



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: «Псевдослучайные числа»

Студент ИУ7-73Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Р. Р. Хамзина  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

И. В. Рудаков  
(И. О. Фамилия)

2022 г.

# СОДЕРЖАНИЕ

<b>1</b>	<b>Задание</b>	<b>3</b>
1.1	Табличный способ	3
1.2	Алгоритмический способ	3
1.3	Критерий оценки случайности	3
<b>2</b>	<b>Реализация</b>	<b>5</b>
2.1	Детали реализации	5
2.2	Полученный результат	7

# 1 Задание

Реализовать программу с графическим интерфейсом для генерации последовательностей псевдослучайных чисел с использованием табличного и алгоритмического способов и определения коэффициента случайности полученных и введенных последовательностей.

## 1.1 Табличный способ

Источником случайных чисел при работе табличного генератора является таблица случайных чисел, расположенная в памяти ЭВМ.

При реализации табличного способа в лабораторной работе использовалась таблица из книги случайных чисел «A Million Random Digits with 100,000 Normal Deviates» организации RAND. Обход таблицы выполнялся слева направо сверху вниз.

## 1.2 Алгоритмический способ

Работа алгоритмического генератора основана на создании случайных чисел с помощью определенных алгоритмов.

При реализации алгоритмического способа в лабораторной работе использовался линейный конгруэнтный метод. В указанном методе каждое новое число определяется предшествующим числом в соответствии с формулой:

$$y_{n+1} = (a \cdot y_n + \mu) \mod m, n \geq 0, \quad (1.1)$$

где  $a > 0$  — множитель,  $\mu \geq 0$  — приращение,  $m > 0$  — модуль. В данной лабораторной работе использовались следующие значения:  $a = 84589$ ,  $\mu = 45989$ ,  $m = 217728$ .

## 1.3 Критерий оценки случайности

При реализации лабораторной работы был составлен критерий оценки случайности, при котором коэффициент случайности определяется следующим образом:

1. Вычисляется список *differences* модулей разности чисел, расположенных рядом:

$$difference_i = |number_{i+1} - number_i|. \quad (1.2)$$

2. Элементы списка *differences* увеличиваются на единицу.
3. Определяется максимальное значение *max\_difference* элементов списка *differences*.
4. Вычисляется список *ratios* отношений элементов списка *differences* к значению *max\_difference*:

$$ratio_i = \frac{difference_i}{max\_difference}. \quad (1.3)$$

5. Коэффициент случайности *chance* определяется как среднее арифметическое элементов списка *ratios*.

Коэффициент случайности может принимать значения из интервала  $(0; 1]$ . При коэффициенте случайности, равном единице, последовательность не является случайной. Последовательность стремится к случайной при приближении коэффициента случайности к нулю.

## 2 Реализация

### 2.1 Детали реализации

На листинге 2.1 показана реализация функции генерации последовательности псевдослучайных чисел с использованием табличного способа.

Листинг 2.1 – Генерация последовательности псевдослучайных чисел с использованием табличного способа

```
1 def get_sequence(self, digits_number, elements_number):
2     sequence = []
3
4     divider = pow(10, digits_number)
5
6     while elements_number:
7         number = self.table.read(6)
8
9         if number == ',':
10             self.table.seek(0)
11             number = self.table.read(6)
12
13         number = int(number[:5])
14
15         while number:
16             if elements_number:
17                 random_number = number % divider
18
19                 if len(str(random_number)) == digits_number:
20                     sequence.append(random_number)
21                     elements_number -= 1
22
23                 number //= 10
24             else:
25                 return sequence
26
27     return sequence
```

На листинге 2.2 представлена реализация функции генерации последовательности псевдослучайных чисел с использованием линейного конгруэнтного метода.

Листинг 2.2 – Генерация последовательности псевдослучайных чисел с использованием линейного конгруэнтного метода

```
1 def get_sequence(self, digits_number, elements_number):
2     sequence = []
3
4     divider = pow(10, digits_number)
5
6     while elements_number:
7         self.y = (self.a * self.y + self.mu) % self.m
8         number = self.y
9
10        while number:
11            if elements_number:
12                random_number = number % divider
13
14                if len(str(random_number)) == digits_number:
15                    sequence.append(random_number)
16                    elements_number -= 1
17
18                number //= 10
19            else:
20                return sequence
21
22    return sequence
```

На листинге 2.3 показана реализация функции определения коэффициента случайности последовательности псевдослучайных чисел.

Листинг 2.3 – Определение коэффициента случайности последовательности псевдослучайных чисел

```
1 def get_chance(numbers):
2     differences = [abs(numbers[i + 1] - numbers[i]) + 1
3                    for i in range(len(numbers) - 1)]
4     max_difference = max(differences)
5     ratios = [difference / max_difference
6               for difference in differences]
7     chance = sum(ratios) / len(ratios)
8     return chance
```

## 2.2 Полученный результат

На рисунке 2.1 представлена страница программы со сгенерированными последовательностями псевдослучайных чисел с использованием табличного и алгоритмического способов и определенными коэффициентами случайности полученных последовательностей.

**Генерация псевдослучайных чисел**

О программе

**Ручной ввод**

1
1
2
3
4
5
6
7
8
9
10

Коэффициент:

1
1

Вычислить

**Табличный способ**

1	2	3
1	2	30
2	8	73
3	4	10
4	1	70
5	9	47
6	9	33
7	8	23
8	6	32
9	0	43
10	3	22

Коэффициенты:

1	2	3
1	0.42452	0.33394
	0.33454	

Сгенерировать числа

**Алгоритмический способ**

1	2	3
1	3	70
2	8	87
3	2	38
4	0	93
5	8	87
6	1	98
7	8	29
8	2	92
9	8	60
10	7	16

Коэффициенты:

1	2	3
1	0.43035	0.3527
	0.35943	

Вычислить

Рисунок 2.1 – Табличный и алгоритмический генераторы

На рисунке 2.2 показаны части страницы программы с определенными коэффициентами случайности следующих введенных последовательностей:

1. Введена строго возрастающая последовательность чисел. Коэффициент равен единице: последовательность не является случайной.
2. Введена последовательность, состоящая из одинаковых чисел. Коэффициент равен единице: последовательность не является случайной.
3. Введена строго убывающая последовательность чисел. Коэффициент равен единице: последовательность не является случайной.

4. Введена последовательность, состоящая из одинаковых пар чисел. Коэффициент равен единице: последовательность не является случайной.
5. Введена последовательность чисел от 0 до 9 в случайном порядке. Коэффициент равен 0.65278: последовательность стремится к случайной.

Ручной ввод		Ручной ввод		Ручной ввод		Ручной ввод		Ручной ввод	
1		1		1		1		1	
1	0	1	7	1	9	1	4	1	5
2	1	2	7	2	8	2	5	2	8
3	2	3	7	3	7	3	4	3	2
4	3	4	7	4	6	4	5	4	6
5	4	5	7	5	5	5	4	5	7
6	5	6	7	6	4	6	5	6	0
7	6	7	7	7	3	7	4	7	4
8	7	8	7	8	2	8	5	8	9
9	8	9	7	9	1	9	4	9	3
10	9	10	7	10	0	10	5	10	1
Коэффициент:		Коэффициент:		Коэффициент:		Коэффициент:		Коэффициент:	
1		1		1		1		1	
1	1.0	1	1.0	1	1.0	1	1.0	1	0.65278
Вычислить		Вычислить		Вычислить		Вычислить		Вычислить	

Рисунок 2.2 – Определение коэффициентов случайности последовательности