

Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО РГРТУ

Кафедра вычислительной и прикладной математики

Лабораторная работа №1  
“Изучение базовых генераторов псевдослучайных чисел”

Выполнил:  
Студент группы №843  
Редько С.В.

Проверил:  
Овечкин Г.В.

Рязань 2022

## Вариант 10

### Задание:

Составить и отладить программу (подпрограмму) генерирования псевдослучайных чисел с равномерным распределением на интервале  $[0;1)$ . Для заданных объема выборки и числа участков разбиения интервала  $[0;1)$  построить гистограмму частот и статистическую функцию распределения, получить программным способом оценки математического ожидания, дисперсии, второго и третьего моментов. Выполните анализ полученных результатов.

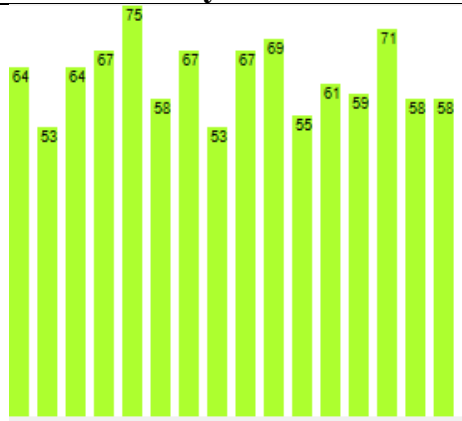
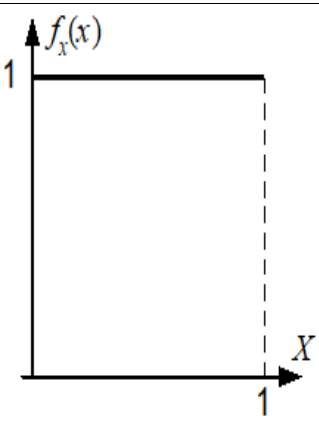
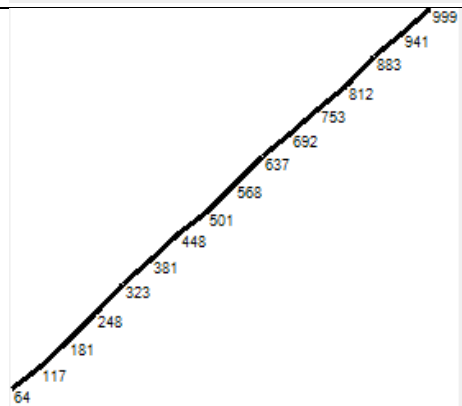
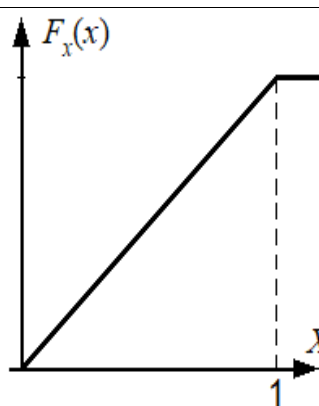
№ вар.	Тип датчика	Начальные данные	Объем выборки	Число участков разбиения
10.	Аддитивный, формула (2.8)	$Y_1 = 4091, Y_2 = m - 5$ $m = 4096 * 4$	1000	16

### Решение:

После генерации последовательности оцениваем ее характеристики:

	Полученное	Ожидаемое
Математическое ожидание	0,497	0,5
Дисперсия	0,082	0,0833
2-й момент	0,323	0,333
3-й момент	0,245	0,25

Построим гистограмму частот и график функции распределения:

	Полученное	Ожидаемое
Плотность распределения		
Функция распределения		

## Приложение:

```
/// <summary>
/// МатОжидание
/// </summary>
/// <param name="parArr">Последовательность</param>
/// <returns></returns>
private double GetMx(double[] parArr)
{
    double sum = 0;
    for (int i = 0; i < parArr.Length; i++)
        sum += parArr[i];
    return sum / parArr.Length;
}
/// <summary>
/// Дисперсия
/// </summary>
/// <param name="parArr">Последовательность</param>
/// <returns></returns>
private double GetD(double[] parArr)
{
    double sum = 0;
    for (int i = 0; i < parArr.Length; i++)
        sum += Math.Pow(parArr[i] - GetMx(parArr), 2);
    return sum / parArr.Length;
}
/// <summary>
/// Второй момент
/// </summary>
/// <param name="parArr">Последовательность</param>
/// <returns></returns>
private double Get2Moment(double[] parArr)
{
    double sum = 0;
    for (int i = 0; i < parArr.Length; i++)
        sum += Math.Pow(parArr[i], 2);
    return sum / parArr.Length;
}
/// <summary>
/// Третий момент
/// </summary>
/// <param name="parArr">Последовательность</param>
/// <returns></returns>
private double Get3Moment(double[] parArr)
{
    double sum = 0;
    for (int i = 0; i < parArr.Length; i++)
        sum += Math.Pow(parArr[i], 3);
    return sum / parArr.Length;
}
/// <summary>
/// Генерация псевдослучайных чисел с равновероятным распределением
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void generate1_Click(object sender, EventArgs e)
{
    listNumber.Text = "";
    numericUpDownCountIntervals.Enabled = true;

    Y0 = (int)numericUpDownY1.Value;
    M = (int)numericUpDownM.Value;
    Y1 = (int)numericUpDownY2.Value;
    CountNumbers = (int)numericUpDownN.Value;
    double[] arr = new double[CountNumbers];
    CountIntervals = (int)numericUpDownCountIntervals.Value;
```

```

for (int i = 0; i < CountNumbers; i++)
{
    arr[i] = Random();
    if (i < 30)
        listNumber.Text += arr[i].ToString() + "\n";
}
Array.Sort(arr);
array = arr;

textBoxMx.Text = GetMx(arr).ToString();
textBoxD.Text = GetD(arr).ToString();
textBox2Moment.Text = Get2Moment(arr).ToString();
textBox3Moment.Text = Get3Moment(arr).ToString();

double min = arr.Min();
double max = arr.Max();
double lengthPart = (max - min) / CountIntervals;
//плотность вероятности
int[] a = new int[CountIntervals];
//функция распределения
int[] b = new int[CountIntervals];

for (int i = 0; i < CountIntervals; i++)
{
    a[i] = 0;
    for (int j = 0; j < arr.Length; j++)
        if (arr[j] >= min + lengthPart * i && arr[j] < min + lengthPart * (i + 1))
            a[i]++;
}
b[0] = a[0];
for (int i = 1; i < CountIntervals; i++)
    b[i] = b[i - 1] + a[i];
DrawHistogramm(a, CountIntervals);
DrawGraph(b, CountIntervals);
}

/// <summary>
/// Гистограмма
/// </summary>
/// <param name="parArr"></param>
private void DrawHistogramm(int[] parArr, int parCount)
{
    int intervalColumn = 5;
    int widthColumn = panel1.Width / parCount - intervalColumn;

    Graphics gPanel = panel1.CreateGraphics();

    gPanel.FillRectangle(new SolidBrush(Color.White),
        new Rectangle(0, 0, panel1.Width, panel1.Height));

    for (int i = 0; i < parCount; i++)
    {
        gPanel.FillRectangle(new SolidBrush(Color.GreenYellow),
            new Rectangle((widthColumn + intervalColumn) * i,
                panel1.Height - (int)(parArr[i] * (double)panel1.Height /
                    parArr.Max()),
                widthColumn,
                panel1.Height));
        gPanel.DrawString(parArr[i].ToString(),
            new Font("Arial", 7),
            new SolidBrush(Color.Black),
            (widthColumn + intervalColumn) * i,

```

```

        panel1.Height - (int)(parArr[i] * (double)panel1.Height /
        parArr.Max()));
    }
}
/// <summary>
/// График
/// </summary>
/// <param name="parArr"></param>
private void DrawGraph(int[] parArr, int Count)
{
    int intervalColumn = 5;
    int widthColumn = panel1.Width / Count - intervalColumn;

    Graphics gPanel = panel2.CreateGraphics();
    gPanel.FillRectangle(new SolidBrush(Color.White),
    new Rectangle(0, 0, panel2.Width, panel2.Height));

    for (int i = 0; i < Count; i++)
    {
        if (i != 0)
            gPanel.DrawLine(new Pen(Color.Black, 3),
            new PointF((widthColumn + intervalColumn) * i, panel1.Height -
            (int)(parArr[i] * panel1.Height / parArr.Max())),
            new PointF((widthColumn + intervalColumn) * (i - 1), panel1.Height -
            (int)(parArr[i - 1] * panel1.Height / parArr.Max())));
        gPanel.DrawString(parArr[i].ToString(),
        new Font("Arial", 7),
        new SolidBrush(Color.Black),
        (widthColumn + intervalColumn) * i,
        panel1.Height - (int)(parArr[i] * (double)panel1.Height / parArr.Max()));
    }
}
/// <summary>
/// Случайная величина
/// </summary>
/// <returns></returns>
public double Random()
{
    int y = (Y0 + Y1) % M;
    Y0 = Y1;
    Y1 = y;
    return (double)y / M;
}

```