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 №2

по дисциплине «Теория надежности»

тема: «Игра в имитацию»

Выполнил: ст. группы ВТ-32

Егоров В. Ю.

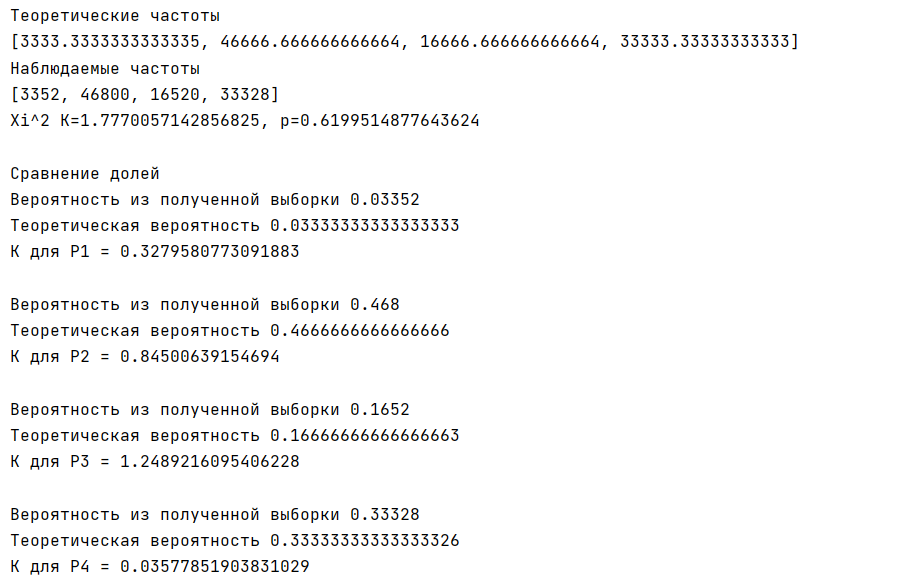
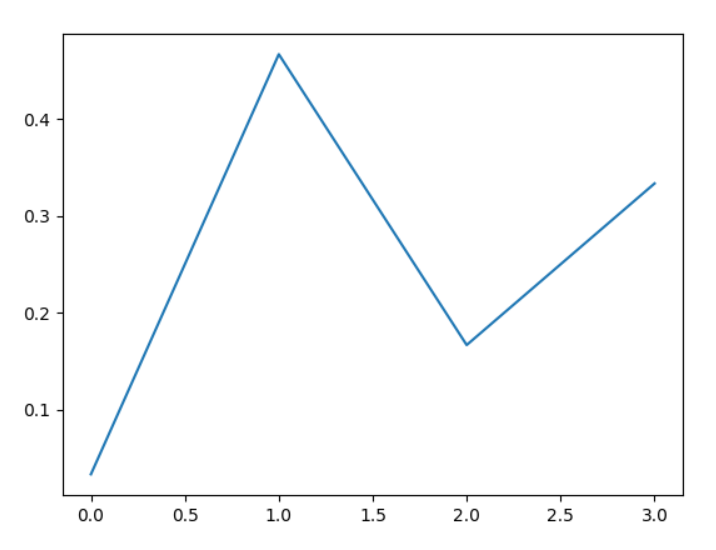
Проверил: Кабалянц П. С.

Белгород 2021 г.

**Задание:**

Дискретная случайная величина ξ принимает значения 0, 1, 2 и 3 с вероятностями соответствующими данным варианта. Используя генератор псевдослучайных чисел, получить 10000 чисел, имитирующих поведение ξ. Проверить на уровне значимости 0,05 соответствие полученных данных распределению ξ, используя критерий сравнения долей. Вероятности: р0=i/2\*(i+j+k), р1=(j+k)/2\*(i+j+k), р2=(i+k)/2\*(i+j+k), р3=j/2\*(i+j+k). Значения i, j, k являются последними тремя цифрами зачетки. Если какая-то из этих трех цифр равна 0, то вместо 0 берется 10.

**Результат работы программы:**



**Код программы:**

from random import random  
from scipy.stats import chisquare  
from math import sqrt  
from matplotlib import pyplot as plt  
  
*# Последние 3 цифры зачетки*i = 1  
j = 10  
k = 4  
  
*# Вероятности*p\_list = [  
 i/(2\*(i+j+k)),  
 (j+k)/(2\*(i+j+k)),  
 (i+k)/(2\*(i+j+k)),  
 j/(2\*(i+j+k))  
]  
print(**"Вероятности"**)  
for i, p in enumerate(p\_list):  
 print(**f"P**{i} **=** {p}**"**)  
  
  
class Discrete\_Random\_Variable:  
 def \_\_init\_\_(self, p\_list):  
 self.p\_list = p\_list  
 self.split\_segment = self.get\_split\_segment(p\_list)  
  
  
 @staticmethod  
 def get\_split\_segment(p\_list):  
 split\_segment = p\_list.copy()  
 for i in range(len(split\_segment)-1):  
 split\_segment[i+1] += split\_segment[i]  
 return split\_segment  
  
  
 def get\_drv\_value(self):  
 random\_value = random()  
 segment = 0  
  
 while random\_value >= self.split\_segment[segment]:  
 segment += 1  
  
 return segment  
  
  
drv = Discrete\_Random\_Variable(p\_list)  
n = 100000  
generated\_p\_values = [drv.get\_drv\_value() for \_ in range(n)]  
print(**"Ваша выборка (10000)"**)  
print(generated\_p\_values)  
  
  
p = [current\_p\*n for current\_p in p\_list]  
print(**"Теоретические частоты"**)  
print(p)  
  
  
generated\_p = [generated\_p\_values.count(p) for p in range(len(p\_list))]  
print(**"Наблюдаемые частоты"**)  
print(generated\_p)  
  
  
K, P = chisquare(generated\_p, p)  
print(**f"Xi^2 K=**{K}**, p=**{P}\n**"**)  
print(**"Сравнение долей"**)  
i=0  
for p\_index in range(len(p)):  
 i=i+1  
 gen\_p = generated\_p[p\_index] / n  
 teor\_p = p[p\_index] / n  
 print(**f"Вероятность из полученной выборки** {gen\_p}\n**"  
 f"Теоретическая вероятность** {teor\_p}**"**)  
 print(**f"K для P**{i} **=** {abs((gen\_p - teor\_p) / (sqrt(gen\_p \* (1-gen\_p) / n)))}\n**"**)  
  
  
print(**"Выборка"**)  
plt.plot(generated\_p)  
plt.show()  
  
print(**"Заданное распределение"**)  
plt.plot(p\_list)  
plt.show()