Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе № 2

“Формирование случайных чисел с заданным распределением”

Выполнил: Русель А.А.

Проверил:  Алексеев И.Г.

МИНСК 2023

**1 ЦЕЛЬ**

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

**2 ЗАДАНИЕ**

Написать приложение на языке С# для моделирования генераторов псевдослучайных чисел. Требуется реализовать выбрать два наиболее успешных алгоритмов генерации из предыдущей лабораторной работы. В данной работе используются следующие генераторы:

1. Генератор на основе алгоритма Лемера;
2. Генератор на основе метода серединных произведений.

Подобрать параметры для генераторов с целью получения наилучших результатов для каждого генератора (для выборок N = {10000, 500000, 100000000}).

Вывести гистограмму, матожидание, корреляцию и результаты распределения для лучшего случая.

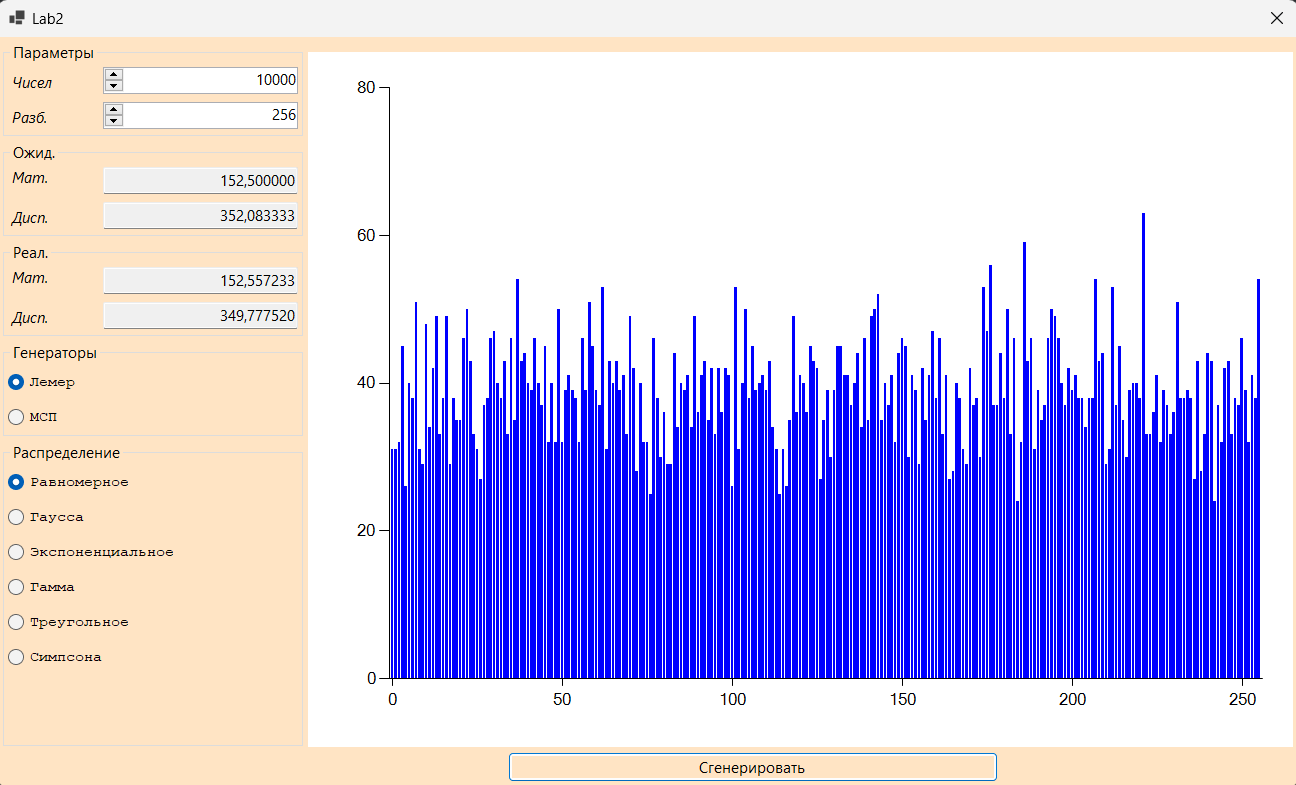
**3 ХОД РАБОТЫ**

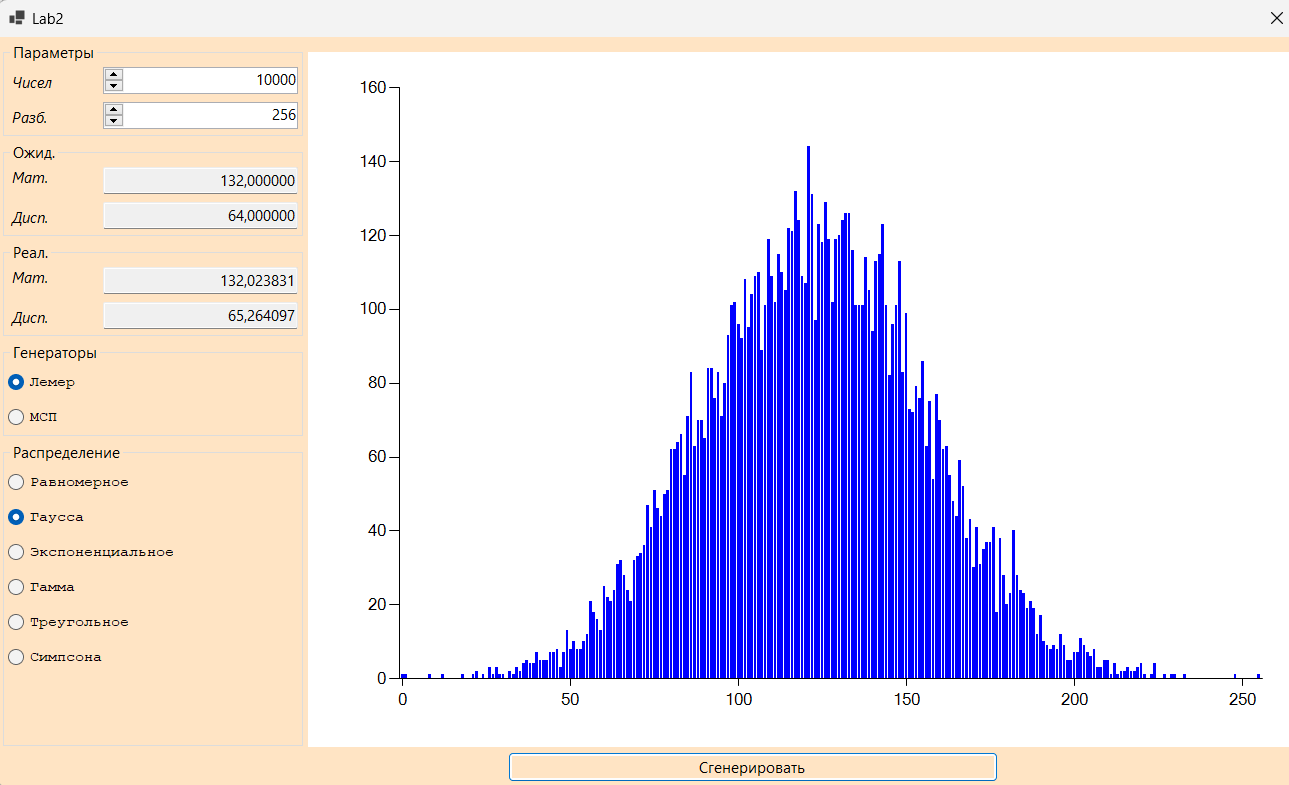
**3.1 Генератор Лемера**

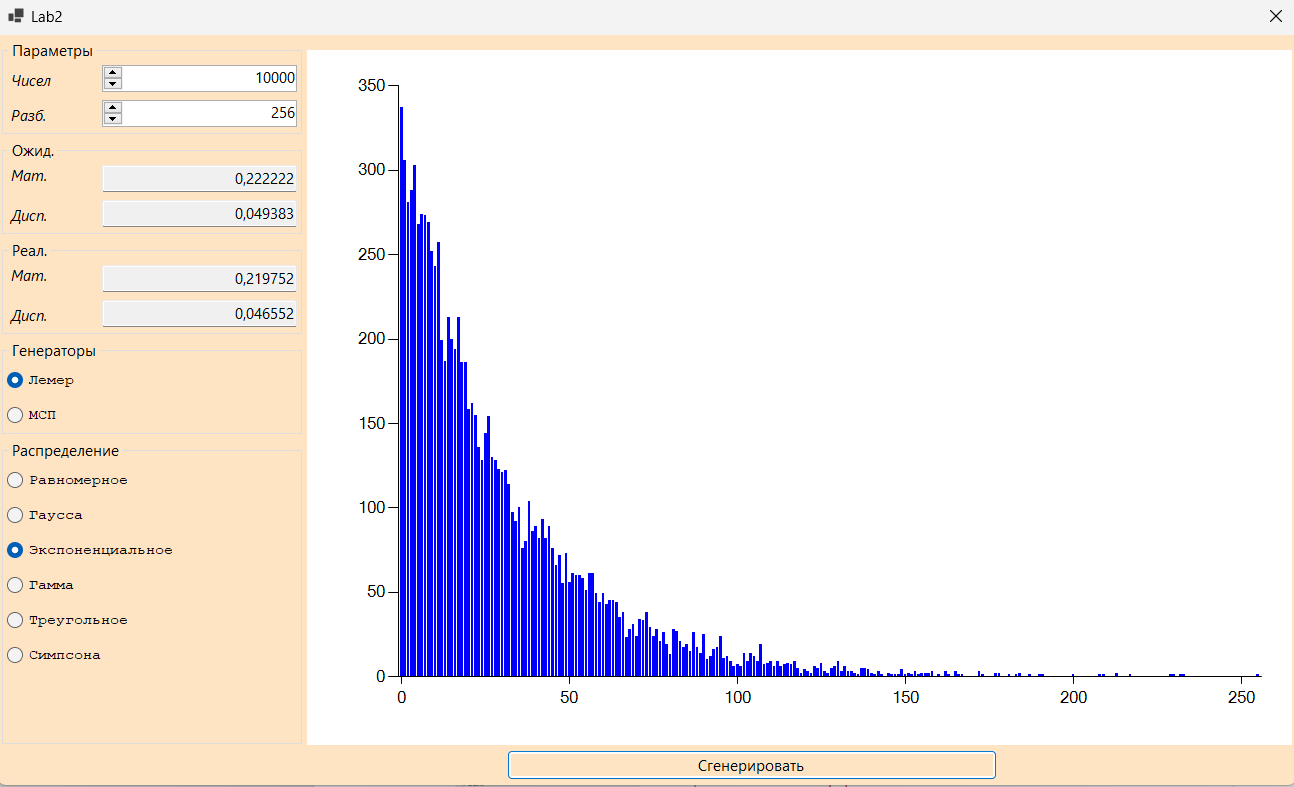
Генератор на основе алгоритма Лемера. В данном генераторе используются следующие коэффициенты:

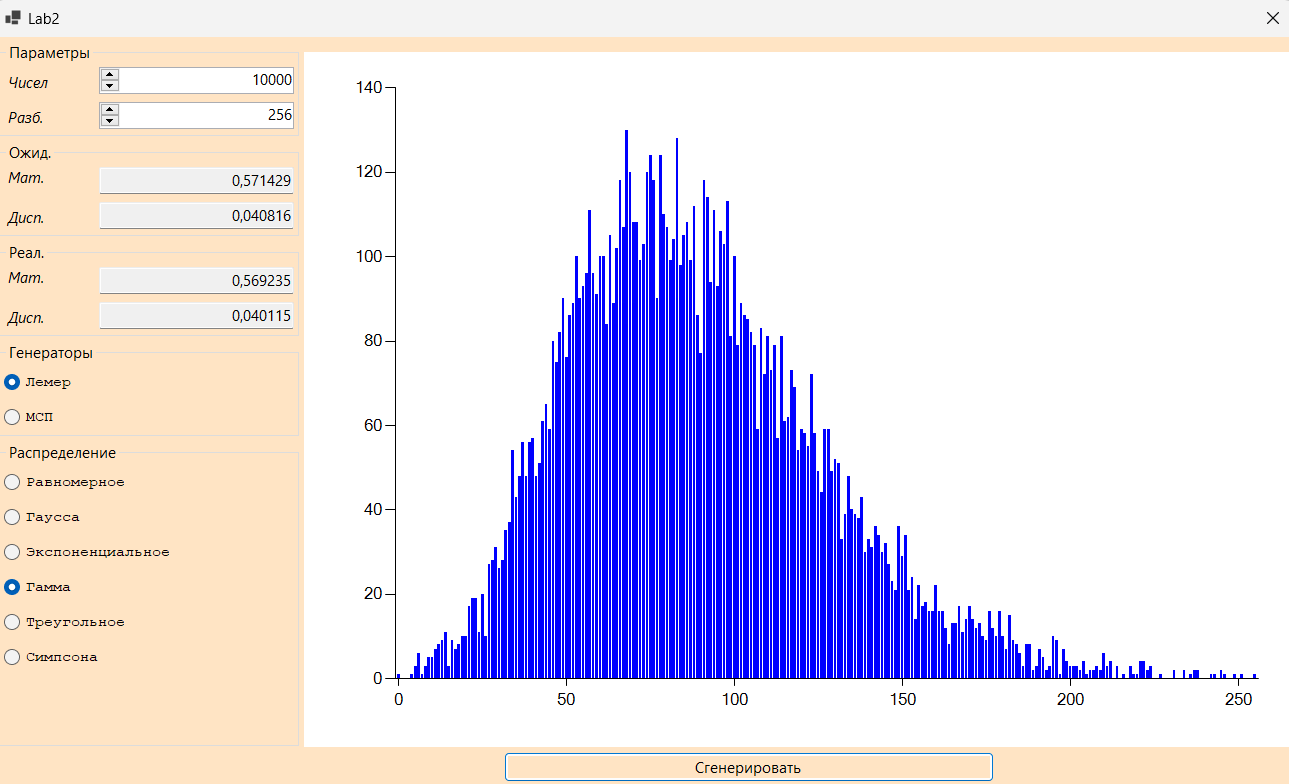
1. A = 16\_769\_023;
2. M = ulong.MaxValue - 1\_678;
3. X = 5\_542\_985\_019\_387.

