Министерство науки и высшего образования Российской Федерации   
Федеральное государственное бюджетное образовательное   
учреждение высшего образования   
«Алтайский государственный технический   
университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Андреева А.Ю.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Отчёт

по лабораторной работе № 5

**«СОЗДАНИЕ СПИСКОВ С ИСПОЛЬЗОВАНИЕ СПИСКОВЫХ ВКЛЮЧЕНИЙ И ПРОСТЕЙШАЯ ОБРАБОТКА С ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ LAMBLA И MAP»**

по дисциплине «Функциональное и логическое программирование»

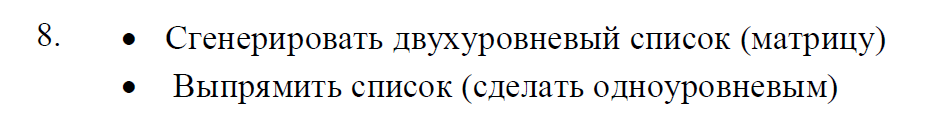
Студент группы ПИ-92 Савиков Д.А.

Преподаватель доцент, к.ф.-м.н. Андреева А.Ю.

Барнаул 2022

Номер в списке – 24

**Задание 1:** Используя списковые включения (list comprehension) сгенерировать одномерный список (возможно в 2-3 этапа). Выполнить обработку списка с использованием функции map и labmda.

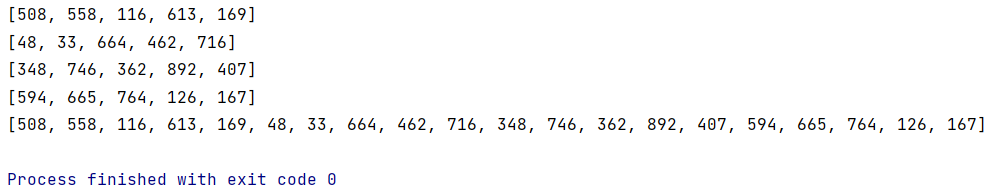


**Решение:**

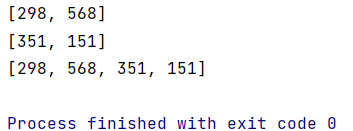
from random import randint  
  
def generate\_matrix(n, m):  
 *"""генерация двухуровневого списка"""* return [[randint(0, 1000) for j in range(m)] for i in range(n)]  
  
def matrix\_to\_list(matrix):  
 *"""выпрямление двухуровнего списка (превращение в одноуровневый)"""* return [x for row in matrix for x in row]  
  
def print\_matrix(matrix):  
 *"""красивый вывод двухуровневого списка"""* for row in matrix:  
 print(row)  
  
def main():  
 matrix = generate\_matrix(4, 5)  
 new\_list = matrix\_to\_list(matrix)  
  
 print\_matrix(matrix)  
 print(new\_list)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Тесты:**

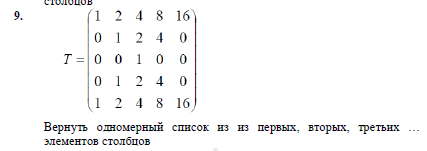
**1)**



2)



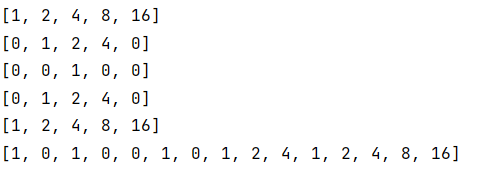
**Задание 2:** Используя списковые включения (list comprehension) сгенерировать двумерный список (возможно в 2-3 этапа) порядка n (для некоторых заданий порядок должен быть нечетным) и выполнить его обработку с использованием функции lambda.

****

**Решение:**

def generate\_matrix(n):  
 *"""генерация матрицы"""  
 # заполнение нулями ниже главной, но выше побочной  
 # выше главной, но ниже побочной, остальное - 1* return [[0 if (i > j and i + j < n - 1) or (i < j and i + j > n - 1)  
 *# заполнение не нулями* else ((lambda: 2 \*\* (j - i))() if i < n/2 else 2 \*\* (abs(n - (j + i + 1)))) for j in range(n)]  
 for i in range(n)]  
  
  
def filter\_not\_false(elem):  
 if elem is not False:  
 return True  
 else:  
 return False  
  
  
def generate\_list(matrix):  
 *"""вернуть одномерный список из из первых, вторых, третьих … элементов столбцов"""* return list(filter(filter\_not\_false, [matrix[i][j] if i >= j else False for i in range(len(matrix)) for j in range(len(matrix))]))  
  
  
def main():  
 some\_matrix = generate\_matrix(5)  
 for row in some\_matrix:  
 print(row)  
  
 new\_list = generate\_list(some\_matrix)  
 print(new\_list)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

1)



2)

