

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»

Кафедра электронных вычислительных машин

Отчёт
по лабораторной работе № 1
«Формирование последовательности случайных чисел с заданным законом
распределения»

Выполнили
ст. гр. 950502

Сякачëв П.В.

Проверила:

Герман Ю.О.

Минск 2022

Цель работы:

Изучить основные способы создания последовательностей случайных чисел с заданными законами распределения вероятности.

Краткие теоретические сведения:

Случайные и псевдослучайные числа – числа, последовательность появления которых обладает теми или иными статистическими закономерностями. Различают случайные числа, генерируемые каким-либо стохастическим устройством, и псевдослучайные числа, генерация осуществляется с помощью арифметических алгоритмов.

Генераторы случайных чисел (ГСЧ) по способу получения чисел:

- физические;
 - монета;
 - игральная кость;
- табличные;
- алгоритмические (псевдослучайные);
 - метод серединных квадратов (см. рис. 1);
 - метод перемешивания (см. рис. 2);
 - метод серединных произведений (см. рис. 3);
 - линейный конгруэнтный метод (алгоритм Лемера).

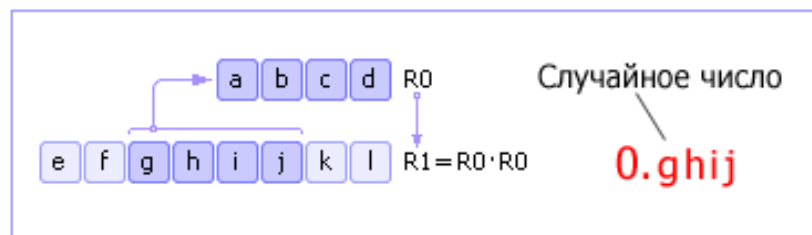


Рисунок 1 – Генерация чисел методом серединных квадратов

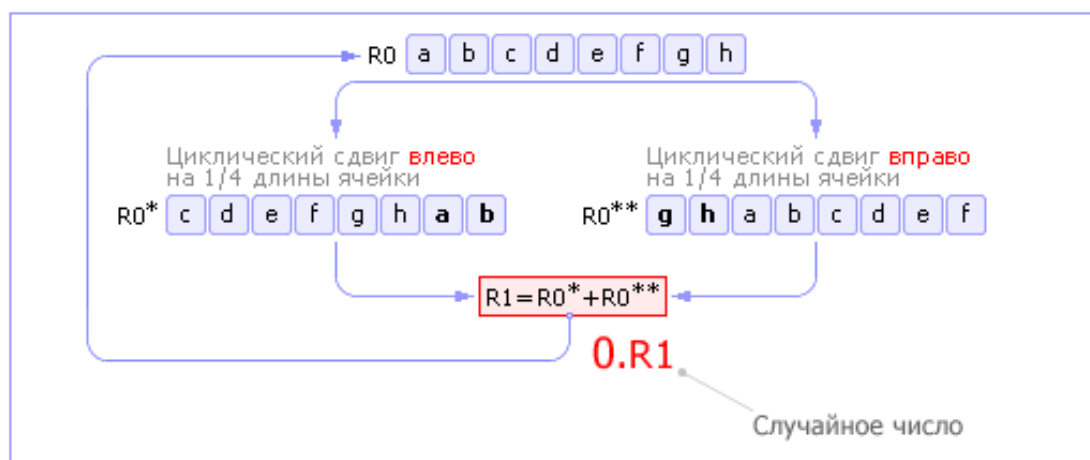


Рисунок 2 – Генерация чисел методом перемешивания

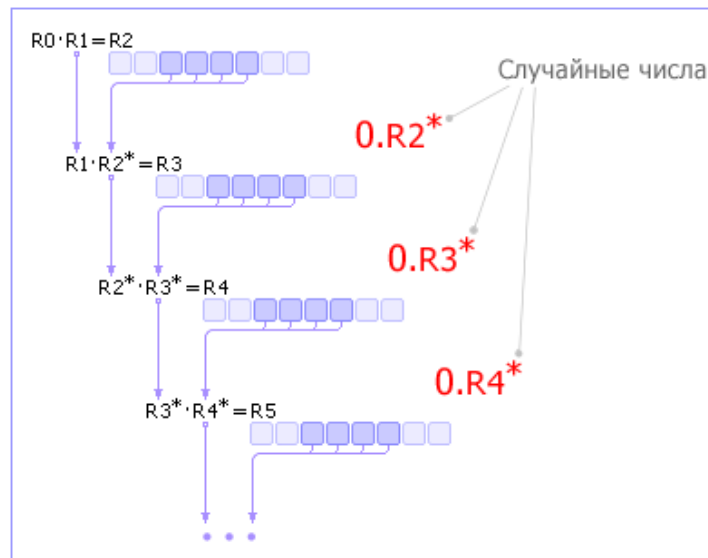


Рисунок 3 – Генерация чисел методом срединных произведений

Самый простой метод генерации случайных чисел – алгоритм Лемера. Выраженный в символьном виде алгоритм Лемера представляет собой следующее выражение: $X(i) = a * X(i-1) \bmod m$

