**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки(специальность) **09.03.03 Прикладная информатика**

**ПРАКТИЧЕСКАЯ РАБОТА №1**

**По дисциплине «Программирование»**

**Тема: Калькулятор**.

**Выполнил** Проскуряков Р. В. K3139

**Проверил** Терещенко В. В.

**Дата** 11.10.2023

**Санкт-Петербург 2023**

**Цель работы**

Познакомиться с разработкой на языке Python, способами тестирования кода с помощью юнит-тестов, с использованием git и githab в качестве системы контроля версий.

**Ход работы**

В качестве тренировки был создан калькулятор, способный вычислять результат математических выражения, состоящих из рациональных чисел, скобок, умножения, деления, сложения, и разности.

1. Привести введённую строчку к требуемому формату, и проверить на отсутствие запрещённых символов.

def change\_format(st):

    s = ''.join(filter(lambda ch: ch not in ' \t', st))

    s = s.replace("--", '+')

    s = s.replace(",", '.')

    tmp0 = [s[0]]

    i = 1

    while(i < len(s) - 1):

        if(s[i] == '-' and (is\_num(s[i - 1]) or s[i - 1] == ')')):

            tmp0.append('+')

            tmp0.append('-')

        else:

            tmp0.append(s[i])

        i += 1

    tmp0.append(s[-1])

    tmp1 = []

    for i in range(len(tmp0) - 1):

        tmp1.append(tmp0[i])

        if(tmp0[i + 1] == '(' and is\_num(tmp0[i]) and tmp1[i] != '-'):

            tmp1.append('\*')

    tmp1.append(tmp0[-1])

    tmp2 = [tmp1[0]]

    for i in range(1, len(tmp1)):

        if(tmp1[i - 1] == ')' and is\_num(tmp1[i])):

            tmp2.append('\*')

        tmp2.append(tmp1[i])

    tmp3 = []

    for i in range(len(tmp1) - 1):

        if(tmp2[i] != '+' or tmp1[i + 1] != '+'):

            tmp3.append(tmp2[i])

    tmp3.append(tmp2[-1])

    return ''.join(tmp3)

1. Выполнить вычисления с помощью разделения выражения на слагаемые, которые в свою очередь на множители и делители и не забываем про скобки.

from collections import deque

def is\_num(ch):

    return ord('0') <= ord(ch) <= ord('9') or ch == '.' or ch == '-'

def if\_normal(ch):

    return is\_num(ch) or ch in "+/\*()"

def calculate(st):

    q = deque()

    i = 0

    while(i < len(st)):

        if(st[i] == '('):

            q.append(i)

        elif (st[i] == ')'):

            if(len(q) > 0):

                li = q.pop()

                old\_len = len(st)

                t = str(plus\_minus(st[li + 1:i]))

                if(st[li - 1] == '-' and t[0] == '-'):

                    st = st[:li - 1] + t[1:] + st[i + 1:]

                else:

                    st = st[:li] + t + st[i + 1:]

                i -= old\_len - len(st)

            else:

                raise ArithmeticError('Неверная скобочная последовательность!')

        i += 1

    if(len(q) > 0):

        raise ArithmeticError('Неверная скобочная последовательность!')

    return plus\_minus(st)

def plus\_minus(st):

    li = 0

    while(li < len(st) and st[li] != '+'):

        li += 1

    a = mult\_div(st[:li])

    i = li + 1

    while(li < len(st)):

        while(i < len(st) and st[i] != '+'):

            i += 1

        b = mult\_div(st[li + 1:i])

        a += b

        li = i

        i += 1

    return int(a) if int(a) == a else round(a, 10)

def mult\_div(st):

    li = 0

    while(li < len(st) and is\_num(st[li])):

        li += 1

    if(li == 0):

        raise ArithmeticError('Операции должны быть записаны между оперантами!')

    a = float(st[:li])

    i = li + 1

    while(li < len(st)):

        while(i < len(st) and is\_num(st[i])):

            i += 1

        if(i - li - 1 == 0):

            raise ArithmeticError('Операции должны быть записаны между оперантами!')

        b = float(st[li + 1:i])

        if(st[li] == '\*'):

            a \*= b

        elif(st[li] == '/'):

            if(b == 0):

                raise ArithmeticError('Деление на ноль!')

            a /= b

        else:

            raise ArithmeticError('Посторонний символ \'' + st[li] + '\'!')

        li = i

        i += 1

    return a

1. Все вычисления выполняем в конструкции try-extern, чтобы отловить ошибки при неверно введённых выражениях или делении на 0.

while True:

    print("\nВведите алгебраическое выражение:")

    inp = input()

    fin = change\_format(inp)

    try:

        for el in fin:

            if(not if\_normal(el)):

                raise ArithmeticError('Посторонний символ \'' + el + '\'!')

        print(fin, '=', calculate(fin))

    except Exception as error:

        print(fin, "Произошла ошибка во время рассчёта:", error)

**Вывод**

В результате проделанной работы я, наконец, написал на питоне хоть что-то кроме задач ЕГЭ.