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 №4

по дисциплине «Теория надежности»

тема: «Пьяница на утесе»

Выполнил: ст. группы ВТ-32

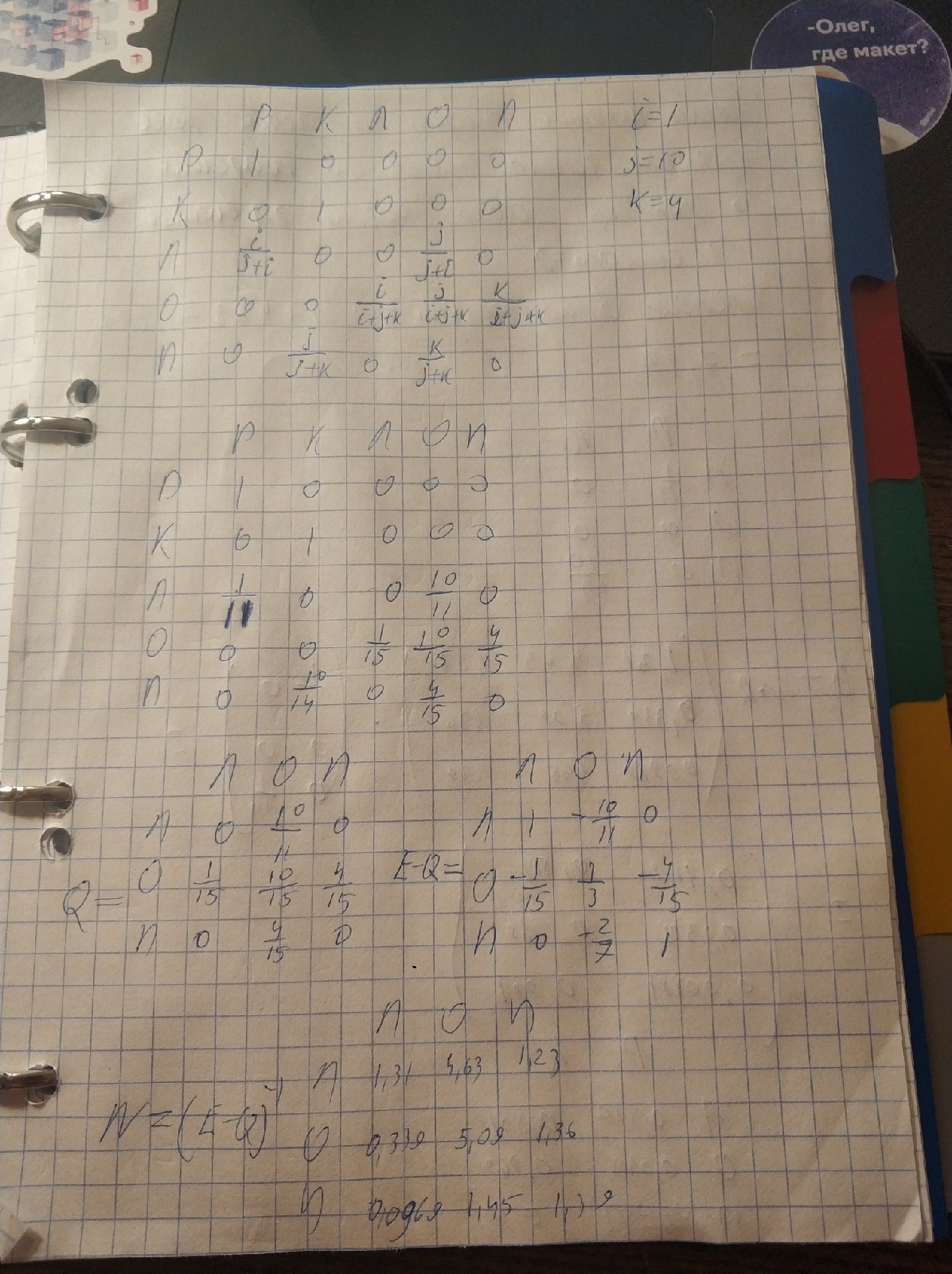
Воскобойников И. С.

