ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель		Е. К. Григорьев
должн., уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1.

МОДЕЛИРОВАНИЕ ГЕНЕРАТОРОВ РАВНОМЕРНО РАСПРЕДЕЛЕННЫХ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ.

по курсу: МОДЕЛИРОВАНИЕ.

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4217		У. А. Мазориев
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

Цель работы:

Получить навыки моделирования наиболее известных генераторов равномерно распределенных псевдослучайных чисел в программной среде Python, а также первичной оценки качества полученных псевдослучайных чисел.

Результат выполнения работы

Программный код на Python, наборы в формате txt, графики в png формате можно посмотреть по ссылке на GitHub: https://github.com/Mrx112426/Modelirovanie/tree/main

Ход выполнения работы

Для начала генерируются 9 наборов псевдослучайных чисел с помощью 3 алгоритмов с заданными количеством числе N = [1000, 5000, 10000] (рис. 1).

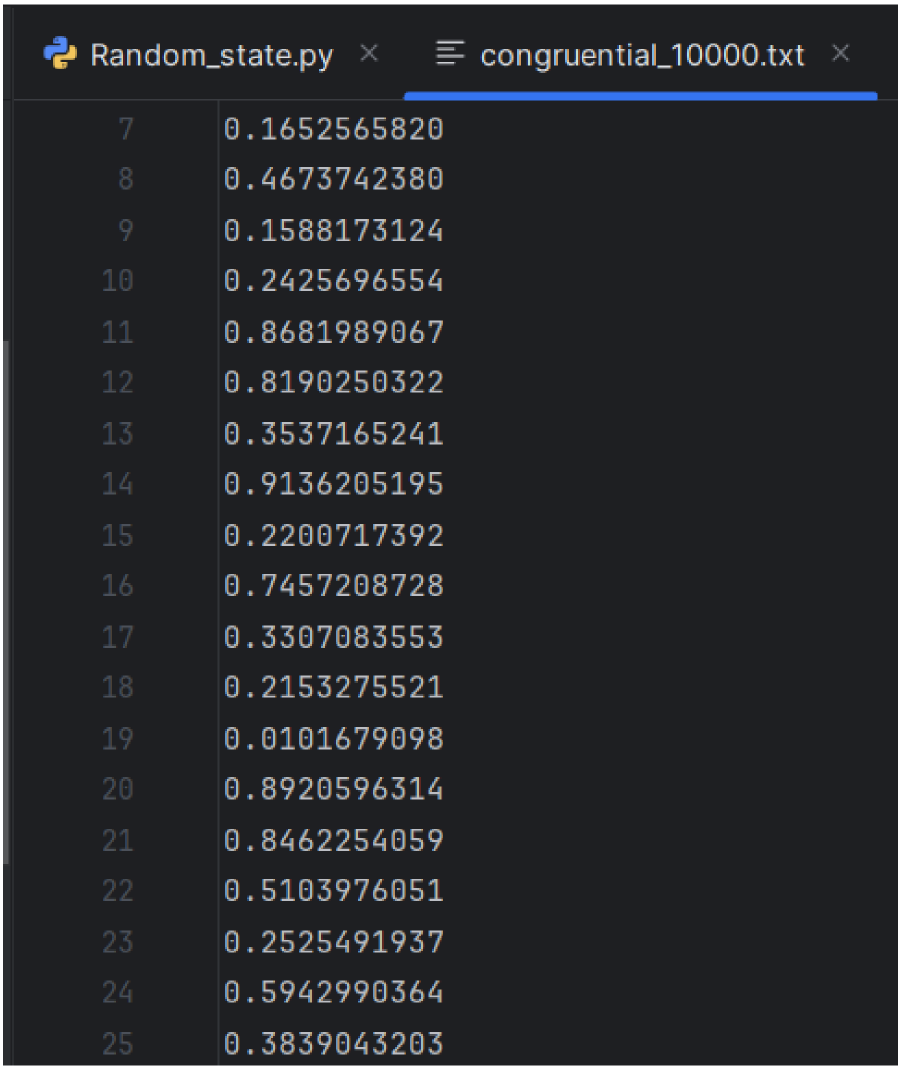

Рисунок 1- Пример сформированного набора

Мультипликативный конгруэнтный генератор. Реализуется следующим образом: $R_{n+1} = \{(AR_n + C) mod M\}$, где A = 7^5, C = 0, M = 2^{32}- 1, также стоит отметит, что в этом способе нужно самостоятельно назначить начальное значение, рекомендуется выбрать $R_n = \ 2^{-p}$ где p зависит от количества двоичных разрядов в мантиссе ячейки. На рисунках 2–4 показаны гистограмма, эмпирическая функция распределения и точечный график для разных N.

