



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 № 1

Тема: Изучение функций распределения и плотности
распределения

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

Разработать программу для построения графиков распределений:

- Равномерное распределение
- Нормальное распределение

Реализовать возможность выбора закона распределения и его параметров при помощи графического интерфейса.

2. Результаты

Примеры построения графиков равномерного и нормального распределения приведены на рисунках 2.1 и 2.2.

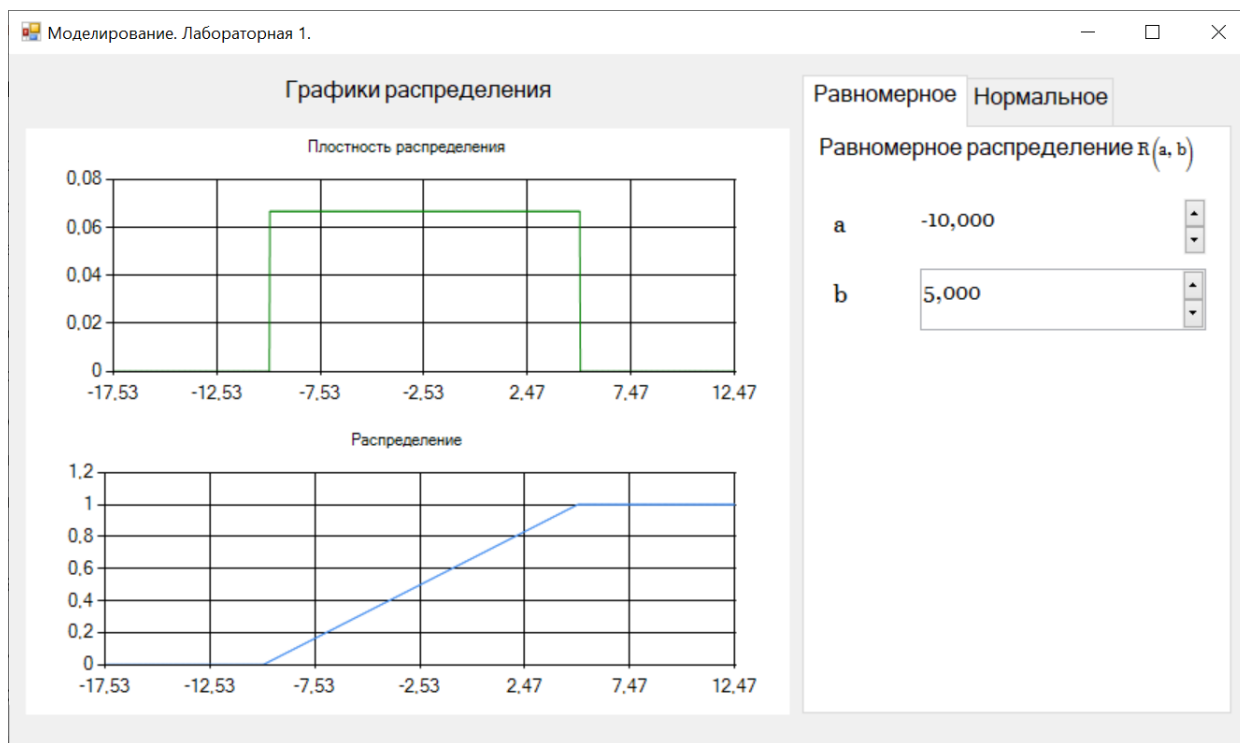


Рис. 2.1 — Графики распределения и плотности распределения для равномерного распределения

