

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

Профессор				Татарникова Т. М.
должность, уч. степень, звание		подпись, дата		инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

**МОДЕЛИРОВАНИЕ НЕПРЕРЫВНОЙ СЛУЧАЙНОЙ
ВЕЛИЧИНЫ**

Вариант 5

по курсу: Моделирование систем

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4128			Воробьев В. А.
			подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1	Постановка задачи	3
1.1	Порядок выполнения работы	3
1.2	Содержание отчета	3
2	Выполнение работы	4
2.1	Математическая модель	4
2.2	Результаты моделирования	5
2.3	Анализ	11
3	Вывод	12
	ПРИЛОЖЕНИЕ	13

1 Постановка задачи

Цель работы: выполнить программную реализацию генератора непрерывной случайной величины с заданным законом распределения.

1.1 Порядок выполнения работы

1. Построить на основе БСВ пять видов распределений:
 - экспоненциальное по формуле (3.5)
 - равномерное по формуле (3.7)
 - Эрланга порядка K по формуле (3.11)
 - нормальное по формуле (3.14)
 - заданное вариантом (табл. 3.1)
2. Для всех генераторов непрерывной СВ построить гистограмму распределения вероятностей СВ.
3. Оценить M и D для всех видов распределений непрерывной СВ по формулам (3.3), (3.8), (3.10), (3.13). Сравнить полученные значения M и D с эмпирическими.

Вариант 5: Распределение Рэлея

1.2 Содержание отчета

1. Цель, задание и последовательность выполнения работы.
2. Гистограммы распределений непрерывной СВ.
3. Результаты сравнений M и D , полученных эмпирическим путем с соответствующими теоретическими значениями для распределений непрерывной СВ.
4. Выводы о результатах моделирования БСВ.

2 Выполнение работы

2.1 Математическая модель

$$x = F^{-1}(z) \quad (1)$$

Экспоненциальная СВ:

$$F(x) = 1 - e^{-\lambda x} \quad (2)$$

$$M(x) = \frac{1}{\lambda} \quad (3)$$

$$D(x) = \frac{1}{\lambda^2} \quad (4)$$

Равномерная СВ:

$$F(x) = \frac{x - A}{B - A} \quad (5)$$

$$M(x) = A + \frac{B - A}{2} \quad (6)$$

$$D(x) = \frac{(B - A)^2}{12} \quad (7)$$

Эрланговская СВ:

$$x = -\frac{1}{\lambda} \ln(z_1 \dots z_k) \quad (8)$$

$$M(x) = k/\lambda \quad (9)$$

$$D(x) = k/\lambda^2 \quad (10)$$

Нормальная СВ:

$$\begin{aligned} x_1 &= \sqrt{-2 \ln z_1} \sin(2\pi z_2), \\ x_2 &= \sqrt{-2 \ln z_1} \cos(2\pi z_2) \end{aligned} \quad (11)$$

$$M(x) = m \quad (12)$$

$$D(x) = \sigma^2 \quad (13)$$

СВ Рэлея:

$$1 - \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (14)$$

$$M(x) = \sqrt{\frac{\pi}{2}}\sigma \quad (15)$$

$$D(x) = (2 - \pi/2)\sigma^2 \quad (16)$$

2.2 Результаты моделирования

Для моделирования были реализованы скрипты на ЯП Python, которые представлены в Приложении и на GitHub (URI - https://github.com/vladcto/suai-labs/tree/main/6_semester/МодСис).

Была реализована логика для построения графиков СВ. Всего взято 100000 точек. Результат представлен на рисунках 2.1 - 2.5.

