1830

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»

ОТЧЕТ

По лабораторной работе №3

По курсу: «Моделирование»

Тема: «Генерация псевдослучайных последовательностей»

Студент: Пронин А. С.

Группа: ИУ7-72Б

Преподаватель: Рудаков И. В.

Оценка: _____

Москва

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.

1 Отчет

1.1 Теория

Существует три метода получения последовательности случайных чисел:

- аппаратный
- табличный
- алгоритмический

В данной лабораторной работе реализованы два последних из них

1.1.1 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

$$X_{n+1} = (aX_n + c) \bmod m, \ n \ge 1 \tag{1.1}$$

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.

1.1.2 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.

1.1.3 Критерий оценки

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

$$V = \frac{1}{n} \sum_{s=min}^{max} (\frac{Y_i^2}{p}) - n \tag{1.2}$$

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 Текст программы

В листинге 2.1 представлен код программы, отвечающий за генерирование последовательностей и вычисление критерия оценки.

Листинг 2.1: Код программы

```
using System;
2 using System.10;
3 using Accord. Math;
  using MathNet. Numerics. Distributions;
 using System. Collections. Generic;
 using System.Linq;
 using System.Text;
  using System. Threading. Tasks;
 namespace lab3
  {
11
      public abstract class IRndGen
12
           public int MaxN;
           public int MinN;
15
           public abstract void SetSid(int s);
16
           public abstract int Rand();
17
           public abstract int[] RandArr(int size);
      }
19
20
      class RndAlgGen: IRndGen
21
      {
22
           private int seed;
23
           public RndAlgGen(int minN, int maxN, int s=0)
24
25
               MinN = minN;
26
               MaxN = maxN;
               SetSid(s);
28
           }
29
30
           public override void SetSid(int s)
31
               seed = s;
33
```

```
}
34
35
           public override int Rand()
36
           {
37
               seed = seed * 16807 + 2147483647;
38
               var res = ((uint)seed >> 16) \% (MaxN - MinN) + MinN;
39
               return (int)res;
40
           }
42
           public override int[] RandArr(int size)
43
44
                int[] res = new int[size];
45
                for (int i = 0; i < size; i++)
46
                    res[i] = Rand();
47
                return res;
48
           }
49
      }
50
51
      class RndTabGen: IRndGen
52
      {
53
           private int seed;
54
           private StreamReader f;
55
           private string fpath;
56
57
           public static void FillFile(string fpath_, int min, int
58
     max)
           {
59
               var rand = new Random();
60
               using (StreamWriter sw = File.CreateText(fpath ))
61
62
                    for (int i = 0; i < (max - min) * 1000; i++)
63
                        sw.WriteLine(rand.Next(min, max));
               }
65
66
           public RndTabGen(int minN, int maxN, string fp, int s =
67
     0)
           {
68
               MinN = minN;
69
               MaxN = maxN;
70
               fpath = fp;
71
                SetSid(s);
72
```

```
var chi = new ChiSquared(1.0);
73
           }
74
           private void OpenFile(string fp)
76
           {
77
                fpath = fp;
78
                if (f != null)
                     f. Close();
                f = File.OpenText(fpath);
81
                for (int i = 0; i < seed; i++)
82
                     f.ReadLine();
83
           }
85
           public override void SetSid(int s)
86
           {
87
                seed = s;
88
                OpenFile (fpath);
           }
90
91
           public override int Rand()
92
93
                string fileStr = f.ReadLine();
                if (fileStr = null)
95
                {
96
                     SetSid(0);
97
                     fileStr = f.ReadLine();
98
                seed++;
100
101
                var res = (UInt32.Parse(fileStr)) % (MaxN - MinN) +
102
     MinN;
                return (int)res;
           }
104
105
           public override int[] RandArr(int size)
106
                int[] res = new int[size];
108
                for (int i = 0; i < size; i++)
109
                     res[i] = Rand();
110
                return res;
111
           }
112
```

```
}
113
114
       class Crit
115
       {
116
            public int[] arr;
117
            public Crit(int[] arr_)
118
119
                 arr = arr_;
            }
121
122
            public double Value(int min, int max)
123
124
                 int n = arr.Length;
125
                double p = 1.0 / (max - min);
126
                 double acc = 0;
127
                 for (int i = min; i < max; i++)
128
                     acc += Math.Pow(arr.Count(x \Rightarrow x == i), 2) / p;
                 acc = acc / (double)n - n;
130
131
                double res = ChiSquared.CDF(max - min - 1, acc);
132
                 return res;
133
            }
       }
135
136 }
```

3 Результаты

Для демонстрации работы критерия предоставленны 10 полей ввода. Оценка производятся для целых чисел от 0 до 9. Примеры работы программы приведены на рисунках 3.1–3.4.

