Fatec Ipiranga

CST em Big Data para Negócios

Disciplina: Estrutura de Dados

Pontuação: 0,0 até 2,0

Data entrega: 25/10/2022 – 23:30

Nome: Marcelo Vieira da Silva

Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Atividade**

Dado o código de avaliação posfixa, faça uma alteração que impeça a avaliação de expressões que contenham caracteres (A...Z e a...z).

R:

class Stack:

    def \_\_init\_\_(self):

        self.items = []

    def is\_empty(self):

        return self.items == []

    def push(self, item):

        self.items.append(item)

    def pop(self):

        return self.items.pop()

    def peek(self):

        return self.items[len(self.items) - 1]

    def size(self):

        return len(self.items)

def postfix\_eval(postfix\_expr):

    operand\_stack = Stack()

    token\_list = postfix\_expr.split()

    for token in token\_list:

        if token in "0123456789":

            operand\_stack.push(int(token))

        elif token in "ABCDEFGHIJKLMNOPQRSTUVWXYZ" or token in "abcdefghijklmnoprstuvwxyz":

                      return "Erro! Não possível calcular valor da posfixa com letras "

                      break

        else:

            operand2 = operand\_stack.pop()

            operand1 = operand\_stack.pop()

            result = do\_math(token, operand1, operand2)

            operand\_stack.push(result)

    return operand\_stack.pop()

def do\_math(op, op1, op2):

    if op == "\*":

        return op1 \* op2

    elif op == "/":

        return op1 / op2

    elif op == "+":

        return op1 + op2

    else:

        return op1 - op2

print(postfix\_eval('A B + C D + /'))