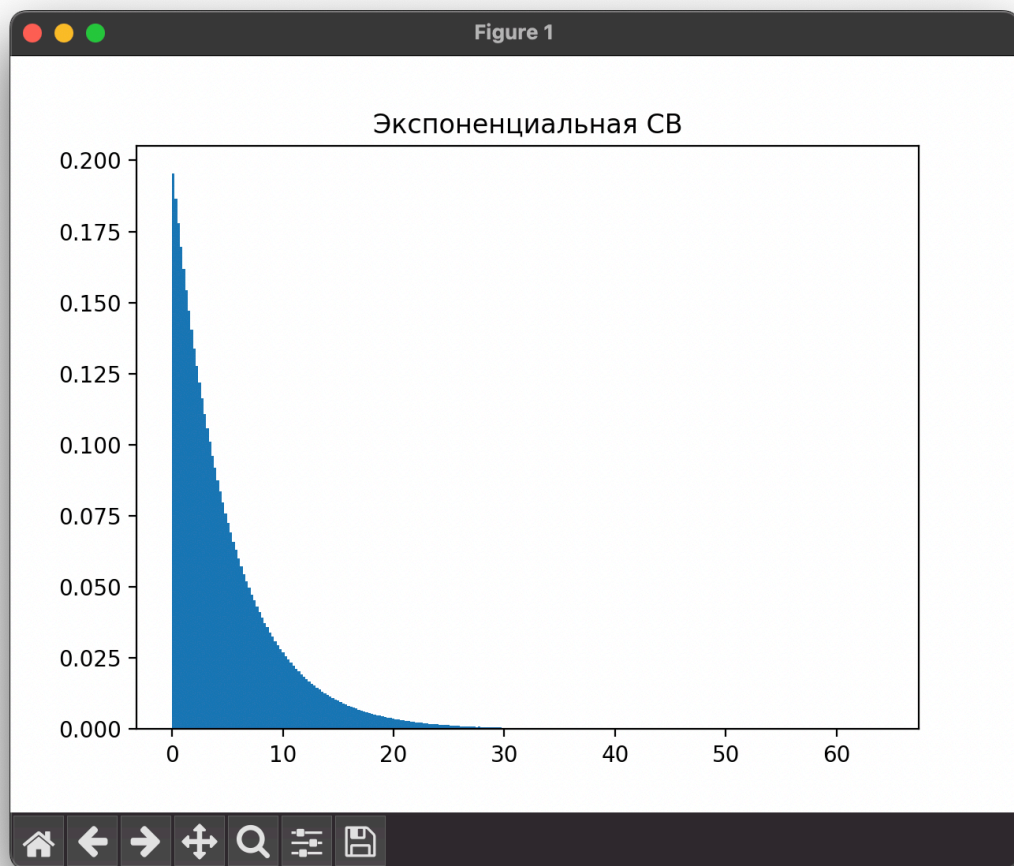


Рисунок 2.1 - Гистограмма экспоненциальной СВ

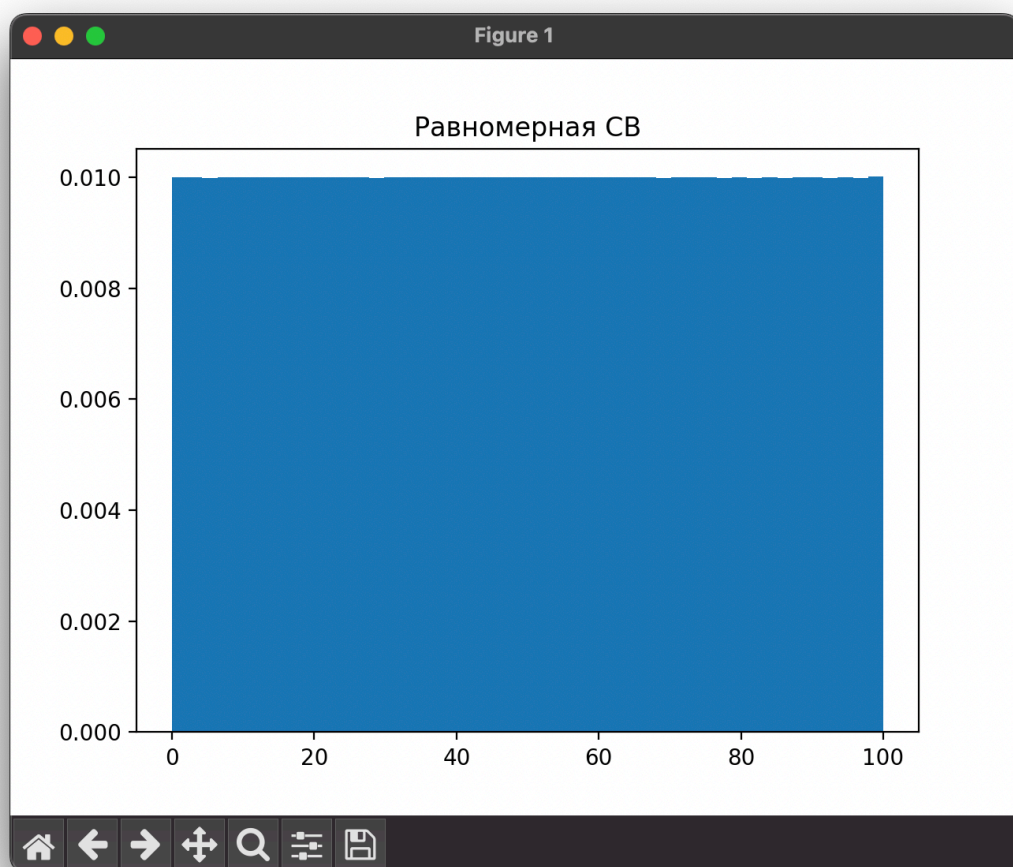


Рисунок 2.2 - Гистограмма равномерной СВ

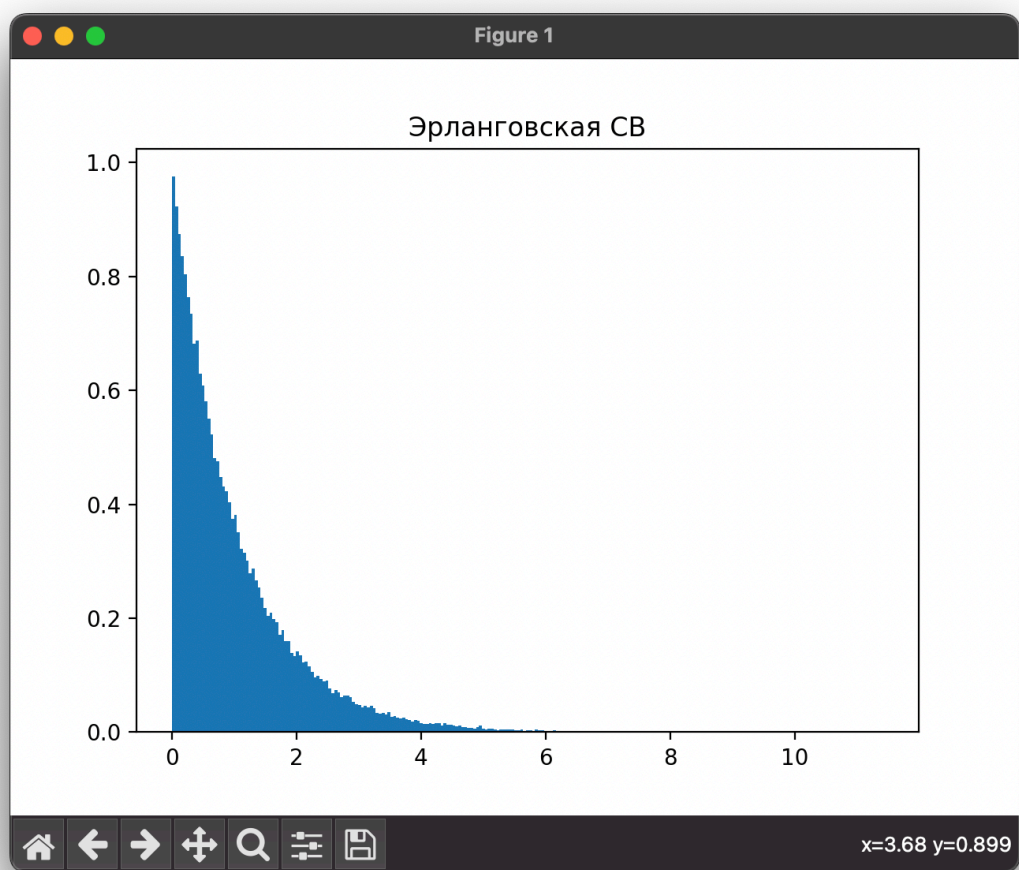


Рисунок 2.3 - Гистограмма эрланговской СВ

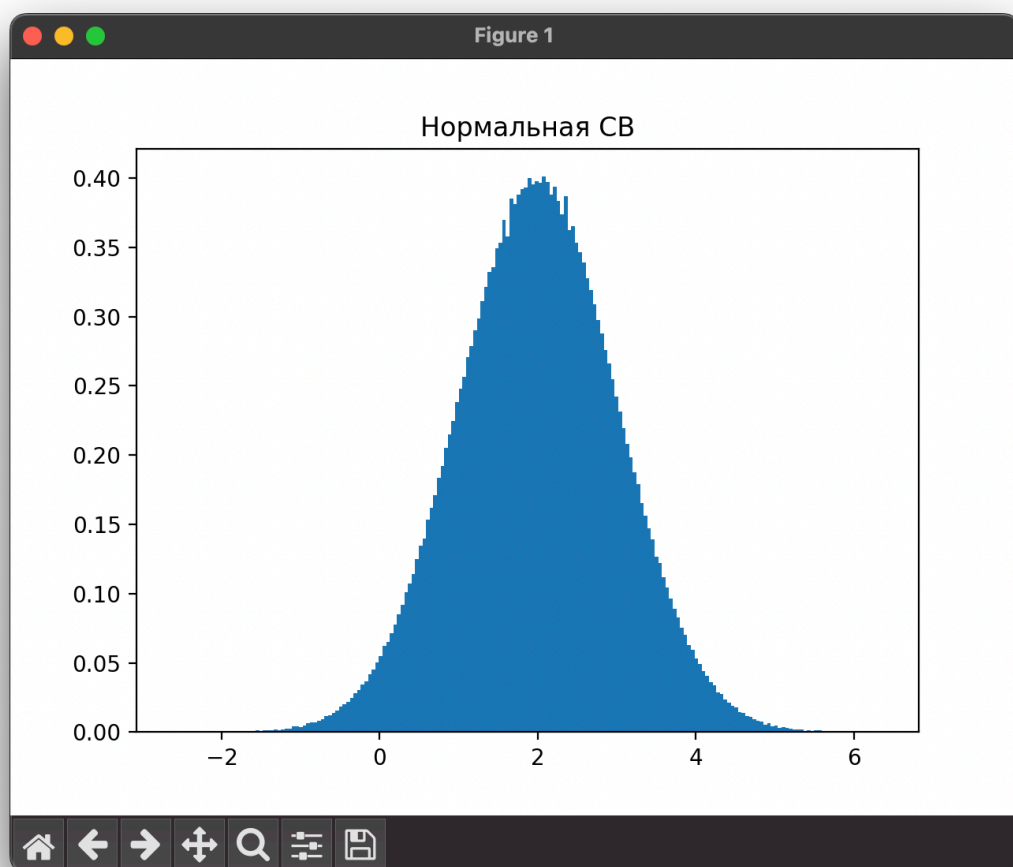