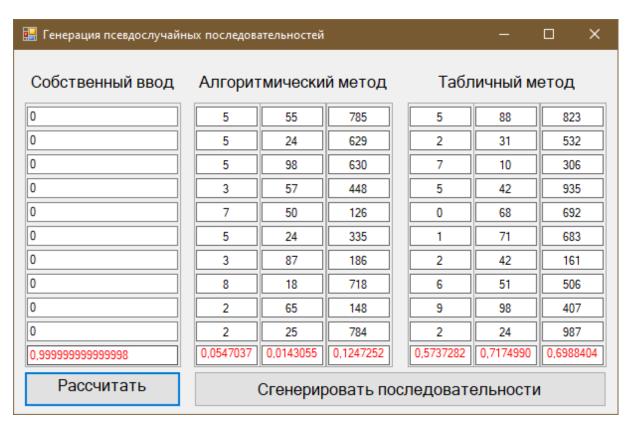


Рис. 3.1: Последовательность из одинаковых чисел

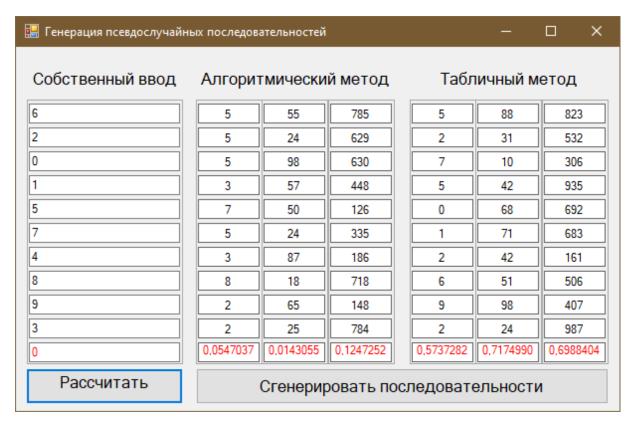


Рис. 3.2: Последовательность из чисел от 0 до 9

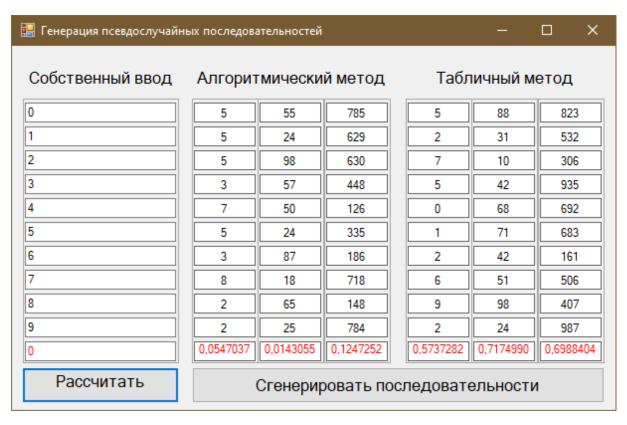


Рис. 3.3: Возрастающая последовательность

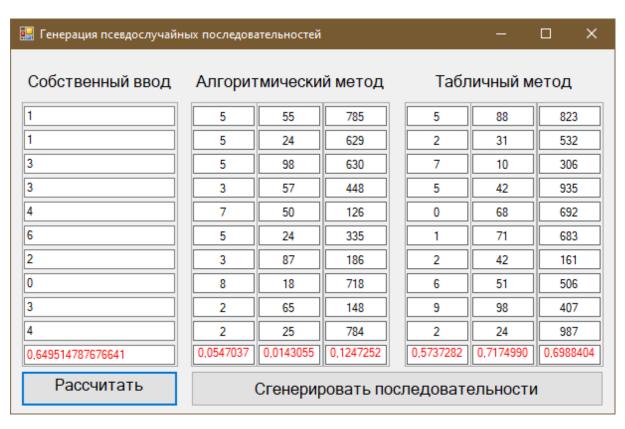


Рис. 3.4: Введена случайная последовательность