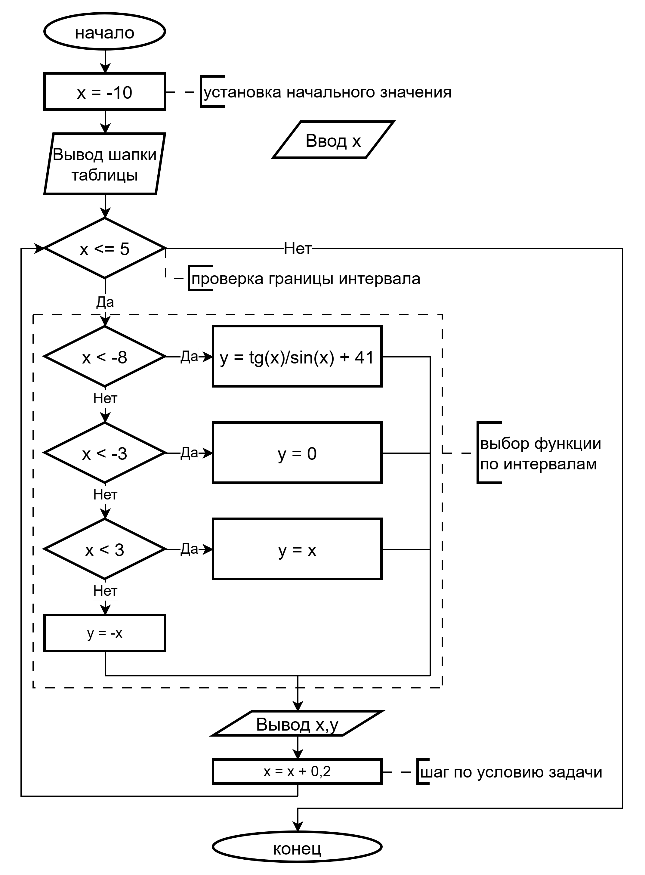
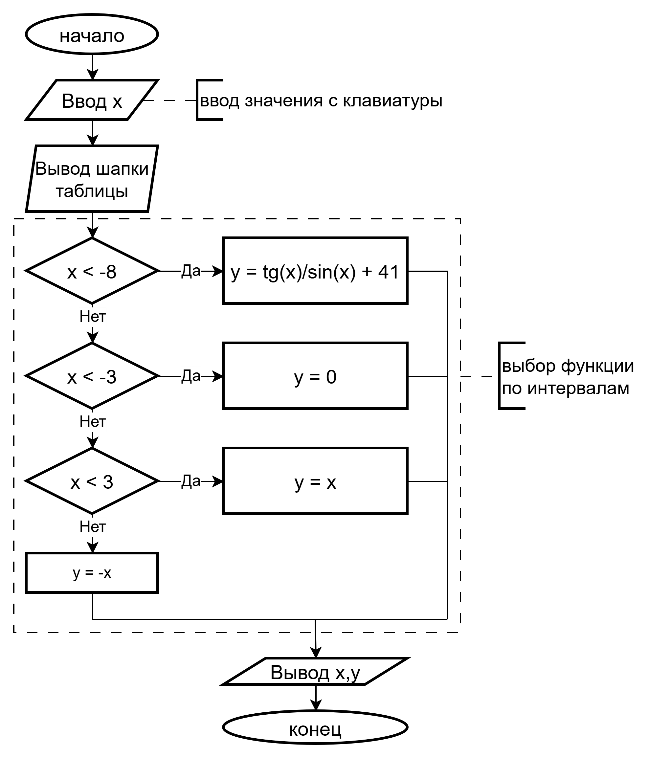
Виды циклических алгоритмов:

* Цикл с предусловием (while-do) — проверка условия до выполнения тела цикла;
* Цикл с постусловием (repeat-until) — проверка условия после выполнения тела цикла;
* Цикл с параметром (for) — выполнение с известным количеством повторений, управляемое счетчиком.