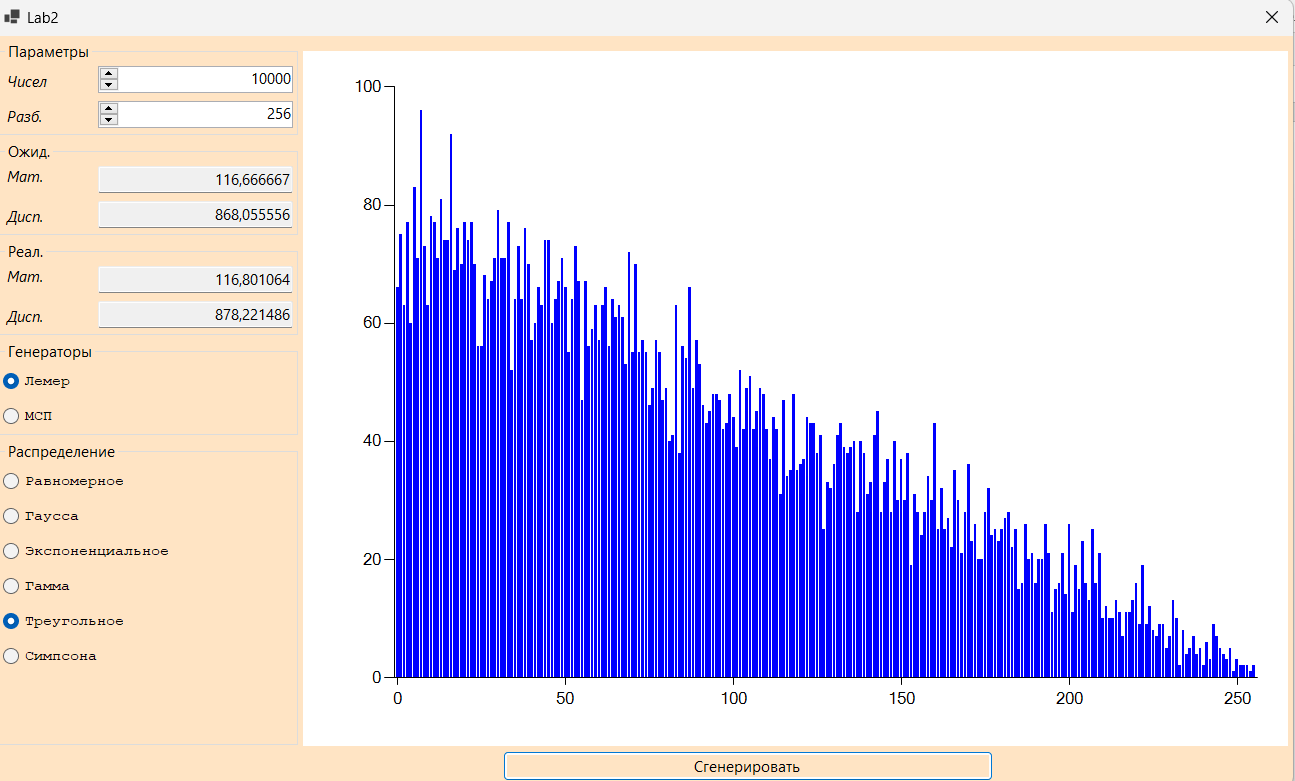
Генератор с выборкой равной n = 10000

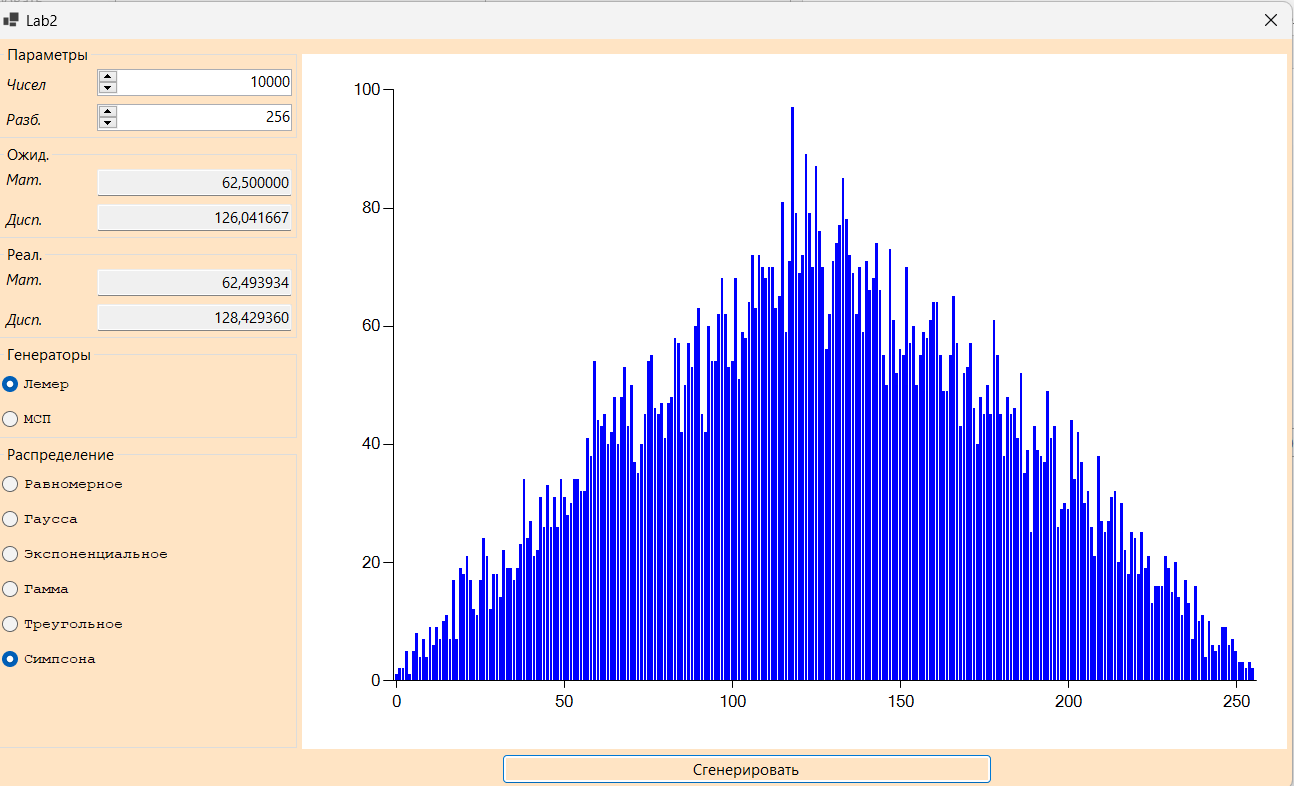




