МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №12**

по дисциплине: Объектно-ориентированное программирование Тема: Знакомство с Python. Основные структуры данных.

Выполнил: студент группы ПВ-223

Дмитриев А.А.

Проверил:

Черников С.В.

Белгород 2024 г.

**Цель работы:** Приобретение практических навыков создания приложений на

языке Python

**Задание:**

Для выполнения лабораторной работы требуется установить

интерпретатор Python версии 3.5+. Выполнить написание программы-сценария

в соответствии с вариантом задания. Провести тестирование. Оформить отчет.

**Вариант 2:**

Дано вещественное число k. Вставить между некоторыми цифрами,

записанными в файле знак (+), (-), (\*), (/), причем если возможно операция

делания и умножения обязательна, так, чтобы значением получившегося

выражения было число k (вычисление выполняется последовательно без учета

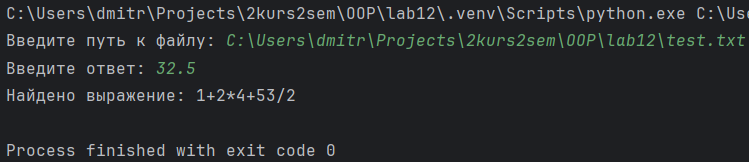
приоритета операций). Напpимеp, в файле хранится: 1,2,3,4,2,2,1 если k=45, тоимеpимеp, в файле хранится: 1,2,3,4,2,2,1 если k=45, то, в файле хранится: 1,2,3,4,2,2,1 если k=45, то подойдет следующая расстановка: 12+34\*2/2-1.Код:

from itertools import product  
  
  
def evaluate\_expression(expression):  
 try:  
 return eval(expression)  
 except ZeroDivisionError:  
 return None  
  
  
def find\_expression(numbers, answer):  
 numbers = list(numbers.replace(",", "").replace(" ", ""))  
  
 for x in numbers:  
 if not x.isdigit() or int(x) <= 0:  
 raise Exception("Invalid input")  
  
 operators = ["\*", "/", "+", "-", ""]  
 n = len(numbers)  
 intermediate\_result = None  
 for ops in product(operators, repeat=n - 1):  
 expr = ""  
 for i in range(n - 1):  
 if ops[i] == "":  
 expr += f"{numbers[i]}{ops[i]}"  
 else:  
 expr = "(" + expr + f"{numbers[i]}){ops[i]}"  
  
 expr += numbers[-1]  
 if abs(evaluate\_expression(expr) - answer) < 0.0001:  
 expr = expr.replace("(", "").replace(")", "")  
 intermediate\_result = expr  
 if ops.count("\*") > 0 and ops.count("/") > 0:  
 return expr  
  
 return intermediate\_result  
  
  
def main():  
 path = str(input("Введите путь к файлу: "))  
 answer = float(input("Введите ответ: "))  
  
 with open(path, "r") as file:  
 numbers = file.read()  
  
 expression = find\_expression(numbers, answer)  
 if expression:  
 print("Найдено выражение:", expression)  
 else:  
 print("Выражение не найдено")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Тесты:

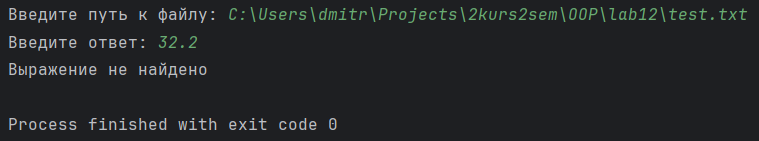
1 2 4 5 3 2

32.5



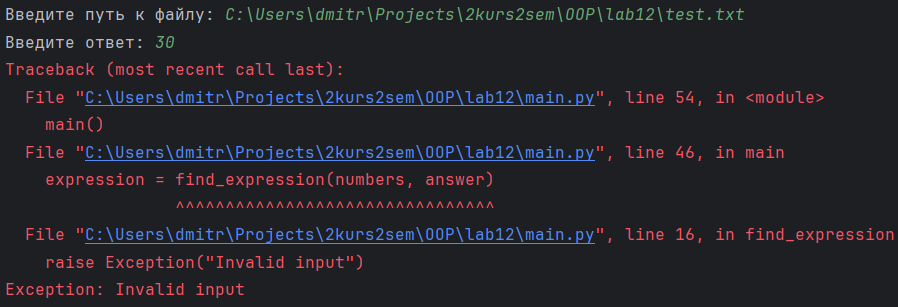
1 2 4 5 3 2

32.2



1 2 4 5 3 0

30



**Вывод:** В ходе лабораторной работы приобрели практические навыки создания приложений на языке Python.