В задании №2 мы использовали только цикл с предусловием для того, чтобы задать вторую границу интервала из второго задания, а в задании №1 и вовсе не использовали циклические алгоритмы

Схемы алгоритмов:

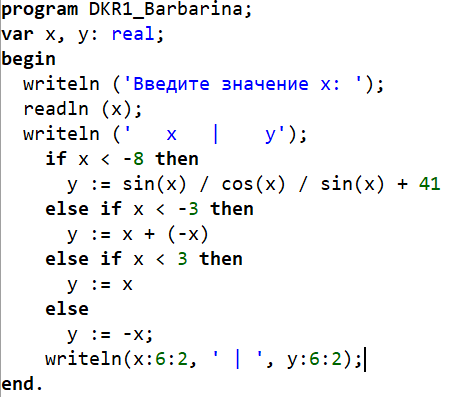
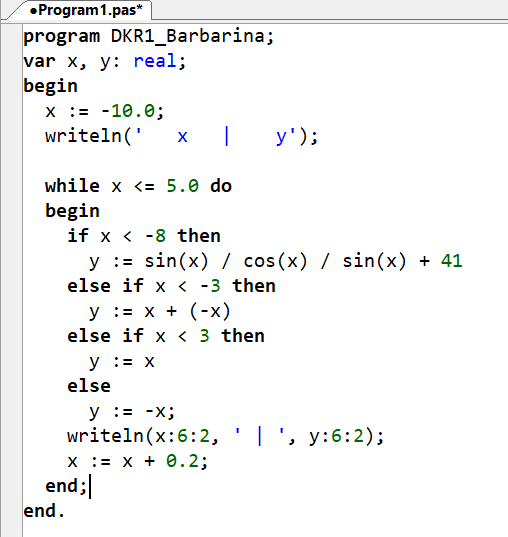
Задание №1 Задание №2



Коды программ:

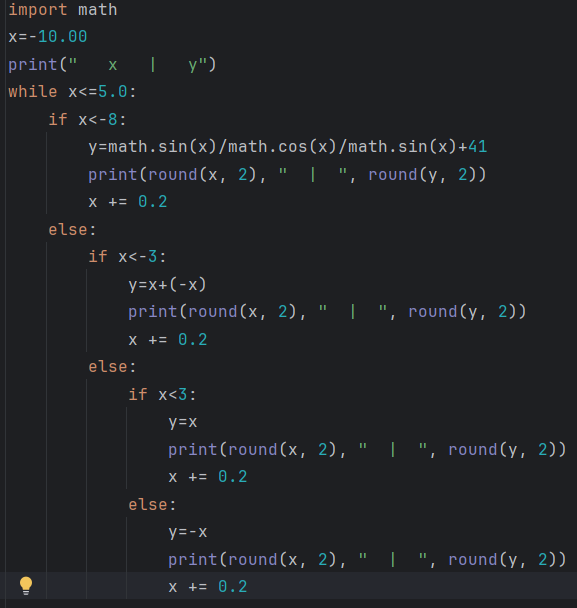
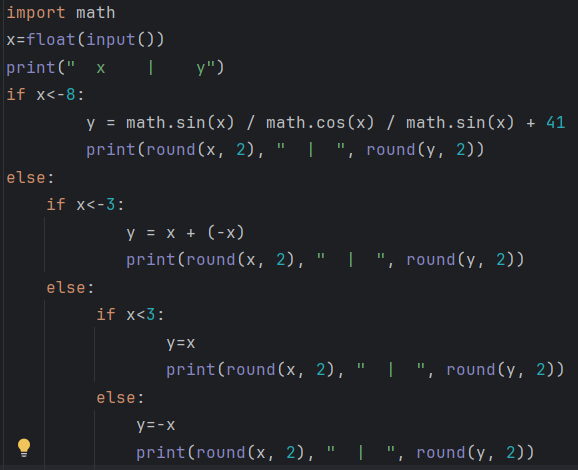
* Pascal:

Задание №1 Задание №2

* Python:

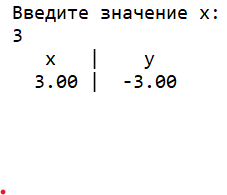
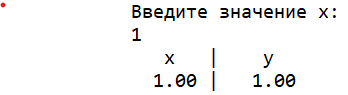
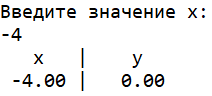
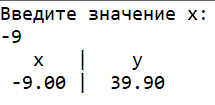
Задание №1 Задание №2