Рисунок 2.4 - Гистограмма нормальной СВ

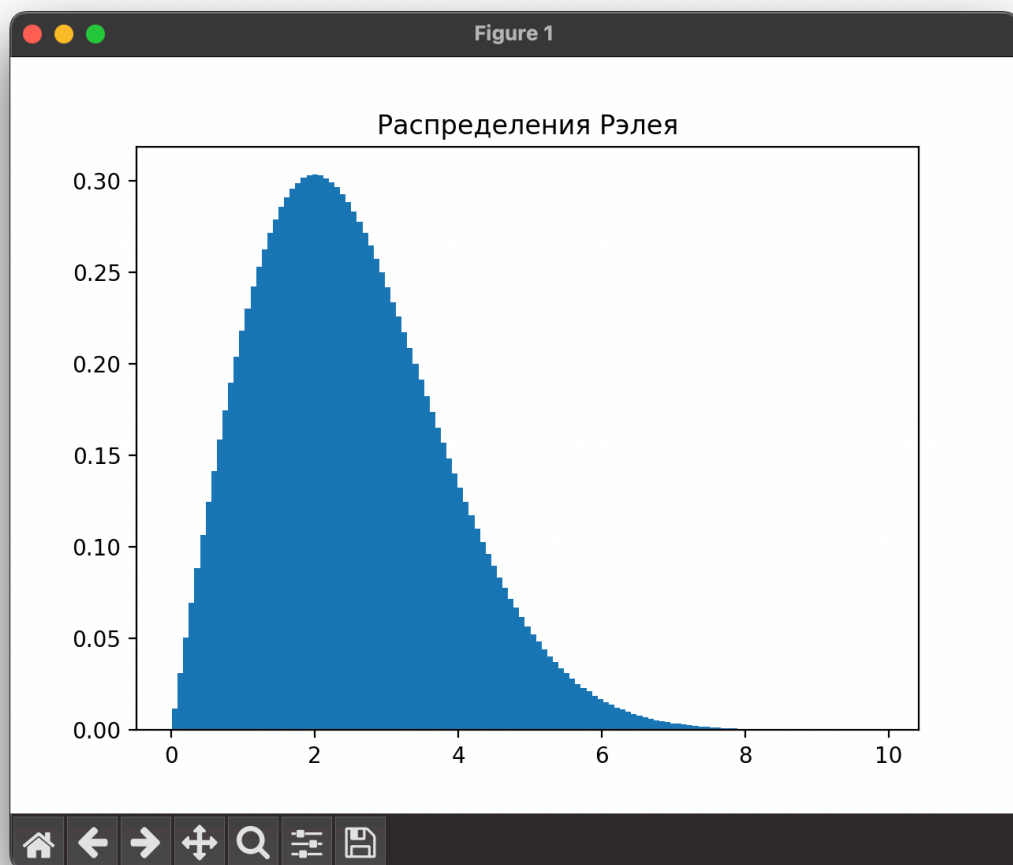


Рисунок 2.5 - Гистограмма распределения Рэля

Также были рассчитаны теоретические и фактические значения математического ожидания M и дисперсии D . Результаты представлены ниже.

```

Экспоненциальная СВ
фкт МО      : 5.000234130764994
фкт дисперсия : 25.018861604607512

эмп МО      : 5.0
эмп дисперсия: 24.999999999999996

=====

Равномерная СВ
фкт МО      : 49.99867642420903
фкт дисперсия : 833.3393020752964

эмп МО      : 50.0
эмп дисперсия: 833.3333333333334

=====

Эрланговская СВ
фкт МО      : 1.007107721930289
фкт дисперсия : 1.014597588746145

эмп МО      : 1.0
эмп дисперсия: 1.0

=====

Нормальная СВ
фкт МО      : 2.000081116565977
фкт дисперсия : 0.9999986178960037

эмп МО      : 2
эмп дисперсия: 1

=====

Распределения Рэлея
фкт МО      : 2.5065750430093297
фкт дисперсия : 1.716651903488

эмп МО      : 2.5066282746310002
эмп дисперсия: 1.7168146928204138