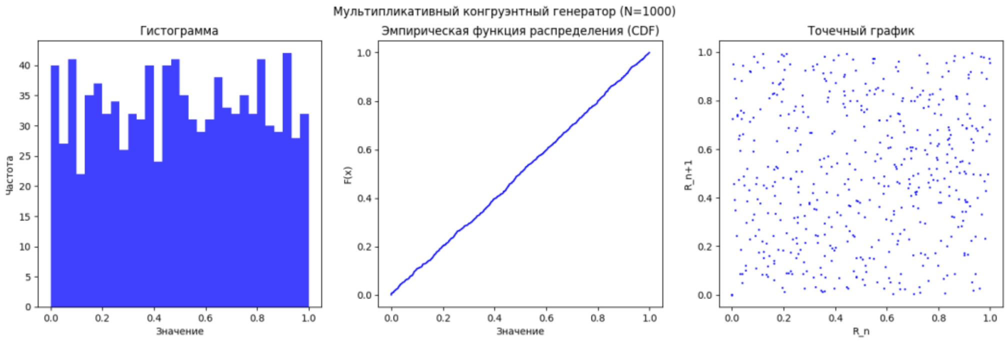

Рисунок 2 – МКГ при N = 1000

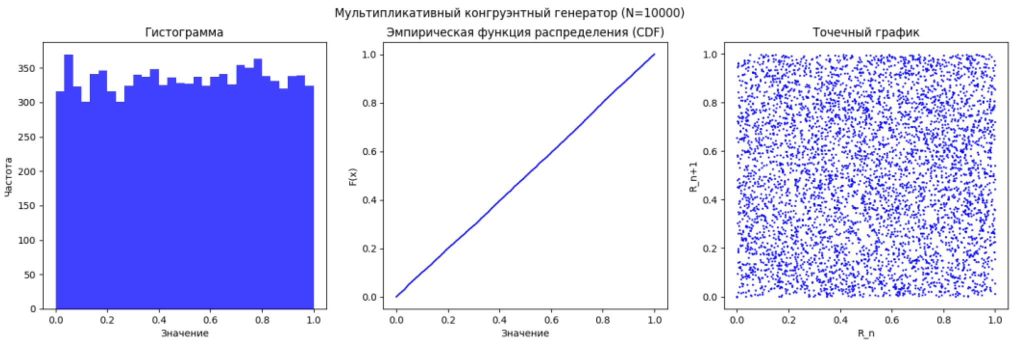

Рисунок 3 – МКГ при N = 5000

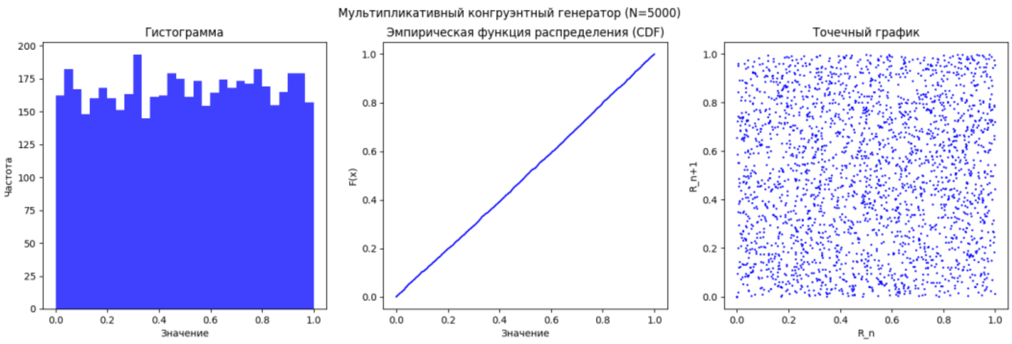

Рисунок 4 – МКГ при N = 10000

Далее реализуется генератор Фибоначчи

$$R^n = \begin{cases} R_{n-a} - R_{n-b}, \text{если } R_{n-a} \geq R_{n-b} \\ R_{n-a} - R_{n-b} + 1, \text{если } R_{n-a} < R_{n-b} \end{cases}$$

Где R_n– псевдослучайные числа в диапазоне [0,1]. a и b - целые положительные числа, называемые лагами или константами запаздывания. Для работы данному генератору требуется max{a,b} предыдущих сгенерированных случайных чисел, которые могут быть реализованы например мультипликативным конгруэнтным генератором. Константы запаздывания рекомендуется выбрать a = 63, b = 31. На рисунках 5–7 показаны гистограмма, эмпирическая функция распределения и точечный график для разных N.