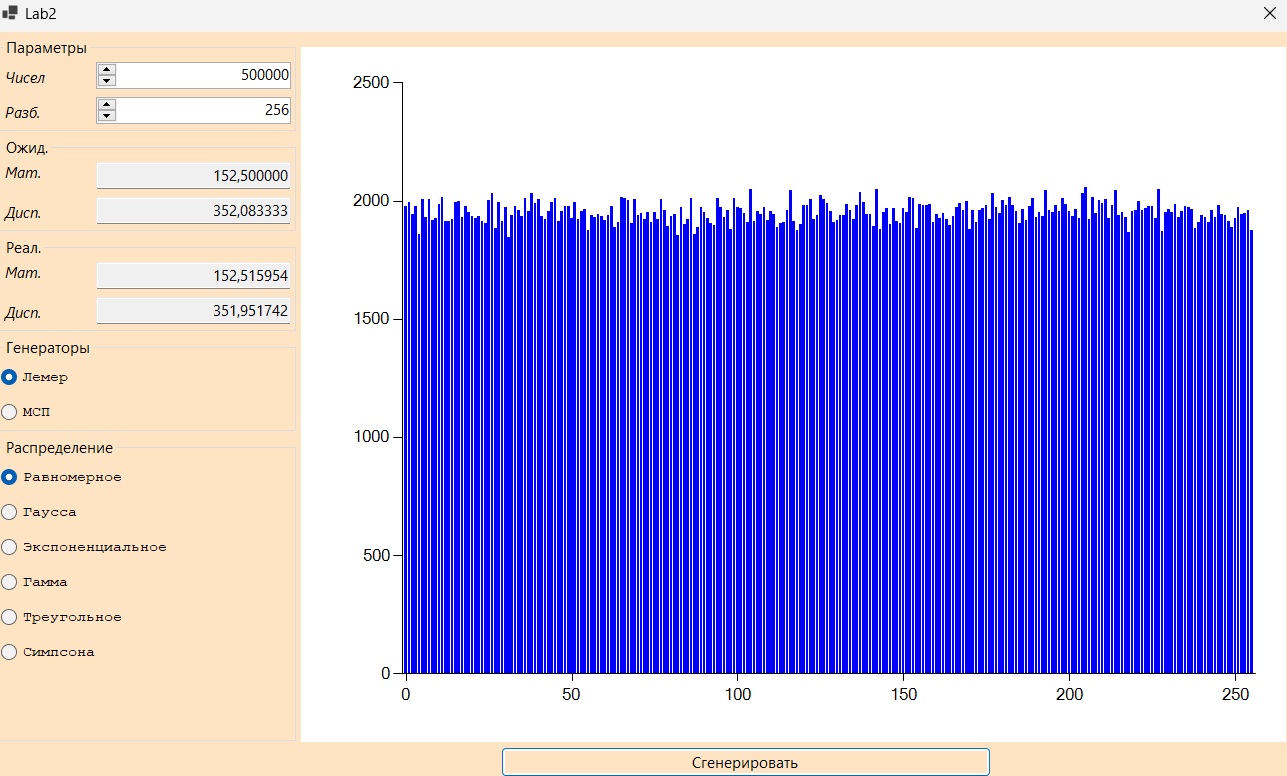


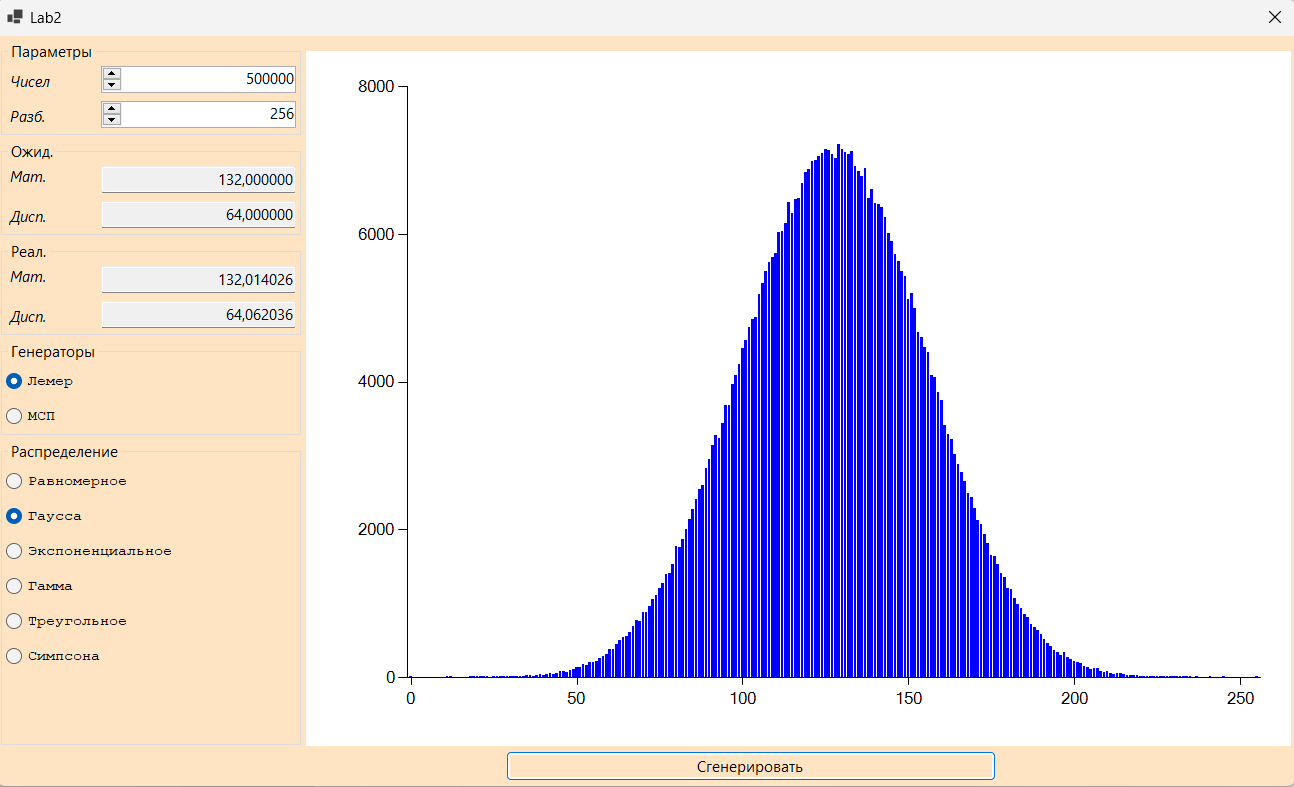


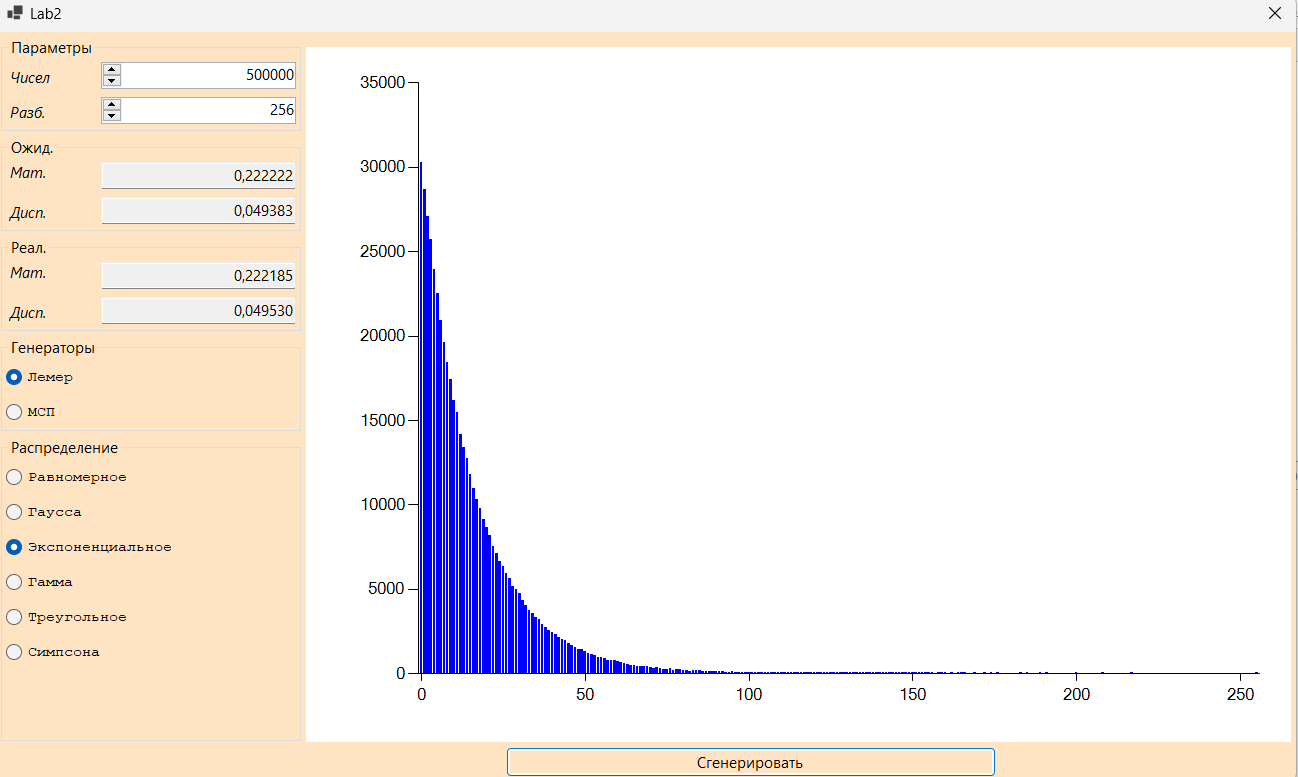


