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 «ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПО»

Листов: 3

|  |  |
| --- | --- |
| Выполнил студент  Группы: П50-2-20  Нефедов Павел Вадиммович | Проверил преподаватель  М. Н. Гацкан  «\_\_\_\_» \_\_\_\_\_\_\_2023 года |

Москва 2023

Цель работы: Сделать калькулятор с функциями (+,-./,\*,возведение в степень) с вводом количества операций.

|  |
| --- |
| *def* find\_str(*s*, *char*):      index = 0      if *char* in *s*:          c = *char*[0]          for ch in *s*:              if ch == c:                  if *s*[index:index+len(*char*)] == *char*:                      return index              index += 1      return -1  isGoing = True  result = None  while isGoing :      if result == None:          mathString = input("Введите простое выражение (пример: 1+1)\n")          if mathString != "":              if find\_str(mathString, "-") >= 0:                  first = float(mathString[:find\_str(mathString, "-")])                  second = float(mathString[find\_str(mathString, "-")+1:])                  print(*f*"{first} - {second} = {first - second}")                  result = first - second              elif find\_str(mathString, "+") >= 0:                  first = float(mathString[:find\_str(mathString, "+")])                  second = float(mathString[find\_str(mathString, "+")+1:])                  print(*f*"{first} + {second} = {first + second}")                  result = first + second              elif find\_str(mathString, "\*\*") >= 0:                  first = float(mathString[:find\_str(mathString, "\*\*")])                  second = float(mathString[find\_str(mathString, "\*\*")+2:])                  print(*f*"{first} ^ {second} = {first \*\* second}")                  result = first \*\* second              elif find\_str(mathString, "\*") >= 0:                  first = float(mathString[:find\_str(mathString, "\*")])                  second = float(mathString[find\_str(mathString, "\*")+1:])                  print(*f*"{first} \* {second} = {first \* second}")                  result = first \* second              elif find\_str(mathString, "/") >= 0:                  first = float(mathString[:find\_str(mathString, "/")])                  second = float(mathString[find\_str(mathString, "/")+1:])                  if(second != 0):                      print(*f*"{first} / {second} = {first / second}")                      result = first / second                  else:                      print("Error")          else:              isGoing = False      else:          mathString = input(*f*"Введите простое выражение (пример: 1+1)\n {result}")          if find\_str(mathString, "-") >= 0:              second = float(mathString[find\_str(mathString, "-")+1:])              print(*f*"{result} - {second} = {result - second}")              result = result - second          elif find\_str(mathString, "+") >= 0:              second = float(mathString[find\_str(mathString, "+")+1:])              print(*f*"{result} + {second} = {result + second}")              result = result + second          elif find\_str(mathString, "\*\*") >= 0:              second = float(mathString[find\_str(mathString, "\*\*")+2:])              print(*f*"{result} ^ {second} = {result \*\* second}")              result = result \*\* second          elif find\_str(mathString, "\*") >= 0:              second = float(mathString[find\_str(mathString, "\*")+1:])              print(*f*"{result} \* {second} = {result \* second}")              result = result \* second          elif find\_str(mathString, "/") >= 0:              second = float(mathString[find\_str(mathString, "/")+1:])              if(second != 0):                  print(*f*"{result} / {second} = {result / second}")                  result = result / second              else:                  print("Error")          else:              isGoing = False |

Таблица – Код программы

Метод find\_str ищет определённую подстроку в строке и возвращает индекс её начала. Это используется в программе для поиска знака действия.

В основной программе происходит следующее:

1. Начинается цикл и проверяется, есть ли результат предыдущего действия.
2. Если его нет, то пользователь вводит целое выражение, иначе только вторую часть.
3. Программа проверяет знак действия между числами и проводит вычисления.
4. Пользователь может закончить программу, если оставит строку пустой.

Результат работы:



Рисунок 1 – Минус

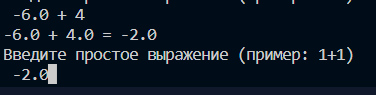


Рисунок 2 – Плюс

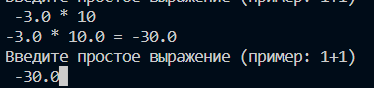


Рисунок 3 – Умножение

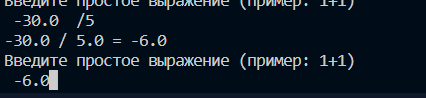


Рисунок 4 – Деление

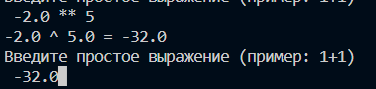


Рисунок 5 – Степень

Как вы можете видеть, результат вычислений сохраняется между процессами (плюс был сделан после всех других).

Вывод: в ходе практической работы был разработан калькулятор с базовыми функциями вычисления.