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-233

Ситников Алексей Павлович

Проверил:

Белгород 2025 г.

**Цель работы:** познакомится с базовыми конструкциями языка. Получить навык создания простых приложений. Изучить базовые типы.

Вариант 13 (6)

В соответствии с вариантом задания требуется выполнить объектную декомпозицию задачи. В качестве одного из обязательных объектов выделить «матрицу». Для реализации соблюдения условия задачи требуется использовать возможности перегрузки операторов. При выводе также требуется выполнить перегрузку соответствующего оператора.

**Задание варианта:** 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: сумма соответствующих столбцов равна. Форма матрицы может быть не полной. Формат вывода требуется соблюсти.

Класс Matrix:

from numpy.matrixlib.defmatrix import matrix  
  
  
class Matrix:  
  
 def \_\_init\_\_(self):  
 self.matrix = []  
  
 def add\_row(self, row):  
 self.matrix.append(row)  
  
  
 def get\_sumCol(self):  
 max\_col = 0  
  
 for i in self.matrix:  
 max\_col = max(max\_col, len(i))  
  
 col\_sum = [0] \* max\_col  
  
 for row in range(0, len(self.matrix)):  
 for col in range(0, len(self.matrix[row])):  
 col\_sum[col] += self.matrix[row][col]  
 return col\_sum  
  
 def print\_row(self, index):  
 if index < len(self.matrix):  
 return ' '.join(map(str, self.matrix[index]))  
 return ' '  
  
 def get\_count\_rows(self):  
 return len(self.matrix)  
  
 def get\_max\_cols(self):  
 max\_cols = 0  
 for i in self.matrix:  
 max\_cols = max(max\_cols, len(i))  
 return max\_cols

Модуль main:

import matrix  
  
def modification\_list(array):  
 while len(array) > 0:  
 if array[-1] == 0:  
 array.pop(-1)  
 else:  
 return array  
 return array  
  
def print\_matrix(array, list\_matrix\_temp):  
 max\_row = 0  
 max\_col = 0  
 str\_ = ""  
  
 for i in array:  
 max\_row = max(list\_matrix\_temp[i].get\_count\_rows(), max\_row)  
  
 temp = list\_matrix\_temp[i].get\_max\_cols()  
  
 max\_col = max (temp + temp - 1 , max\_col)  
  
 for i in range(0, max\_row):  
 for j in array:  
 temp = list\_matrix\_temp[j].print\_row(i)  
 if len(temp) != max\_col:  
 temp += ' ' \* (max\_col - len(temp))  
  
 str\_+= temp  
  
 if i == 0 and j != array[-1]:  
 str\_+=" -> "  
 else:  
 str\_+=" "  
 str\_+="\n"  
 print(str\_)  
  
list\_matrix = []  
first\_matrix = matrix.Matrix()  
list\_matrix.append(first\_matrix)  
  
flag\_delete\_last = 0  
  
while True:  
 line = input()  
 if line == "стоп":  
 if flag\_delete\_last:  
 list\_matrix.pop(-1)  
 break  
 elif line == "":  
 flag\_delete\_last = 1  
 list\_matrix.append(matrix.Matrix())  
 else:  
 row\_elements = [int(x) for x in line.split()]  
 list\_matrix[-1].add\_row(row\_elements)  
 flag\_delete\_last = 0  
  
  
list\_sum = []  
  
for i in list\_matrix:  
 list\_sum.append(i.get\_sumCol())  
  
unique\_sum = []  
answer = []  
  
for i in range(0, len(list\_sum) - 1):  
  
 list\_sum[i] = modification\_list(list\_sum[i])  
  
 if list\_sum[i] in unique\_sum:  
 continue  
 answer.append([])  
 answer[-1].append(i)  
 for j in range(i+1, len(list\_sum)):  
 list\_sum[j] = modification\_list(list\_sum[j])  
 if list\_sum[i] == list\_sum[j]:  
 answer[-1].append(j)  
 if len(answer[-1]) > 1:  
 unique\_sum.append(list\_sum[i])  
  
  
count = 1  
print("\n")  
for i in answer:  
 if len(i) > 1:  
 print("Множество матриц № ", count)  
 print\_matrix(i, list\_matrix)  
 count += 1

Вывод программы:  
Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**Вывод:** в ходе проделанной работы я познакомился с базовыми конструкциями языка python, получил навыки создания простых приложений, изучил базовые типы.