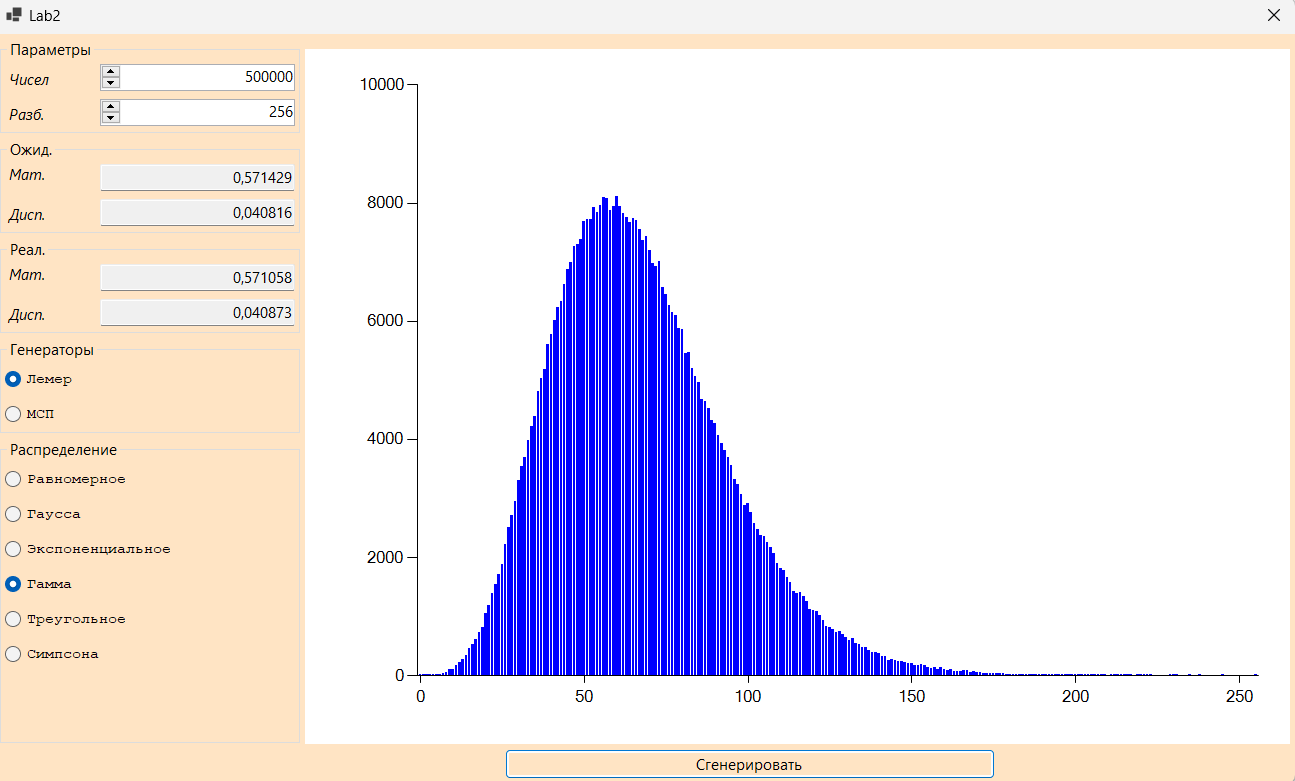


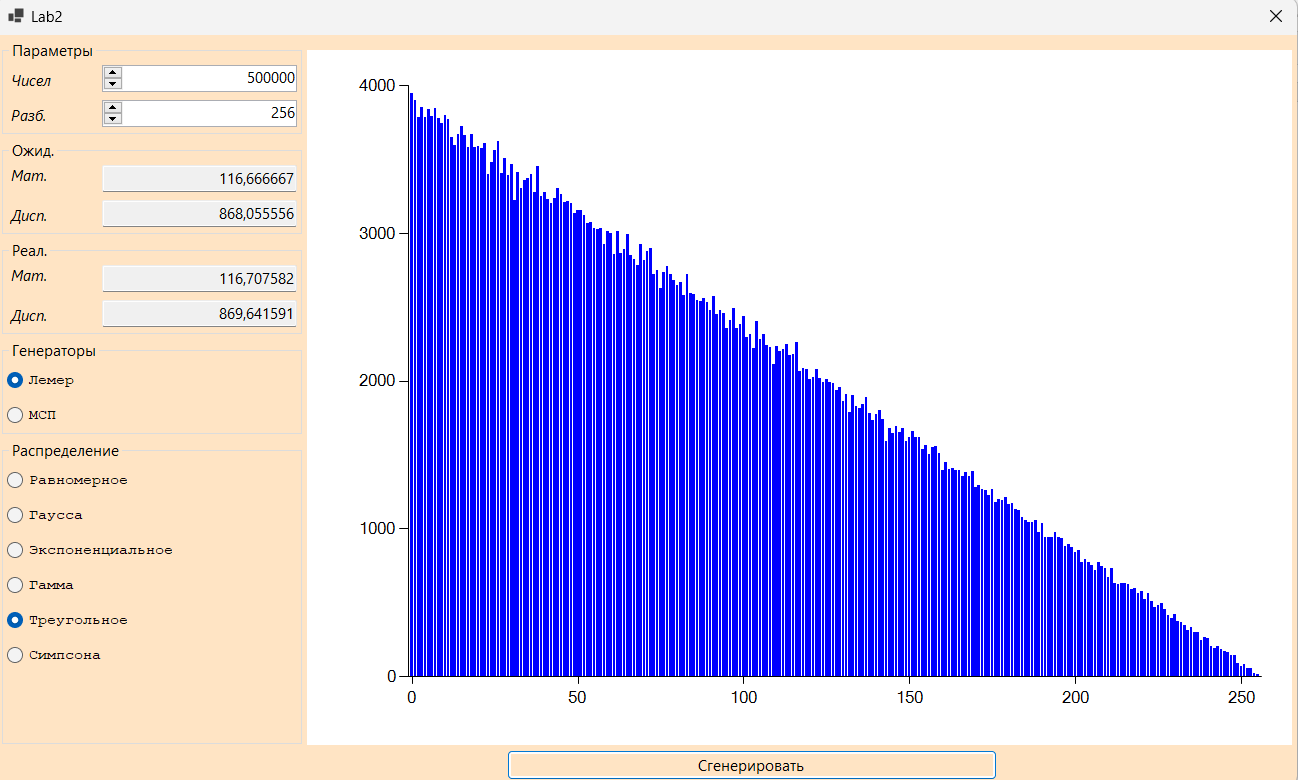
Генератор с выборкой равной n = 500000

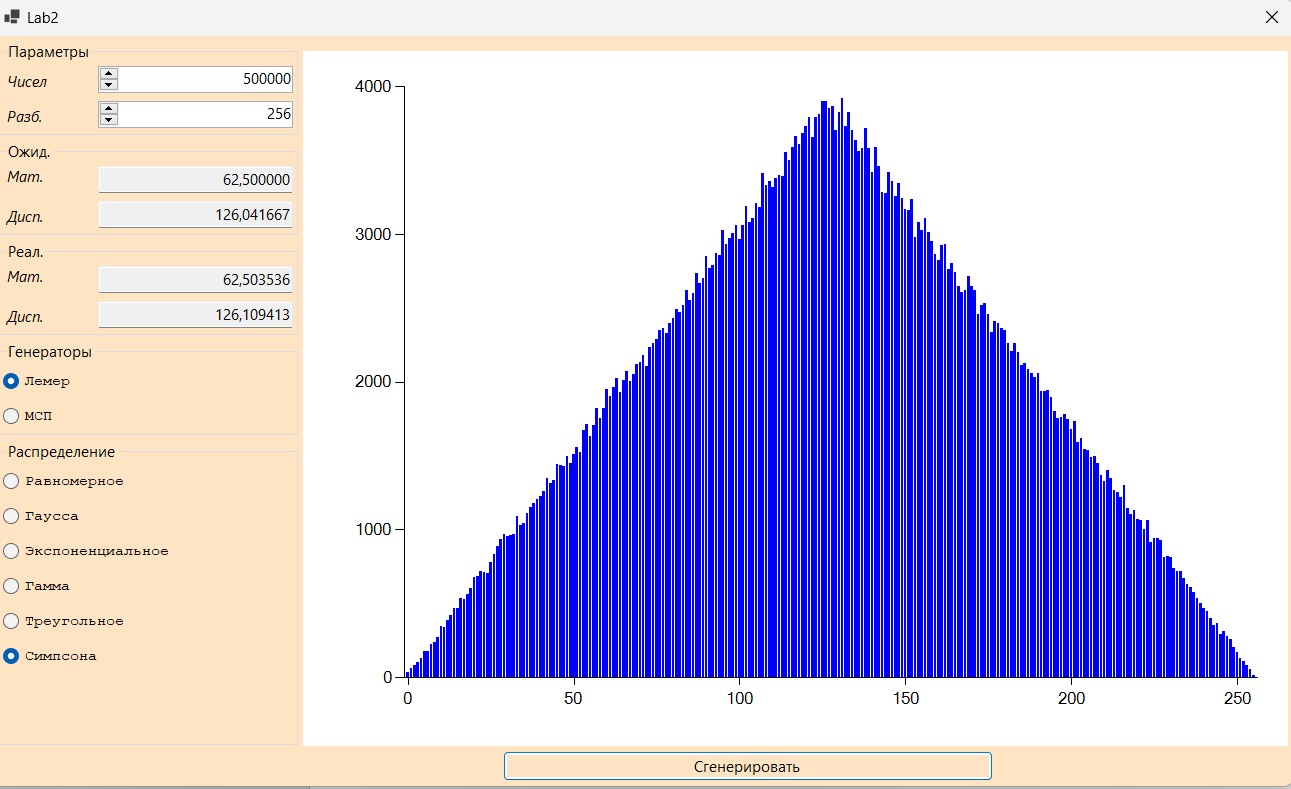