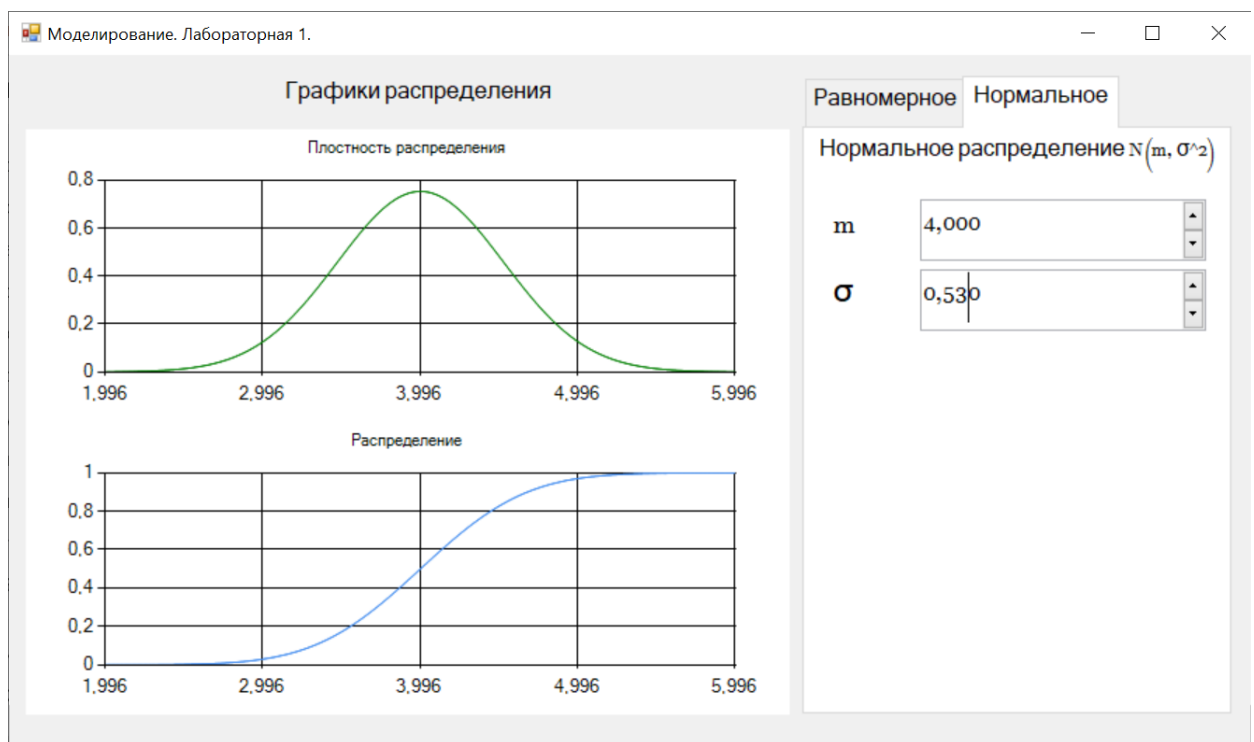


Рис. 2.2 — Графики распределения и плотности распределения для нормального распределения

3. Текст программы

В листинге 3.1 представлен фрагмент кода программы, отвечающий за вычисление значений функций для различных распределений.

Листинг 3.1 — Вычисление функций распределений

```
1 public abstract class IDist
2 {
3     public abstract double DistVal(double x);
4     public abstract double DensVal(double x);
5 }
6
7 public class EvalDist: IDist
8 {
9     private double a;
10    private double b;
11    private double p;
12
13    public EvalDist(double a_, double b_)
14    {
15        a = a_;
16        b = b_;
17        p = 1 / (b - a);
18    }
19
20    public override double DistVal(double x)
21    {
22        return Math.Max(Math.Min((x - a)*p, 1), 0);
23    }
24
25    public override double DensVal(double x)
26    {
27        if (a <= x && x <= b) return p;
28        else return 0;
29    }
30 }
```

```

31
32 public class NormDist : IDist
33 {
34     private double m;
35     private double sig;
36
37     public NormDist(double m_, double sig_)
38     {
39         m = m_;
40         sig = sig_;
41     }
42
43     public override double DistVal(double x)
44     {
45         return (1 + SpecialFunctions.Erf((x - m) / (sig * Math.
46             Sqrt(2)))) / 2;
47     }
48
49     public override double DensVal(double x)
50     {
51         return (1 / (sig * Math.Sqrt(2 * Math.PI))) *
52             Math.Exp(-(x-m) * (x-m) / 2 / sig / sig);
53     }
54 }

```