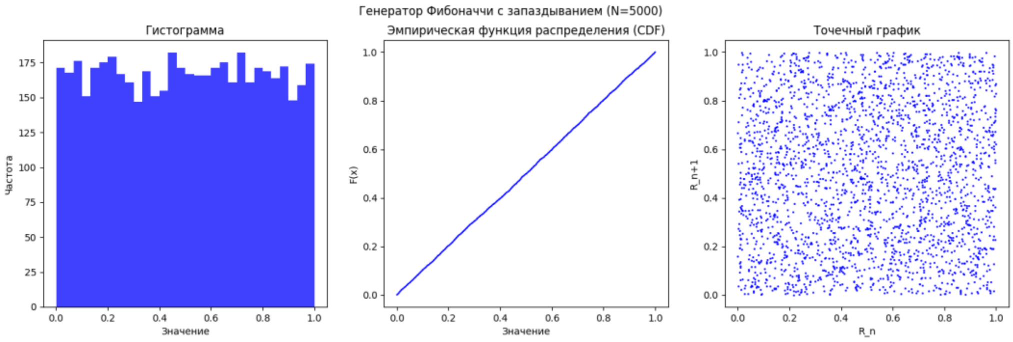

Рисунок 5 – Генератор Фибоначчи при N = 1000

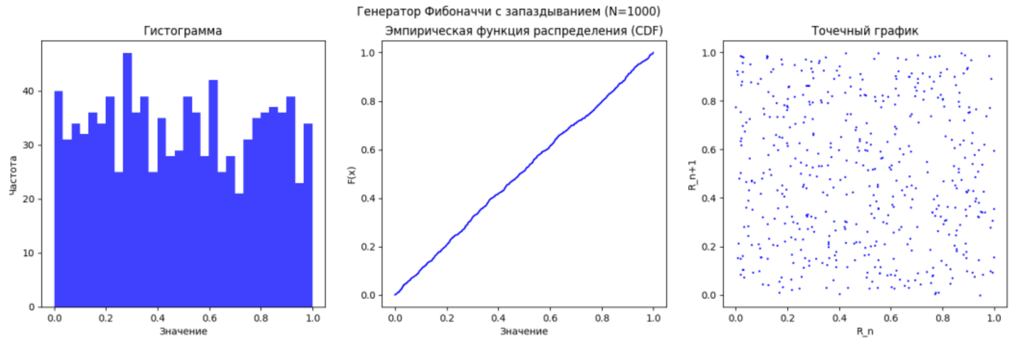

Рисунок 6 – Генератор Фибоначчи при N = 5000

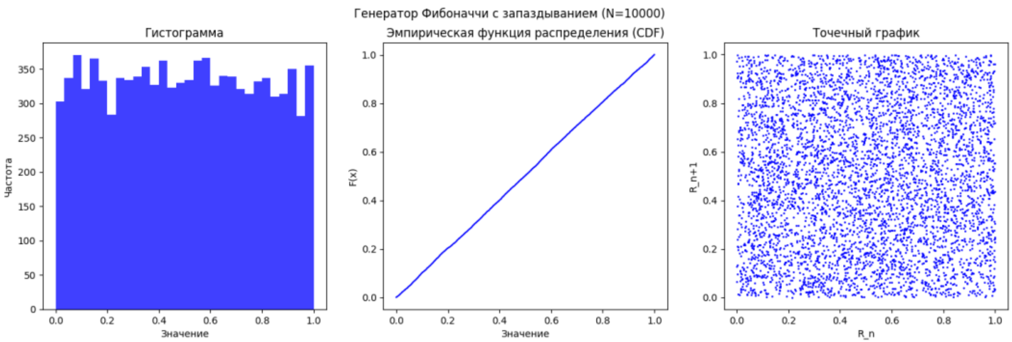

Рисунок 7 – Генератор Фибоначчи при N = 10000

После реализуется Вихрь Мерсенна, в нем генерируются две различные независимые последовательности 32-битовых целых чисел, из которых составляется одно 64-битовое число. На рисунках 8–10 показаны гистограмма, эмпирическая функция распределения и точечный график для разных N.