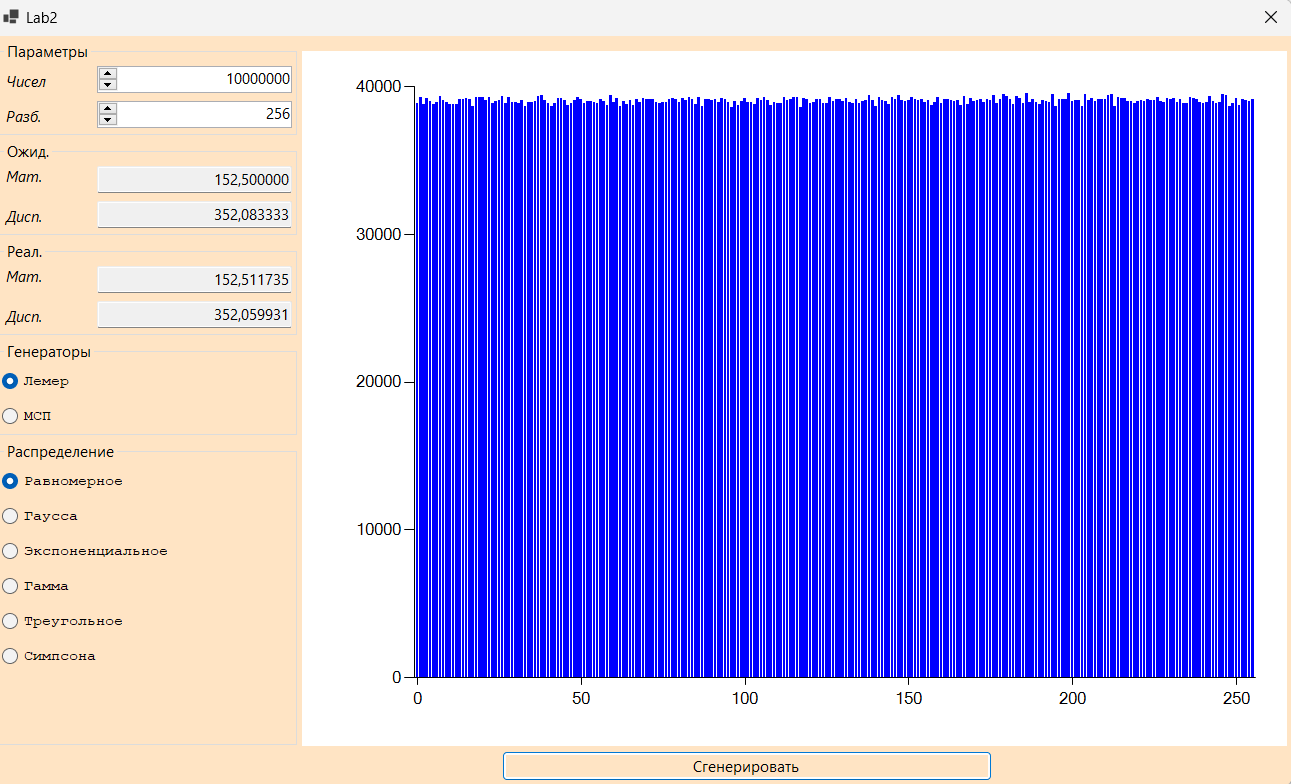


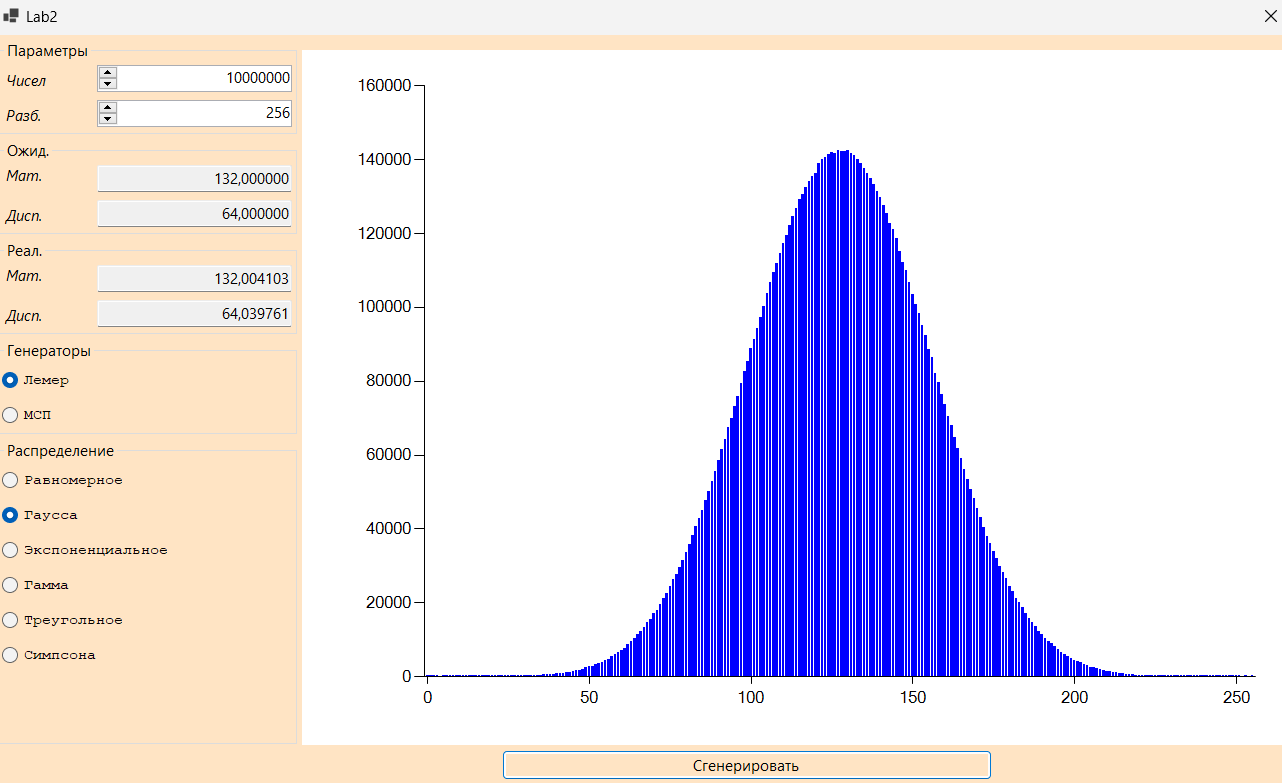


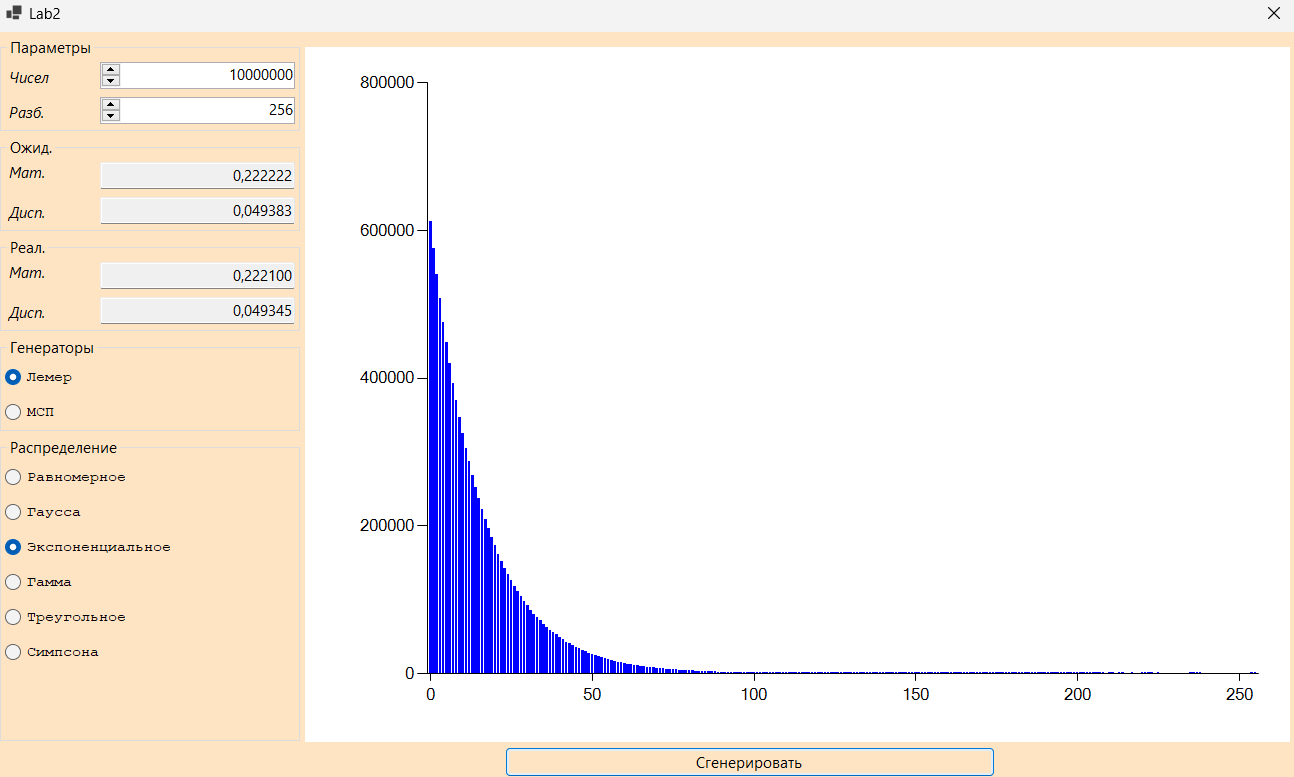