От качества работы ГСЧ зависит качество работы всей системы и точность результатов. Поэтому случайная последовательность, порождаемая ГСЧ, должна удовлетворять целому ряду критериев.

Осуществляемые проверки бывают двух типов:

- проверки на равномерность распределения (критерий «хи-квадрат»);
- проверки на статистическую независимость (проверка на частоту появления цифры в последовательности).

Качество формируемой последовательности по критерию хи-квадрат в рамках данной лабораторной работы заключается в проверке факта, что «хи-квадрат» табличный, вычисленный в Excel, больше «хи-квадрата» расчётного, полученный по формуле, записанный в коде программы ниже.

Ход работы:

```
namespace Lemer
{
    public class Lehmer
    {
        private const int a = 11817;
        private const int m = 36471;

        public int Next()
        {
            int z1 = ((a * Program.ri) % m);

            return z1;
        }
    }
    public class Method
    {
        private const double first = 4598754, second = 8653147;

        public static void Run()
        {
            double[] random = new double[20];
            random[0] = first;
            random[1] = second;
            for (int i = 0, j = 0; i < 20; i++)
            {
                double temp = random[j] * random[j + 1];
                string nextrand = temp.ToString();
                int poz = (nextrand.Length / 4) % 4 == 3 ? nextrand.Length / 4 + 1 :
nextrand.Length / 4;
                int len = (nextrand.Length / 4) % 4 == 1 || (nextrand.Length / 4) %
4 == 2 ?
                    nextrand.Length / 2 + 1 : nextrand.Length / 2;
                nextrand = nextrand.Substring(poz, len);
                random[i] = Convert.ToDouble(nextrand);
                if (i > 1) j++;
            }
            for (int i = 0; i < random.Length; i++)
            {
                Console.WriteLine(" " + random[i]);
            }
        }
    }
    class Program
    {
        public static int ri = 7;

        static void Main(string[] args)
        {
            int e = 20;
            int[] counts = new int[e];
            int iter_count = 10000;
            int actual_part = iter_count / e;

            Lehmer lehmer = new();
            for (int i = 0; i < iter_count; ++i)
            {
                ri = lehmer.Next();
                int z1 = ri % e;
                counts[z1]++;
            }
            for (int i = 0; i < e; ++i)
```

```

    {
        Console.WriteLine("" + counts[i]);
    }
    double x;
    double summ = 0;
    for (int i = 0; i < counts.Length; ++i)
    {
        summ += Math.Pow((actual_part - counts[i]), 2) / actual_part;
    }
    x = summ / e;
    Console.WriteLine("X = " + x + Environment.NewLine);
    Method.Run();
}
}
}

```

Задача 1:

При $e = 20$, $\chi^2_{\text{теор}} = 10,117$, $\chi^2_{\text{эксп}} = 0,793$

При $e = 60$, $\chi^2_{\text{теор}} = 42,339$, $\chi^2_{\text{эксп}} = 0,773$

Результат выполнения программы:

Частотное распределение при $e = 20$	Частотное распределение при $e = 70$
512	139
496	140
504	144
500	147
497	134
508	137
508	145
516	144
495	144
489	138
488	150
498	138
487	156
499	137
499	149
487	145
506	144
509	144
488	151
514	138
X = 0,16599999999999998	144
	150
	144
	146
	143
	150
	137
	139
	151
	145
	153
	141
	146
	143
	140
	150
	142
	132
	142
	141
	143
	139
	141
	143
	147
	132
	141
	143
	141
	151
	141
	X = 0,18953722334004028
	143

Задание 2:

Результат выполнения программы для генерации 20 чисел методом срединных произведений

```
3694378
7995907
9902910
2747389
7146001
2844541
7092830
5845741
2847137
3625493
2275263
950079
679595
689380
992011
725431
355317
579666
651841
500651
```

Вывод:

В ходе выполнения лабораторной работы №1 были изучены основные способы создания последовательностей случайных чисел, а также реализована программа для генерации последовательности случайных чисел.