```

Рисунок 2.6 - МО и дисперсия в программе

Таблица 2.1 - значения M и D

Распределение	Факт МО	Факт Дисперсия	Эмп МО	Эмп Дисперсия
Экспоненциальное	1.0	1.0	1.0	1.0
Равномерное	50.0	833.3	50.0	833.3
Эрланга	2.0	4.0	2.0	1.0
Нормальное	2.0	1.0	2.0	1.0
Рэлея	2.5	1.7	2.5	1.7

2.3 Анализ

В ходе статистического анализа базовой случайной непрерывной величины были получены эмпирические значения математического ожидания и дисперсии для каждого из распределений. Для проверки гипотезы о соответствии эмпирических данных теоретическим моделям были проведены соответствующие расчеты.

В результате сравнения эмпирических и теоретических значений математического ожидания и дисперсии было установлено, что они соответствуют друг другу с высокой степенью точности. Кроме того, были построены гистограммы для каждого из распределений, которые демонстрируют высокую степень совпадения с эталонными распределениями.

Полученные результаты свидетельствуют о корректности моделирования базовой случайной непрерывной величины и могут быть использованы для дальнейшего анализа и исследований.

Таким образом, на основе проведенного статистического анализа можно сделать вывод, что эмпирические данные соответствуют теоретическим моделям и что моделирование базовой случайной непрерывной величины было проведено корректно с высокой степенью точности.

3 Вывод

В ходе выполнения данной работы было реализовано программное моделирование генератора непрерывной случайной величины с заданным законом распределения. Были построены гистограммы распределений вероятностей СВ для экспоненциального, равномерного, эрланговского, нормального и распределения Рэлея. Также были рассчитаны теоретические и фактические значения математического ожидания и дисперсии для каждого из распределений.

В результате проведенного анализа было установлено, что теоретические и фактические значения математического ожидания и дисперсии для каждого из распределений совпадают с высокой степенью точности. Это свидетельствует о корректности реализации генератора непрерывной случайной величины.

В ходе выполнения данной работы были получены навыки программной реализации генератора непрерывной случайной величины, построения гистограмм распределений вероятностей СВ, расчета математического ожидания и дисперсии для различных распределений. Данные навыки могут быть использованы в дальнейшем для решения задач, связанных с моделированием случайных процессов.

Таким образом, цель работы была достигнута, были получены навыки программной реализации генератора непрерывной случайной величины и проведения статистического анализа результатов моделирования.

ПРИЛОЖЕНИЕ

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from fish_generator import FishGenerator
4 from generators import *
5
6
7 def plot_histogram(random_vars, func_name):
8     plt.hist(random_vars, bins='auto', density=True)
9     plt.title(f'{func_name}')
10    plt.show()
11
12
13 fish_gen = FishGenerator(seed=42)
14
15 num_vars = 100000
16
17 print("=====\n")
18
19 print("Экспоненциальная СВ")
20 lambd = 0.2
21 random_vars = [exponential_random_variable(
22     next(fish_gen.generate())[0], lambd=lambd) for _ in range
23     (num_vars)]
24 plot_histogram(random_vars, 'Экспоненциальная СВ')
25 print(f"фкт МО : {np.mean(random_vars)}")
26 print(f"фкт дисперсия : {np.var(random_vars)}\n")
27 print(f"эмп МО : {1 / lambd}")
28 print(f"эмп дисперсия : {1 / (lambd**2)}\n")
29
30 print("=====\n")
31
32 print("Равномерная СВ")
33 A = 0
34 B = 100
35 random_vars = [uniform_random_variable(
36     next(fish_gen.generate())[0], A=A, B=B) for _ in range(
37     num_vars)]
38 plot_histogram(random_vars, 'Равномерная СВ')
39 print(f"фкт МО : {np.mean(random_vars)}")
40 print(f"фкт дисперсия : {np.var(random_vars)}\n")
```