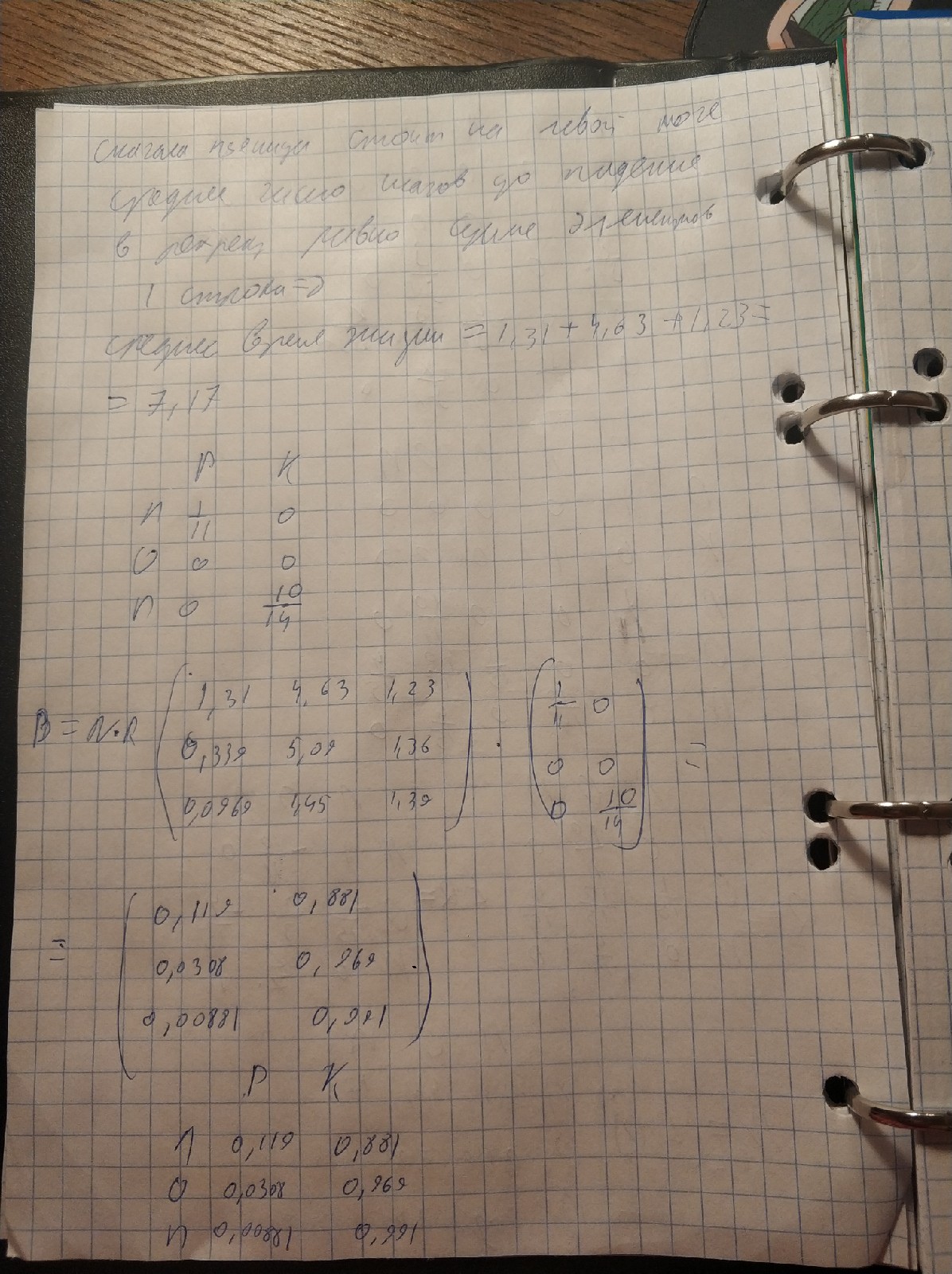
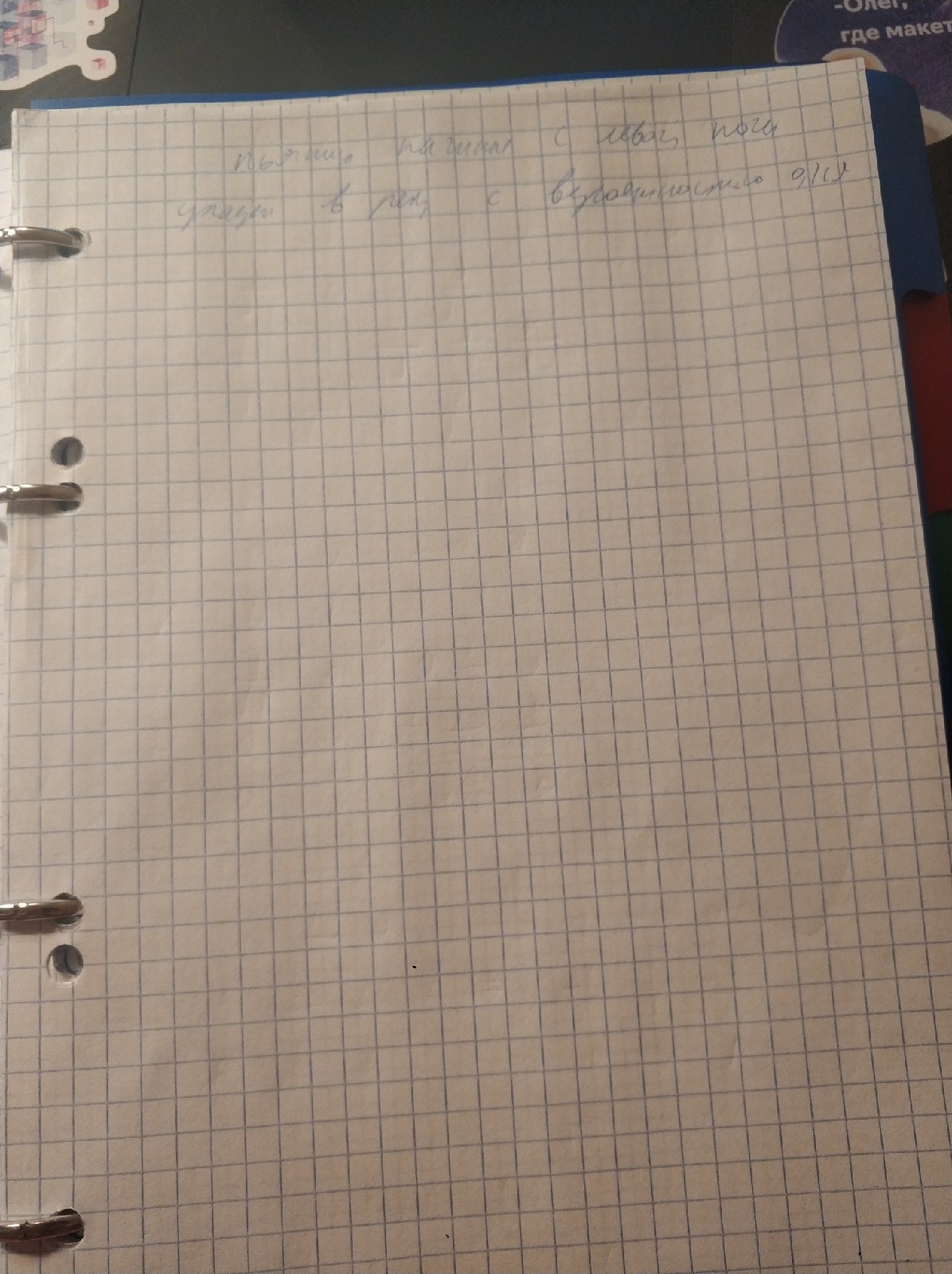
Проверил: Кабалянц П. С.

