



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

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ» (ИУ7)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № 3

Тема: Генераторы псевдослучайных последовательностей

Дисциплина: Моделирование

Студент

ИУ7-72Б

(Группа)

(Подпись, дата)

В.А. Иванов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

И.В. Рудаков

(И.О. Фамилия)

Москва, 2021

1. Задание

Используя алгоритмический и табличный метод сгенерировать последовательность из 1000 одно-, дву- и трёхразрядных чисел. Отобразить первые 10 в графическом интерфейсе. Составить статистический критерий оценки случайности последовательностей и вывести его значение для каждой последовательности. Также предусмотреть возможность получения оценки для 10 чисел введённых пользователем.

2. Результаты

2.1. Теория

Генераторы случайных чисел разделяются на:

- аппаратные
- табличные
- алгоритмические

В данной лабораторной работе реализованы два последних из них

2.1.1. Алгоритмический генератор

Алгоритмический генератор использует одно или несколько значений для вычисления нового числа, после чего также используется для генерации последующих чисел.

В данной работе был использован линейный конгруэнтный метод. Последовательность в нём вычисляется по формуле

$$X_{n+1} = (aX_n + c) \bmod m, n \geq 1 \quad (2.1)$$

В данной лабораторной использованы значения из минимального стандартного генератора случайных чисел: $a = 16807$, $c = 2147483647$, $m = MAX_INT$. В качестве начального значения X_1 используется системное время.

2.1.2. Табличный генератор

Табличный генератор использует заранее подготовленные последовательности хранящиеся в памяти компьютера.

В лабораторной работе файлы были сгенерированы при помощи стандартной библиотеки языка C#. Начальное значение генерируется так же и обозначает смещение относительно начала файла.

2.1.3. Критерий оценки

В качестве статистического критерия оценки случайности был выбран критерий "хи-квадрат". В первую очередь вычисляется статистика V

$$V = \frac{1}{n} \sum_{s=min}^{max} \left(\frac{Y_i^2}{p} \right) - n \quad (2.2)$$

где n - длина последовательности, $[min, max)$ - диапазон чисел, Y_i - количество повторений числа i , $p = 1/(max - min)$.

После этого вычисляется критерий - квантиль хи-квадрат (степень свободы $max - min - 1$) от значения V . Полученное значение предлагается интерпретировать следующим образом:

- <0.1 или >0.9 - последовательность считается недостаточно случайной.
- >0.1 и <0.9 - последовательность считается случайной.

2.2. Работа программы

Для демонстрации работы критерия предоставлены 10 полей ввода. Оценка производится для целых чисел от 0 до 9. Примеры работы программы приведены на рисунках 2.1, 2.2 и 2.3.

Лабораторная работа 3

Алгоритмический

5	40	437
0	45	204
5	47	731
1	90	831
0	45	166
5	46	582
0	33	469
0	63	382
0	29	857
4	83	132
0.3628805	0.1617299	0.6537551

Табличный

3	95	850
4	25	983
6	40	632
1	81	119
9	24	962
6	54	157
6	31	221
8	26	455
3	77	454
0	81	856
0.4003073	0.3583087	0.7272519

Ввод

0

0

0

0

0

0

0

0

0

0

0.9999999

Рис. 2.1 — Введена последовательность из одинаковых чисел

Лабораторная работа 3

Алгоритмический

5	49	237
0	70	178
0	62	274
1	64	858
2	93	668
1	73	759
6	24	856
5	60	164
9	50	461
2	64	754
0.6623116	0.8101003	0.7132221

Табличный

7	30	712
0	99	150
4	91	346
3	43	276
0	64	507
8	60	663
5	72	382
3	44	683
8	36	254
4	55	104
0.1229170	0.5091719	0.3886619

Ввод

1

2

3

4

5

6

7

8

9

9

0.00853239

Рис. 2.2 — Введена последовательность из разных чисел

Лабораторная работа 3

Алгоритмический

4	36	437
3	11	303
0	42	932
2	17	724
6	38	281
4	97	609
3	69	250
3	71	205
1	92	175
1	49	145
0,4977527	0,8561539	0,9108596

Табличный

4	34	884
5	27	450
9	37	676
8	87	287
9	11	499
8	10	811
5	81	901
3	26	868
1	33	905
8	67	502
0,5056083	0,6388648	0,6537551

Ввод

4
3
0
2
6
4
3
3
1
1
0,64951478

Рис. 2.3 — Введена случайная последовательность

3. Текст программы

В листинге 3.1 представлен фрагмент кода программы, отвечающий за генерирование последовательностей и вычисление критерия.

Листинг 3.1 — Вычисление функций распределений

```
1 class RndAlgGen: IRndGen
2 {
3     private int seed;
4     public RndAlgGen(int minN, int maxN, int s=0)
5     {
6         MinN = minN;
7         MaxN = maxN;
8         SetSid(s);
9     }
10
11     public override void SetSid(int s)
12     {
13         seed = s;
14     }
15
16     public override int Rand()
17     {
18         seed = seed * 16807 + 2147483647;
19         var res = ((uint)seed >> 16) % (MaxN - MinN) + MinN;
20         return (int)res;
21     }
22
23     public override int[] RandArr(int size)
24     {
25         int[] res = new int[size];
26         for (int i = 0; i < size; i++)
27             res[i] = Rand();
28         return res;
29     }
30 }
```

```

31
32 class RndTabGen : IRndGen
33 {
34     private int seed;
35     private StreamReader f;
36     private string fpath;
37
38     public static void FillFile(string fpath_, int min, int max
39 )
40 {
41     var rand = new Random();
42     using (StreamWriter sw = File.CreateText(fpath_))
43     {
44         for (int i = 0; i < (max - min) * 1000; i++)
45             sw.WriteLine(rand.Next(min, max));
46     }
47     public RndTabGen(int minN, int maxN, string fp, int s = 0)
48     {
49         MinN = minN;
50         MaxN = maxN;
51         fpath = fp;
52         SetSid(s);
53         var chi = new ChiSquared(1.0);
54     }
55
56     private void OpenFile(string fp)
57     {
58         fpath = fp;
59         if (f != null)
60             f.Close();
61         f = File.OpenText(fpath);
62         for (int i = 0; i < seed; i++)
63             f.ReadLine();

```



```

64     }
65
66     public override void SetSid(int s)
67     {
68         seed = s;
69         OpenFile(fpath);
70     }
71
72     public override int Rand()
73     {
74         string fileStr = f.ReadLine();
75         if (fileStr == null)
76         {
77             SetSid(0);
78             fileStr = f.ReadLine();
79         }
80         seed++;
81
82         var res = (UInt32.Parse(fileStr)) % (MaxN - MinN) + MinN;
83         return (int)res;
84     }
85
86     public override int[] RandArr(int size)
87     {
88         int[] res = new int[size];
89         for (int i = 0; i < size; i++)
90             res[i] = Rand();
91         return res;
92     }
93 }
94
95
96 class Crit
97 {

```

```

98  public int [] arr;
99  public Crit(int [] arr_)
100 {
101     arr = arr_;
102 }
103
104 public double Value(int min, int max)
105 {
106     int n = arr.Length;
107     double p = 1.0 / (max - min);
108     double acc = 0;
109     for (int i = min; i < max; i++)
110         acc += Math.Pow(arr.Count(x => x == i), 2) / p;
111     acc = acc / (double)n - n;
112
113     double res = ChiSquared.CDF(max - min - 1, acc);
114     return res;
115 }
116 }

```