```

39 print(f"эмп МО          : {(A + B) / 2}")
40 print(f"эмп дисперсия: {(B - A)**2 / 12}\n")
41
42 print("=====\n")
43
44 print("Эрланговская СВ")
45 lambd = 1
46 k = 1
47 random_vars = [erlang_random_variable(
48     generator=lambda: next(fish_gen.generate())[0], lambd=
49         lambd, k=k) for _ in range(num_vars)]
50 plot_histogram(random_vars, 'Эрланговская СВ')
51 print(f"фкт МО          : {np.mean(random_vars)}")
52 print(f"фкт дисперсия : {np.var(random_vars)}\n")
53 print(f"эмп МО          : {k / lambd}")
54 print(f"эмп дисперсия: {k / (lambd**2)}\n")
55
56 print("=====\n")
57
58 print("Нормальная СВ")
59 mu = 2
60 random_vars = [item for sublist in [normal_random_variable(
61     *next(fish_gen.generate()), mu=mu) for _ in range(
62         num_vars)] for item in sublist]
63 plot_histogram(random_vars, 'Нормальная СВ')
64 print(f"фкт МО          : {np.mean(random_vars)}")
65 print(f"фкт дисперсия : {np.var(random_vars)}\n")
66 print(f"эмп МО          : {mu}")
67 print(f"эмп дисперсия: {1}\n")
68
69 print("=====\n")
70
71 print("Распределения Рэлея")
72 sigma = 2
73 random_vars = [rayleigh_random_variable(
74     next(fish_gen.generate())[0], sigma=sigma) for _ in range(
75         num_vars)]
76 plot_histogram(random_vars, 'Распределения Рэлея')
77 print(f"фкт МО          : {np.mean(random_vars)}")
78 print(f"фкт дисперсия : {np.var(random_vars)}\n")
79 print(f"эмп МО          : {np.sqrt(np.pi / 2) * sigma}")

```

```

77 print(f"эмп дисперсия: {(4 - np.pi) / 2 * sigma**2}\n")

1 import math
2
3 import numpy as np
4
5
6 def exponential_random_variable(z, lambd):
7     return -math.log(z) / lambd
8
9
10 def uniform_random_variable(z, A, B):
11     return A + (B - A) * z
12
13
14 def erlang_random_variable(generator, lambd, k):
15     return np.random.gamma(k, scale=1/lambd)
16
17 def normal_random_variable(z1, z2, mu=2):
18     x1 = math.sqrt(-2 * math.log(z1)) * math.cos(2 * math.pi
19         * z2) + mu
20     x2 = math.sqrt(-2 * math.log(z1)) * math.sin(2 * math.pi
21         * z2) + mu
22     return x1, x2
23
24 def rayleigh_random_variable(z, sigma=1):
25     return sigma * math.sqrt(-2 * math.log(1 - z))

1 import random
2
3
4 class FishGenerator:
5     def __init__(self, seed=None):
6         self.A = [random.randint(0, 2**32 - 1) for _ in range
7             (55)]
8         self.B = [random.randint(0, 2**32 - 1) for _ in range
9             (52)]
10        self.index_A = 0
11        self.index_B = 0
12        random.seed(seed)

12 def generate(self):

```

```

13         while True:
14             self.A[self.index_A] = (
15                 self.A[(self.index_A - 55) % 55] + self.A[(
16                     self.index_A - 24) % 55]) % (2**32)
17             self.B[self.index_B] = (
18                 self.B[(self.index_B - 52) % 52] + self.B[(
19                     self.index_B - 19) % 52]) % (2**32)
20             if self.B[self.index_B] & 1:
21                 yield self.A[self.index_A] / (2**32), self.B[
22                     self.index_B] / (2**32)
23             self.index_A = (self.index_A + 1) % 55
24             self.index_B = (self.index_B + 1) % 52

```