Белгород 2021 г.

**Задание:**

Пьяница стоит между двумя пропастями, с одной стороны река, с другой копья. В начальный момент времени пьяница стоит на левой ноге. Его поведение задается графом марковского процесса (смотри приложенный рисунок). 1) Необходимо определить среднее время жизни пьяницы и вероятность упасть в реку. 2) Написать программу, которая имитирует поведение пьяницы и выводит среднее количество переходов до падения с утеса и долю падений в реку. 3) Сравнить теоретическую вероятность падению в реку с долей падения в реку критерием сравнения долей.



  from random import random  
  
i = 1  
j = 10  
k = 4  
  
  
  
def step(state):  
 random\_value = random()  
  
 if state == **"Р"**:  
 return state  
  
 if state == **"К"**:  
 return state  
  
 if state == **"Л"**:  
 if random\_value < (1. / 11.):  
 state = **"Р"** else:  
 state = **"О"** return state  
  
 if state == **"О"**:  
 if random\_value < (1. / 15.):  
 state = **"Л"** elif random\_value < (10. / 15.):  
 state = **"О"** else:  
 state = **"П"** return state  
  
 if state == **"П"**:  
 if random\_value < (10. / 14.):  
 state = **"К"** else:  
 state = **"О"** return state  
  
  
def run\_to\_die():  
 state = **'Л'** count\_step = 0  
 while True:  
 count\_step += 1  
 state=step(state)  
 if ((state == **'Р'**) or (state == **'К'**)):  
 return count\_step, state  
  
  
  
  
  
n = 10000  
generate\_result = [run\_to\_die() for \_ in range(n)]  
  
generate\_last\_states = [state for \_, state in generate\_result]  
generate\_time\_life = [time for time, \_ in generate\_result]  
  
print(generate\_last\_states)  
print(generate\_time\_life)  
  
  
  
avg\_die\_time = sum(generate\_time\_life) / len(generate\_time\_life)  
p\_die\_in\_river = generate\_last\_states.count(**"Р"**) / len(generate\_last\_states)  
  
print(**f"Среднее время жизни:** {avg\_die\_time}**"**)  
print(**f"Вероятность упасть в реку:** {p\_die\_in\_river}**"**)  
  
  
from math import sqrt  
  
gen\_p = p\_die\_in\_river  
  
teor\_p = 0.119  
print(**f"|K| =** {abs((gen\_p - teor\_p) / (sqrt(gen\_p \* (1 - gen\_p) / n)))} **< 1.96**\n**"  
 "Разница не большая значит все расчеты были выполненны верно. Значит гипотеза верна"**)