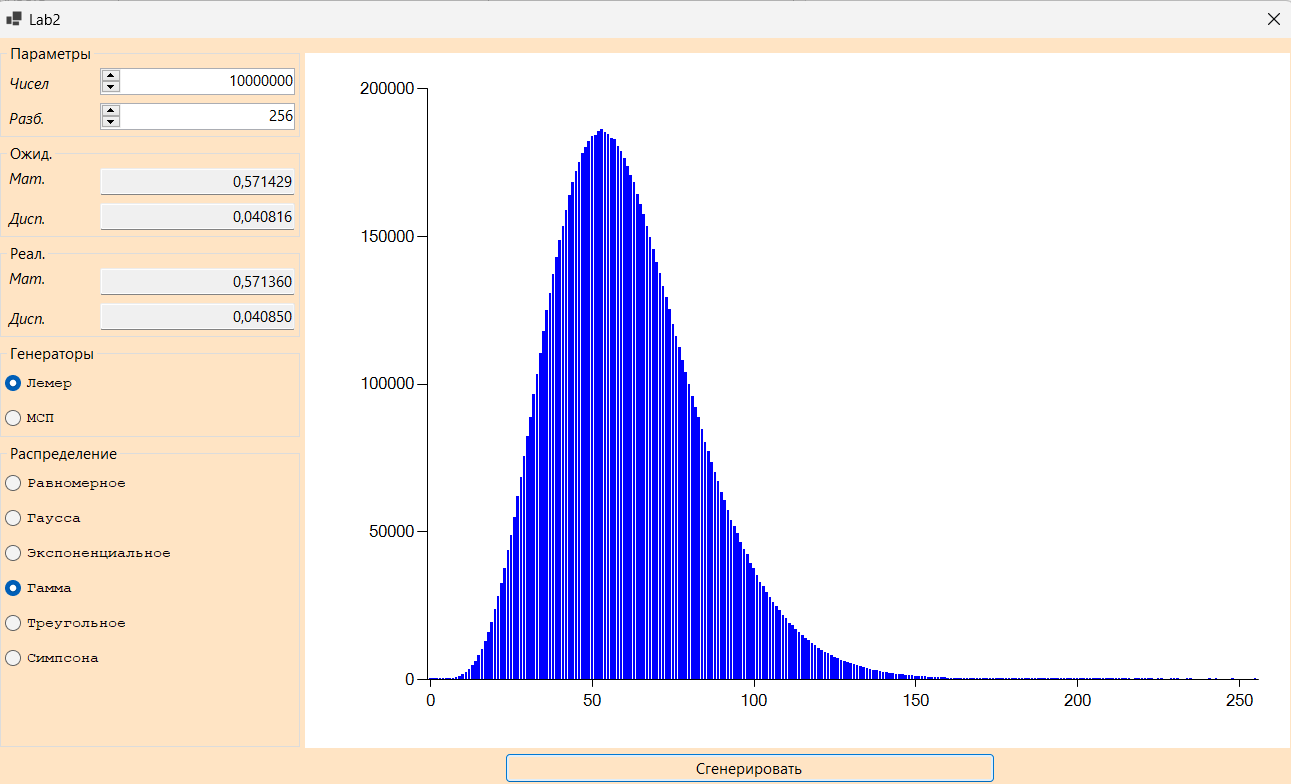


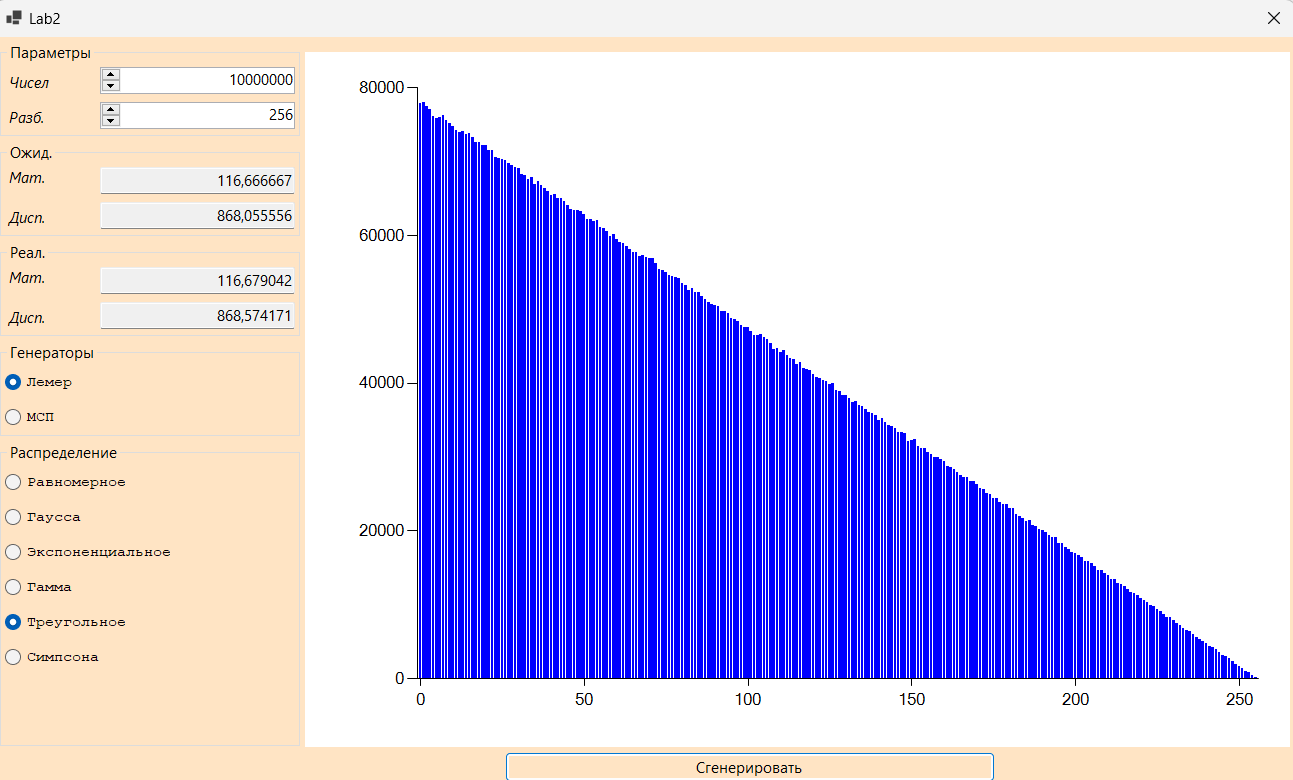
Генератор с выборкой равной n = 100000000

