Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКАЯ РАБОТА №1

ПО ОП.04 «Основы алгоритмизации и программирования»

|  |  |
| --- | --- |
| Выполнил студент  группы П50-3-22  Биктимиров Р.О. | Проверил преподаватель  \_\_\_\_\_\_\_\_\_Т.Д. Артамонова  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 года |

Москва 2023

Цель: Написать код для инженерного калькулятора. Реализовать такие операции как сложение, вычитание, умножение, деление, возведение в степень, квадратный корень, факториал, синус, косинус, тангенс.

Ход работы:

1. Первым делом создаем переменную work, которая будет хранить булево значение, которое будет показывать статус работы калькулятора. Создаем цикл while с проверкой переменной work, пока work = True, калькулятор работает

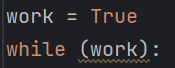


Рисунок 1 – work и while

1. После переходим в цикл while и с помощью функции print выводим все возможные операции и запрашиваем у пользователя номер операции с помощью функции input, которую помещаем в переменную operation

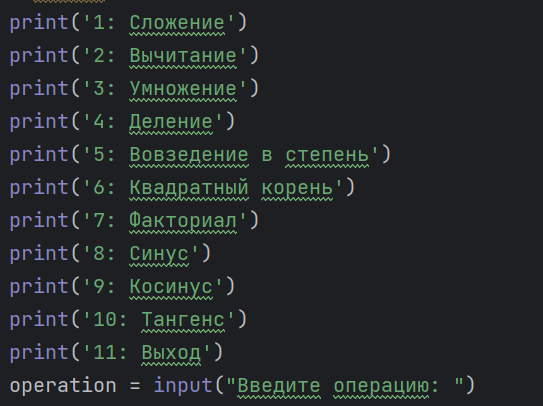


Рисунок 2 – print и operation

1. Для фильтрации введенной пользователем операции используем конструкцию match case с проверкой значений переменной operation. Каждой определенной цифре соответствует своя математическая операция.
2. Сначала рассмотрим первые 4 операции: сложение, вычитание, умножение, деление.

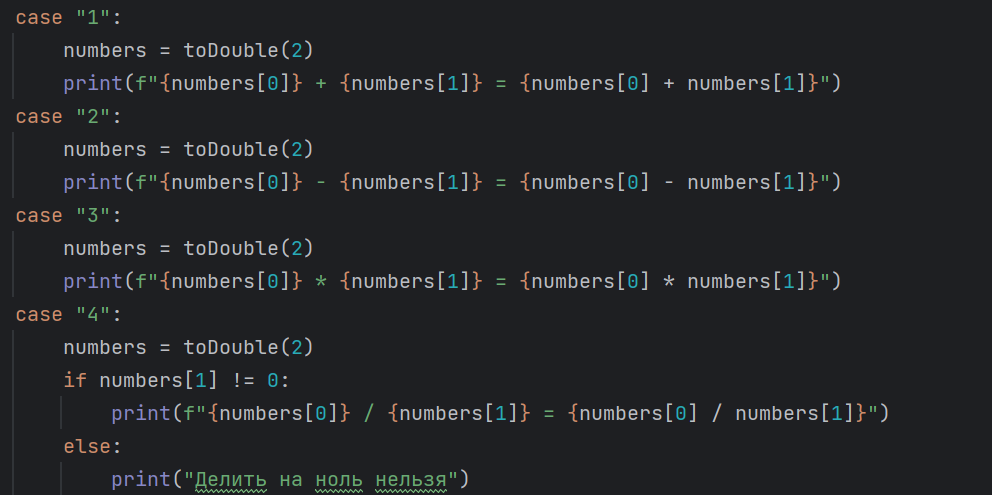


Рисунок 3 – первые 4 операции

1. В каждом case в первой строчки идет инициализация переменной numbers, которая вызывает функцию toDouble, с аргументом 2, вторая строчка с помощью функции print и интерполяции выводит пользователю ответ в соответствии с номером операции и введенных им чисел, которые получаем numbers[0] и numbers[1]. Также в четвертой операции – деления, мы проверяем что пользователь ввел второе число не равное нулю, потому что на ноль делить нельзя, если же всё-таки такое произошло выводим уведомлению о том что такая операция невозможна.
2. Функция toDouble была создана для запроса у пользователя нужного количества чисел, в примере выше 2. Функция принимает 1 параметр countNumbers – количество чисел, которое должен ввести пользователь. Внутри функции создается переменная numbers, которая является списком, также инициализируется вспомогательная переменная b изначально равная нулю. Дальше создается цикл while с условием работы пока вспомогательная переменная b не равна параметру функции countNumbers. Внутри цикла с помощью конструкции try, except реализован запрос пользователю на ввод числа и проверка введенного числа, если пользователь ввел число, то к вспомогательной переменной b прибавляется 1 и в список numbers добавляется введенное пользователем число, если же введенное значение не число с помощью блока except вызывается функции print, которая оповещает пользователя о том, что ему необходимо ввести именно число. По завершению цикла с помощью return функция возвращает список введенных пользователем чисел.