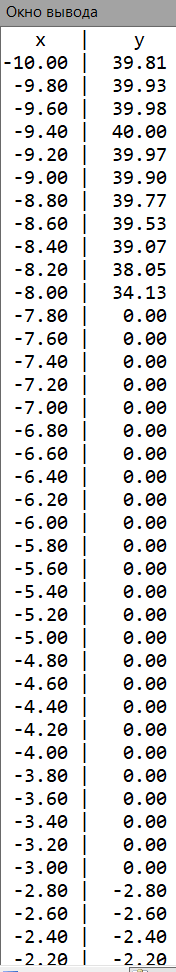
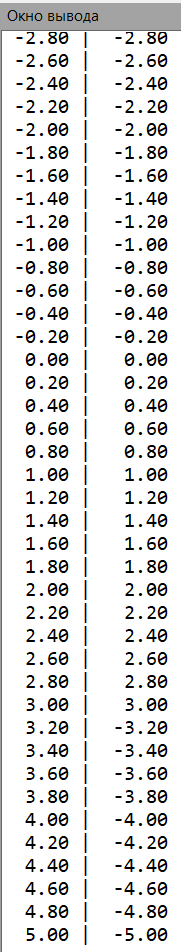


Результат выполнения:

* Pascal:

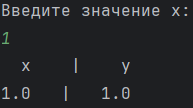
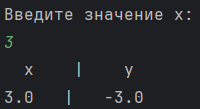
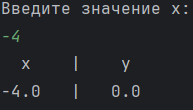
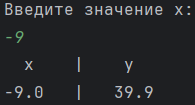
Задание №1 Задание №2

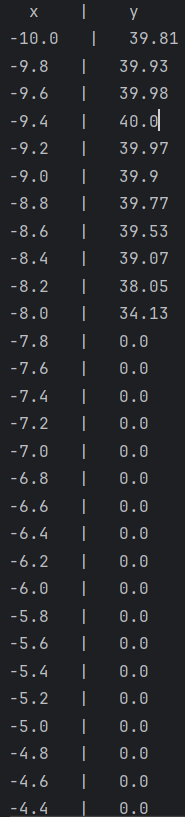
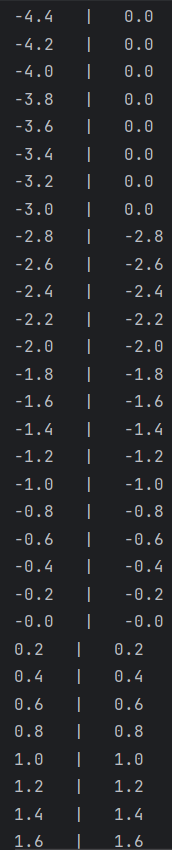
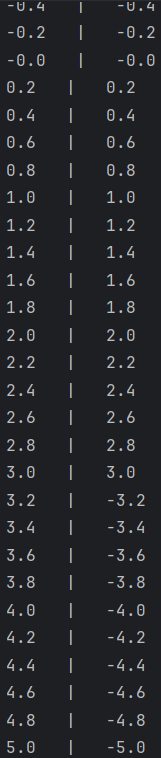


* Python:

Задание №1 Задание №2



Вывод: в ходе выполнения лабораторной работы №1 "Вычисление значения функции" была успешно реализована программа на языке Pascal для вычисления значений функций на интервале [-10; 5] с шагом 0,2. Мы смогли реализовать практическое применение базовых конструкций языка Pascal, включая объявление переменных вещественного типа, организацию циклических вычислений с использованием оператора while, реализацию сложного ветвления с помощью оператора if-else, а также форматированный вывод результатов. Результаты работы программы представлены в удобном табличном формате, что позволяет наглядно проследить поведение функции на всем заданном интервале. В проделанной работе мы подтвердили важность точной реализации алгоритма и тщательной проверки граничных условий при работе с кусочно-заданными функциями. Полученные навыки будут полезны нам для решения более сложных вычислительных задач в дальнейшем изучении программирования.