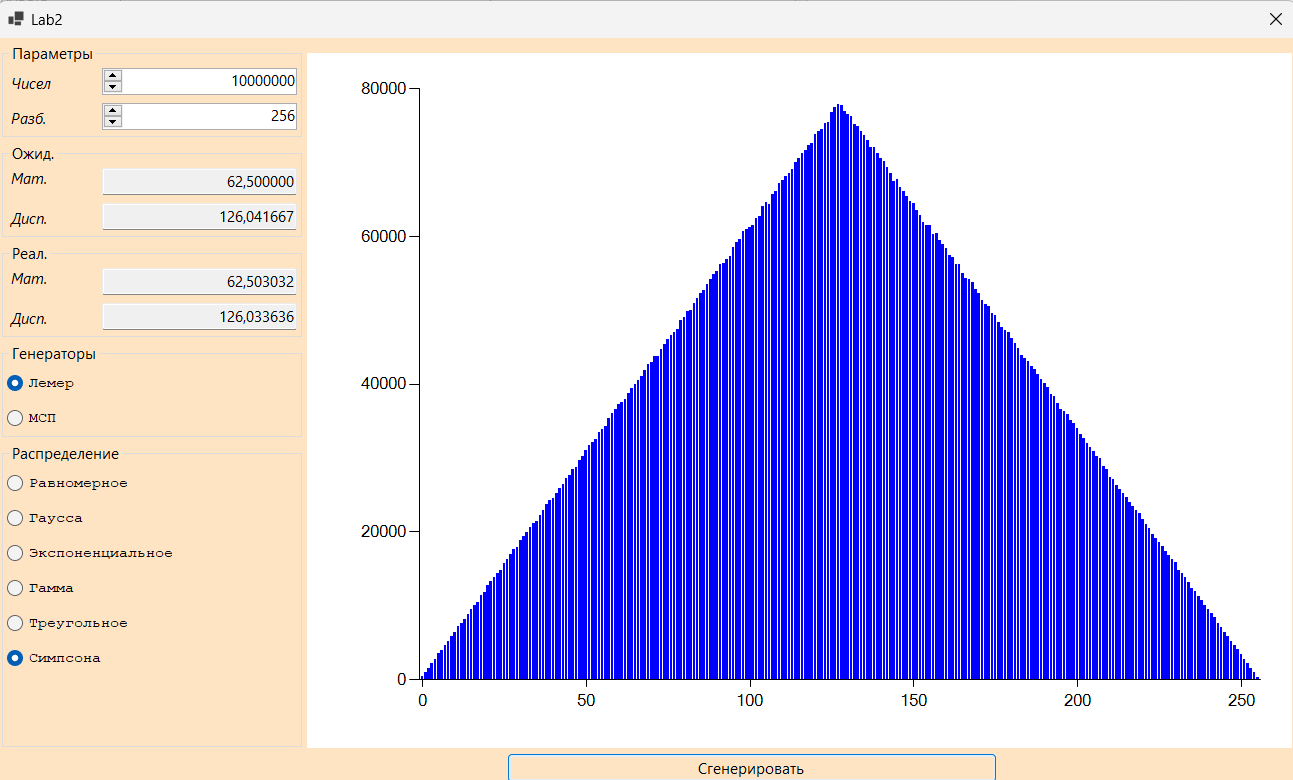










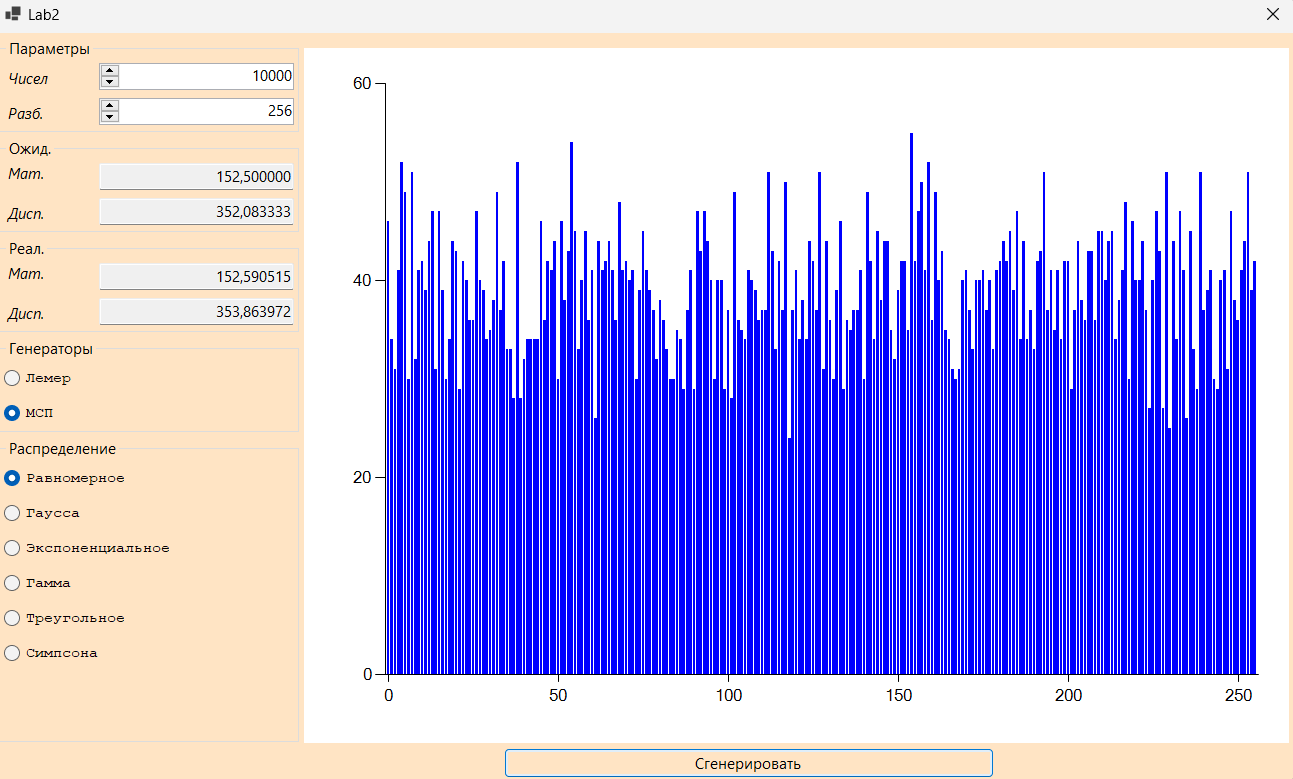


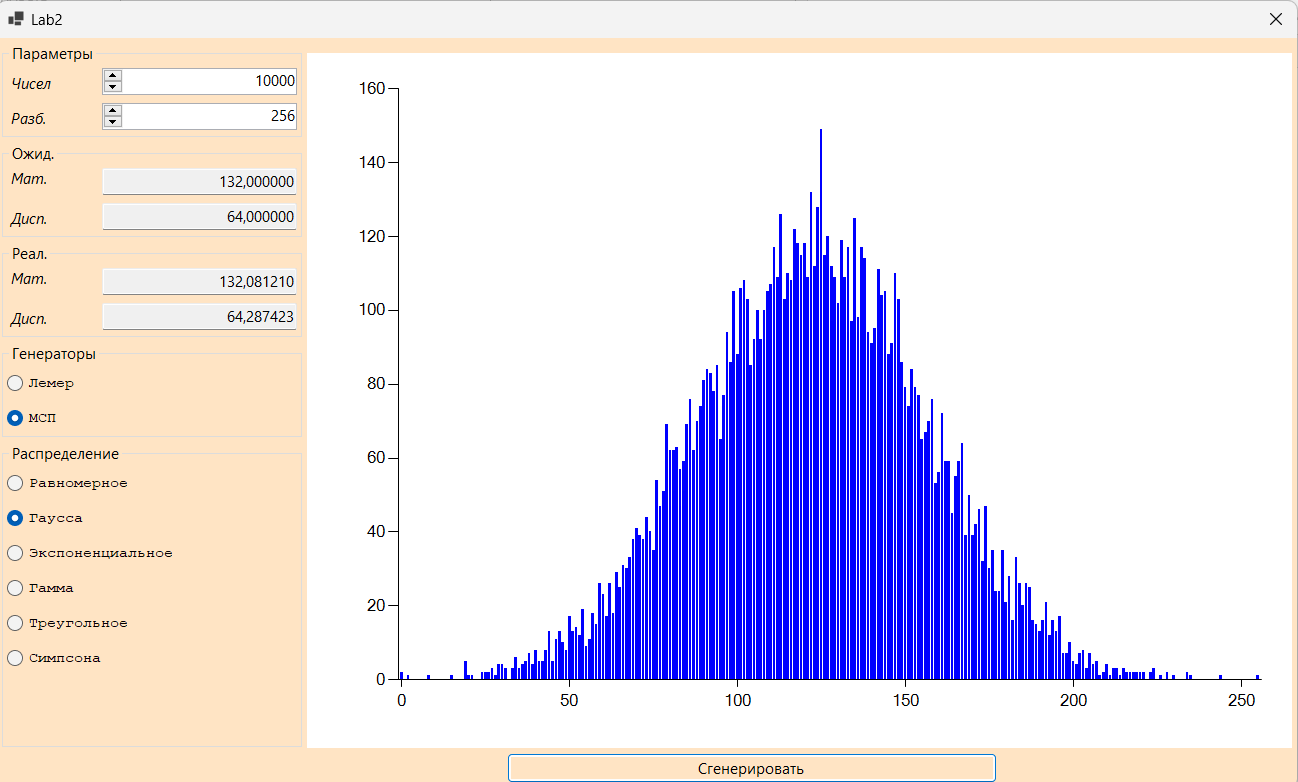
**3.2 Генератор серединных произведений**

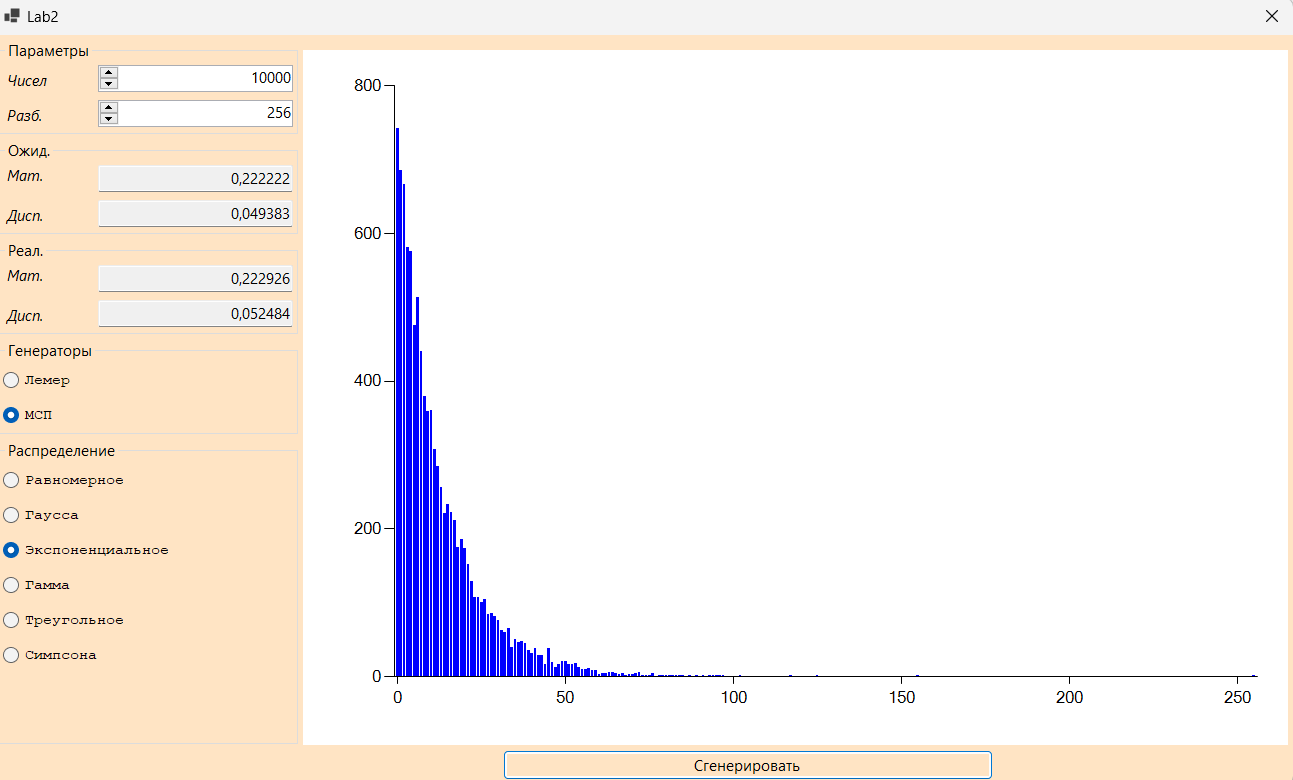
