



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

ОТЧЕТ

по Лабораторной работе №3

по курсу «Моделирование»

на тему: «Генераторы псевдослучайных чисел»

Студент ИУ7-73Б
(Группа)

(Подпись, дата)

К.Э. Ковалец
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

И.В. Рудаков
(И. О. Фамилия)

2022 г.

Содержание

1	Задание	3
2	Теоретическая часть	4
2.1	Генераторы псевдослучайных чисел	4
2.2	Табличный способ	4
2.3	Алгоритмический способ	4
2.4	Реализуемый алгоритмический способ	4
2.5	Критерий случайности	5
3	Результаты работы	6
3.1	Листинги программы	6
3.2	Демонстрация работы программы	7

1 Задание

Разработать графический интерфейс, который позволяет сгенерировать последовательность псевдослучайных чисел алгоритмическим и табличным методами, а также рассчитать коэффициенты их случайности. Необходимо предусмотреть возможность ввода чисел.

2 Теоретическая часть

2.1 Генераторы псевдослучайных чисел

Среди способов получения последовательности псевдослучайных чисел различают:

- аппаратные;
- табличные;
- алгоритмические.

2.2 Табличный способ

Если случайные числа, оформленные в виде таблицы, помещать во внешнюю или оперативную память ЭВМ, предварительно сформировав из них соответствующий файл, то такой способ будет называться табличным. Однако этот способ получения случайных чисел при моделировании систем на ЭВМ обычно рационально использовать при сравнительно небольшом объеме таблицы и, соответственно, файла чисел, когда для хранения можно применять оперативную память. Хранение файла во внешней памяти при частом обращении в процессе статистического моделирования не рационально, так как вызывает увеличение затрат машинного времени при моделировании системы из-за необходимости обращения к внешнему накопителю.

2.3 Алгоритмический способ

Алгоритмический способ — это способ получения последовательности случайных чисел, основанный на формировании случайных чисел в ЭВМ с использованием специальных алгоритмов и реализующих их программ.

2.4 Реализуемый алгоритмический способ

В качестве используемого метода генерации последовательности случайных чисел был выбран линейный конгруэнтный метод. Он заключается в том, что каждое последующее число образуется на основе предыдущего по формуле:

$$X_{n+1} = (a \cdot X_n + c) \bmod m, \quad (2.1)$$

где a , c – коэффициенты, а m – модуль, которые подобраны специальным образом. В данной лабораторной работе $a = 36261$, $c = 66037$, $m = 312500$.

2.5 Критерий случайности

В качестве критерия случайности было использовано отношение средних арифметических четных и нечетных элементов друг к другу. Чем ближе коэффициент к 1, тем последовательность более случайна.

3 Результаты работы

3.1 Листинги программы

В листинге 3.1 представлен класс *MyRandom*, отвечающий за вычисление случайного числа линейным конгруэнтным методом, а также нахождение коэффициента случайности.

Листинг 3.1 — class MyRandom

```
1  class MyRandom:
2      def __init__(self):
3          self.current = 1
4          self.a = 36261
5          self.c = 66037
6          self.m = 312500
7
8      def getNumber(self, minNumb, maxNumb):
9          '''
10         Линейный конгруэнтный метод
11         '''
12         self.current = (self.a * self.current + self.c) % self.m
13         return int(minNumb + self.current % (maxNumb - minNumb))
14
15     def getCoeff(self, numbers: list):
16         '''
17         Отношение средних арифметических четных и нечетных элементов друг к другу.
18         Чем ближе коэффициент к 1, тем последовательность более случайна.
19         '''
20         evenNumbers = list()
21         oddNumbers = list()
22
23         for numb in numbers:
24             if numb & 1:
25                 oddNumbers.append(numb)
26             else:
27                 evenNumbers.append(numb)
28
29         if len(evenNumbers) == 0 or len(oddNumbers) == 0:
30             return 0
31
32         avgEvenNumbers = sum(evenNumbers) / len(evenNumbers)
33         avgOddNumbers = sum(oddNumbers) / len(oddNumbers)
34         avgArr = [avgEvenNumbers, avgOddNumbers]
35
36         return min(avgArr) / max(avgArr)
```

3.2 Демонстрация работы программы

На рисунках 3.1 - 3.2 представлены примеры работы программы.

Лабораторная работа №3

ТАБЛИЧНЫЙ МЕТОД				АЛГОРИТМИЧЕСКИЙ МЕТОД			
№	1 разряд	2 разряда	3 разряда	№	1 разряд	2 разряда	3 разряда
1	1	17	196	1	8	38	798
2	8	31	520	2	5	95	615
3	7	38	192	3	2	52	952
4	5	82	513	4	9	39	209
5	3	42	530	5	6	26	786
6	8	47	454	6	3	53	583
7	8	51	775	7	0	30	500
8	0	30	706	8	7	97	137
9	7	78	421	9	4	54	694
10	3	48	836	10	1	21	371
11	7	18	363	11	8	88	568
12	2	97	559	12	5	35	885
13	1	92	499	13	2	52	922
14	3	18	594	14	9	49	279
15	1	46	173	15	6	46	256

Коэф. 0.8 0.98 0.99 Коэф. 0.8 0.98 1.0

Сгенерировать случайные числа и рассчитать критерий оценки случайности

Ручное тестирование Информация о программе

Рисунок 3.1 – Алгоритмический и табличный методы

Лабораторная работа №3

РУЧНОЕ ТЕСТИРОВАНИЕ

1 разряд

1	0
2	2
3	1
4	6
5	3
6	5
7	8
8	4
9	1
10	8

Кэф. 0.54

Расчитать критерий оценки случайности

Рисунок 3.2 – Ручной ввод последовательности