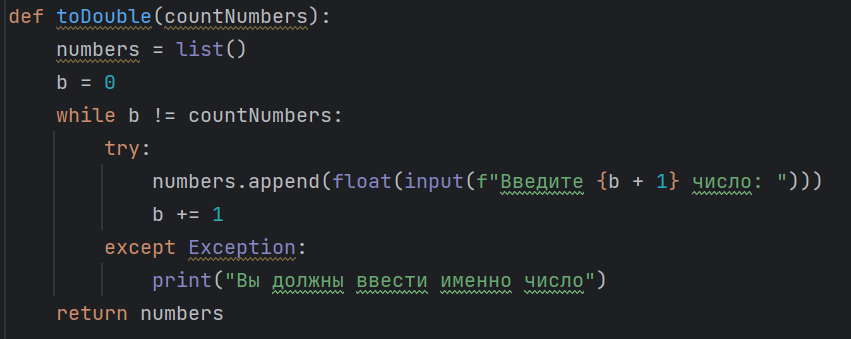


Рисунок 4 – функция toDouble

1. Дальше пройдемся от 5 по 10 операции: возведение в степень, квадратный корень, факториал, синус, косинус, тангенс.

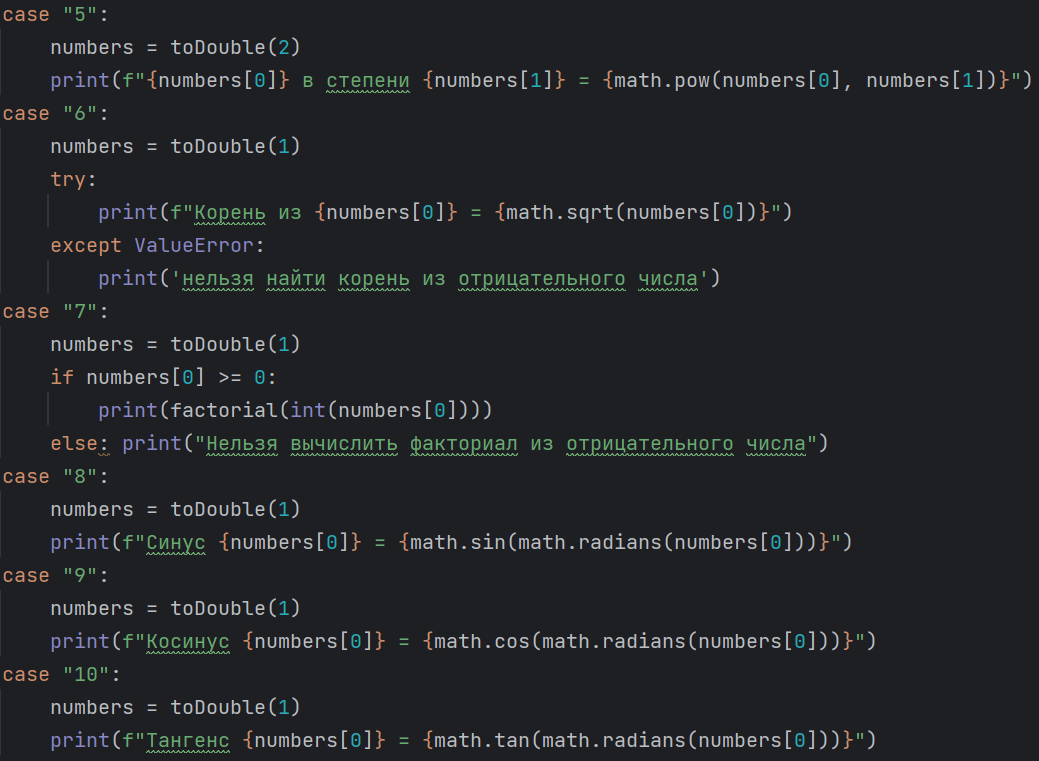


Рисунок 5 –с 5 по 10 операции

1. В операциях 5, 6, 8, 9, 10 используется библиотека math, импорт которой мы произвели в самой первой строчке кода