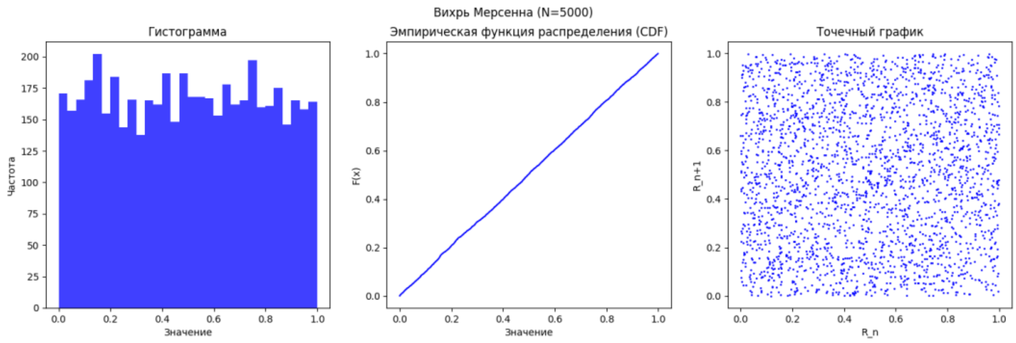

Рисунок 8 – Вихрь Мерсенна при N = 1000

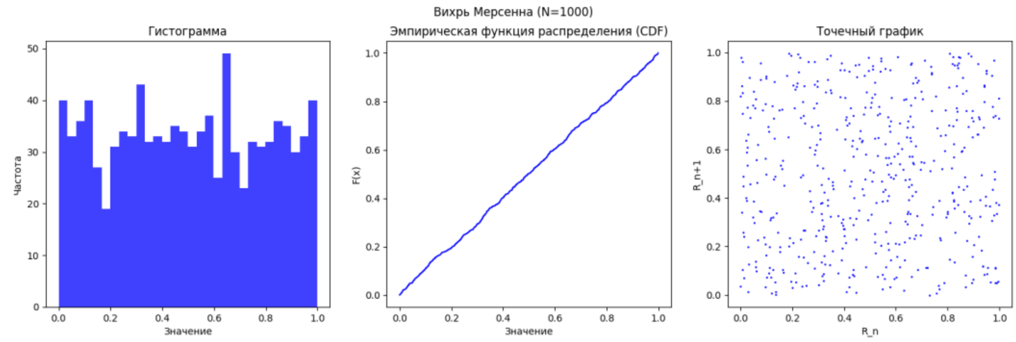

Рисунок 9 – Вихрь Мерсенна при N = 5000

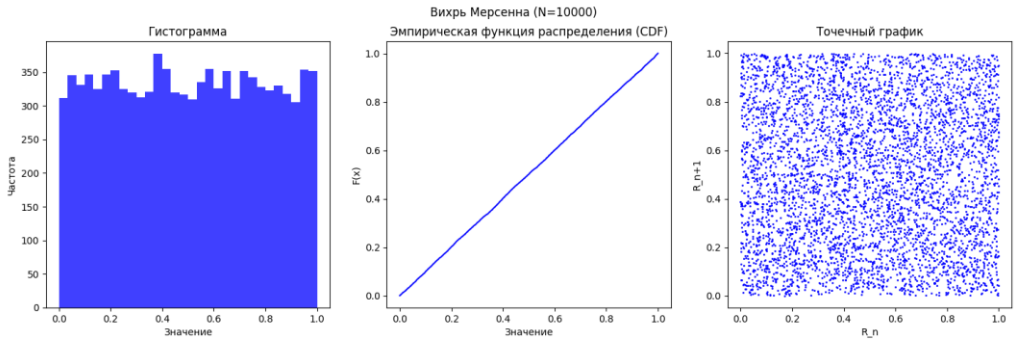

Рисунок 10 – Вихрь Мерсенна при N = 10000

Можно сделать вывод, что на основе гистограмм все три генератора эффективно генерируют равномерные псевдослучайные числа. Однако, на основании данных диаграмм невозможно однозначно определить, какой из генераторов дает более стабильные результаты, поскольку графики всех трех методов выглядят схожими. На малых значениях наблюдаются небольшие колебания, в то время как на больших значениях они практически отсутствуют.

Графики эмпирических функций распределения демонстрируют линейное поведение для всех трех генераторов, что свидетельствует о высоком качестве генерации равномерных псевдослучайных чисел.

