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 № 11**

по дисциплине: Объектно-ориентированное программирование

тема: «Знакомство с языком программирования Python. Базовые

структуры данных.»

Выполнил: ст. группы ПВ-223

Игнатьев Артур Олегович

Проверил:

асс. Черников Сергей Викторович

Белгород 2024г.

**Лабораторная работа №11**

**«Знакомство с языком программирования Python. Базовые**

**структуры данных.»**

**Цель работы:** Познакомится с базовыми конструкциями языка. Получить

навык создания простых приложений. Изучить типы.

**Вариант 3**

На вход подаются данные в форме двумерных «матриц», количество матриц заранее не определено, разделителем между матрицами являются строки. Для каждой матрицы найти все, которые удовлетворяют следующему условию: четность/нечетность соответствующих элементов матриц совпадает. Форма матрицы может быть не полной. Формат вывода требуется соблюсти.

Код программы:

class MatrixProcessor:  
 def read\_matrix(self):  
 matrix = []  
 while True:  
 string = input()  
 if string[0] == '.':  
 return matrix, string[0]  
 elif string[0] == '=':  
 return matrix, string[0]  
 else:  
 matrix.append(string.split())  
  
 def read\_matrices(self):  
 matrices = []  
 while True:  
 t = self.read\_matrix()  
 if t[1] == '=':  
 matrices.append(t[0])  
 return matrices  
 else:  
 matrices.append(t[0])  
  
 def check\_parity(self, matrix):  
 rows = len(matrix)  
 cols = max(len(row) for row in matrix)  
 result = []  
 for i in range(rows):  
 for j in range(cols):  
 if j < len(matrix[i]):  
 result.append(int(matrix[i][j]) % 2)  
 else:  
 result.append(0)  
 return result  
  
 def print\_matrices(self, arr\_first, arr\_second, s):  
 print()  
 for i in range(len(arr\_first)):  
 if i == s:  
 print(f'{" ".join(arr\_first[i])} -> {" ".join(arr\_second[i])}')  
 else:  
 print(f'{" ".join(arr\_first[i])} {" ".join(arr\_second[i])}')  
  
 def main(self):  
 matrices = self.read\_matrices()  
 parity\_results = []  
 for matrix\_index, matrix in enumerate(matrices):  
 parity = self.check\_parity(matrix)  
 for other\_index, other\_matrix in enumerate(matrices):  
 if matrix\_index != other\_index:  
 other\_parity = self.check\_parity(other\_matrix)  
 if parity == other\_parity and (other\_index, matrix\_index) not in parity\_results:  
 parity\_results.append((matrix\_index, other\_index))  
 self.print\_matrices(matrix, other\_matrix, matrix\_index)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 processor = MatrixProcessor()  
 processor.main()

Результат работы программы:

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

**Вывод:** Целью работы было познакомиться с базовыми конструкциями языка, получить навыки создания простых приложений и изучить типы данных.