Рисунок 6 – импорт библиотеки math

1. Её функции соответственно: 5 - операция возведения в степень – функция pow, 6 - корень из числа – sqrt, 8 – вычисление синуса – sin, для расчета введенного числа в градусах также используется функция radians в 9 и 10 операциях, 9 – вычисление косинуса – cos, 10 – вычисление тангенса – tan. В 5 операции пользователь должен ввести два числа для подсчета поэтому аргументом функции toDouble будет 2, во всех следующих операциях аргумент этой функции равен 1.
2. В 6 операции с помощью конструкции try, except пытаемся вычислить квадратный корень из числа, введенного пользователем, если у нас это получается, то выводится ответ, если выпадает ошибка, то уведомляем пользователя о том, что найти корень из отрицательного числа невозможно.

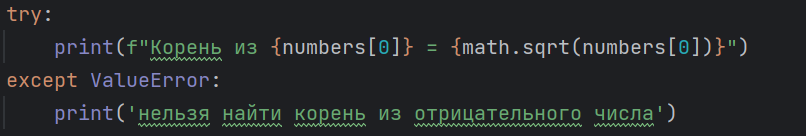


Рисунок 7 – пытаемся вычислить квадратный корень

1. В операции под номером 7 – нахождение факториала числа, сначала мы проверяем что пользователь ввел число большее или равное 0, если же это не так, то выводим уведомление о невозможности нахождения факториала от отрицательного числа, если же проверка проходит успешно, то вызываем функцию factorial, в аргумент которой передаем число, введенное пользователем, преобразуя его в тип данных int с помощью функции int.

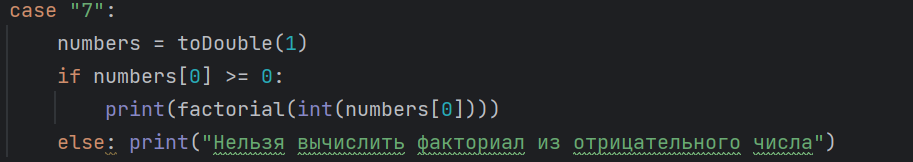


Рисунок 8 – операция 7 – вычисление факториала числа

1. Функция factorial принимает на вход число, для которого необходимо посчитать факториал. Внутри данной функции идет проверка на число, введенное как параметр, если же число – параметр равно 0, то функция возвращает 1, если же это не так, то функция возвращает число – параметр умноженное на вызов этой же функции только аргумент, которой уменьшен на 1.

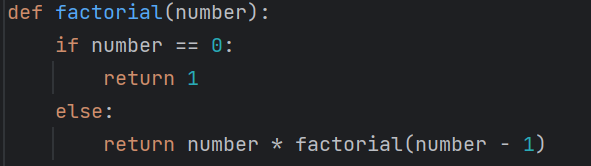


Рисунок 9 – функция factorial

1. Дальше рассмотрим выход из калькулятора и обработчик неизвестной команды.

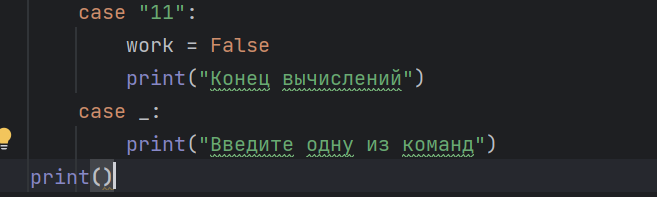


Рисунок 10 – выход из калькулятора и обработка неизвестной команды

1. Если пользователь вводит операцию 11, то вычисления заканчиваются и переменной work присваивается значение False, что означает что цикл while прекращает свою работу, также выводится уведомление о завершении вычислений.
2. Если же пользователь нечаянно ошибется в вводе команд или введет неизвестную команду, то выведется сообщение о необходимости ввести именно одну из команд.
3. После конструкции match case есть функция print без аргумента, которая служит для красоты вывода ответов и уведомлений.

Вывод: В ходе практической работы был реализован код для инженерного калькулятора с операциями: сложение, вычитание, умножение, деление, возведение в степень, квадратный корень, факториал, синус, косинус, тангенс.