Диаграммы рассеяния подтверждают, что все три генератора успешно обеспечивают равномерное распределение чисел, не показывая явных закономерностей.

После вычисляются оценки математического ожидания, дисперсии и среднеквадратического отклонения (СКО) для каждой выборки (рис. 11).

	Метод	N	Матем. Ож	Дисперсия	СКО
0	Мультипликативный конгруэнтный генератор	1000	0.506139	0.079803	0.282493
1	Генератор Фибоначчи с запаздыванием	1000	0.493851	0.081827	0.286054
2	Вихрь Мерсенна	1000	0.522640	0.085855	0.293010
3	Мультипликативный конгруэнтный генератор	5000	0.503010	0.081325	0.285176
4	Генератор Фибоначчи с запаздыванием	5000	0.501405	0.083843	0.289556
5	Вихрь Мерсенна	5000	0.497823	0.083819	0.289514
6	Мультипликативный конгруэнтный генератор	10000	0.500454	0.082354	0.286974
7	Генератор Фибоначчи с запаздыванием	10000	0.497208	0.083984	0.289800
8	Вихрь Мерсенна	10000	0.495530	0.083317	0.288646

Рисунок 11 - Сравнение показателей по каждому методу

Существенных отличий между методами нет, все они хорошо подходят для генерации равномерных случайных чисел.

После аналитически вычисляются значения математического ожидания, дисперсии и среднеквадратического отклонения для равномерного распределения с параметрами (0, 1), также построить графики плотности распределения f(x) и функции распределения F(x) равномерного закона распределения (рис. 6).

Для равномерного распределения (a, b) математическое ожидание рассчитывается по формуле:

$$M = \frac{a+b}{2}$$

Дисперсия:

$$D = \frac{(b-a)^2}{12}$$

Среднеквадратическое отклонение:

$$\sigma = \sqrt{D}$$

Данные значения вычисляются для равномерного распределения с параметрами (0, 1):

$$M = \frac{0+1}{2} = 0{,}5$$

$$D = \frac{(1-0)^2}{12} = 0{,}0833333333333\ldots$$

$$\sigma = \sqrt{0{,}083} = 0{,}288$$

Плотность вероятности для равномерного распределения (0, 1) описывается следующей функцией:

$$f(x) = \begin{cases} 1, \text{если } 0 \le x \le 1 \\ 0, \text{иначе} \end{cases}$$

Функция распределения для равномерного распределения:

$$F(x) \begin{cases} 0, \text{если } x < 0 \\ x, \text{если } 0 \le x \le 1 \\ 1, \text{если } x > 1 \end{cases}$$

После визуализируются графики плотности распределения f(x) и функции распределения F(x) равномерного закона распределения (рис. 12).

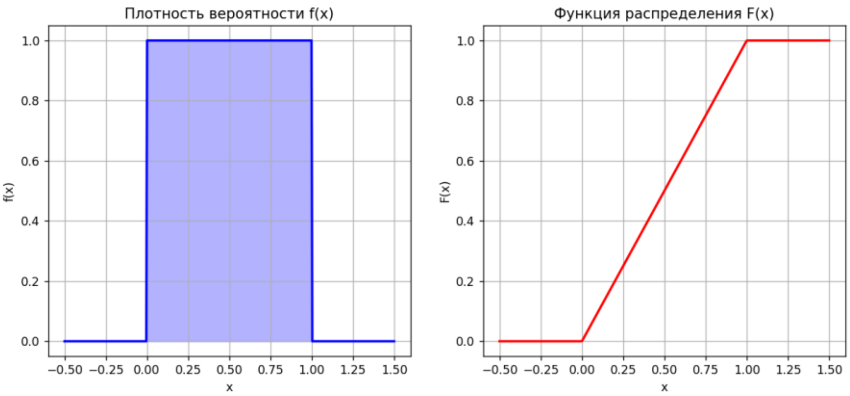

Рисунок 12 – График плотности вероятности и функции распределения

Вывод

В ходе лабораторной работы были освоены методы моделирования наиболее известных генераторов равномерно распределённых псевдослучайных чисел в среде Python, а также проведена их первичная оценка.

Анализ математических характеристик, таких как математическое ожидание, дисперсия и стандартное отклонение, показал, что генератор Вихрь Мерсенна демонстрирует несколько более высокое качество генерации по сравнению с мультипликативным конгруэнтным генератором и генератором Фибоначчи с запаздыванием.