Генератор на основе алгоритма серединных произведений. В данном генераторе используются следующие коэффициенты:

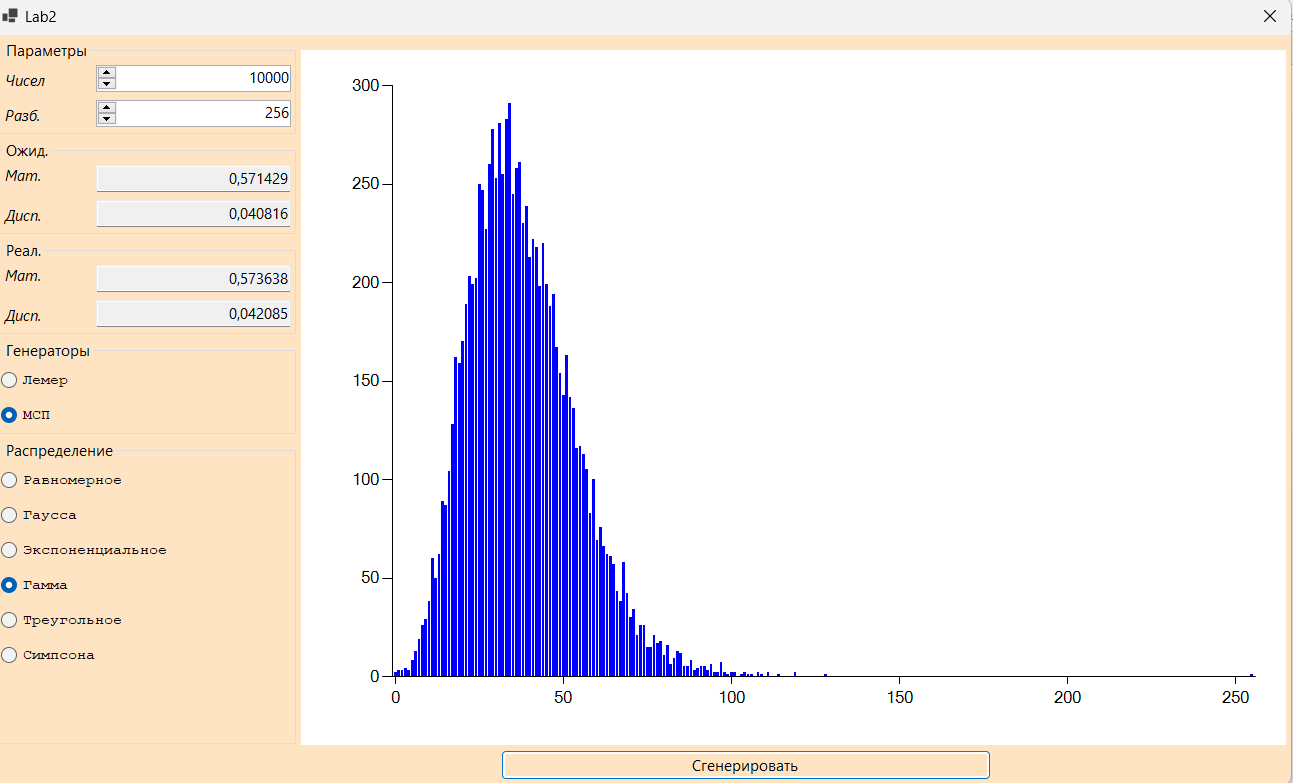
1. R0 = 20\_214\_752;
2. R1 = 9\_842\_349\_174\_754.

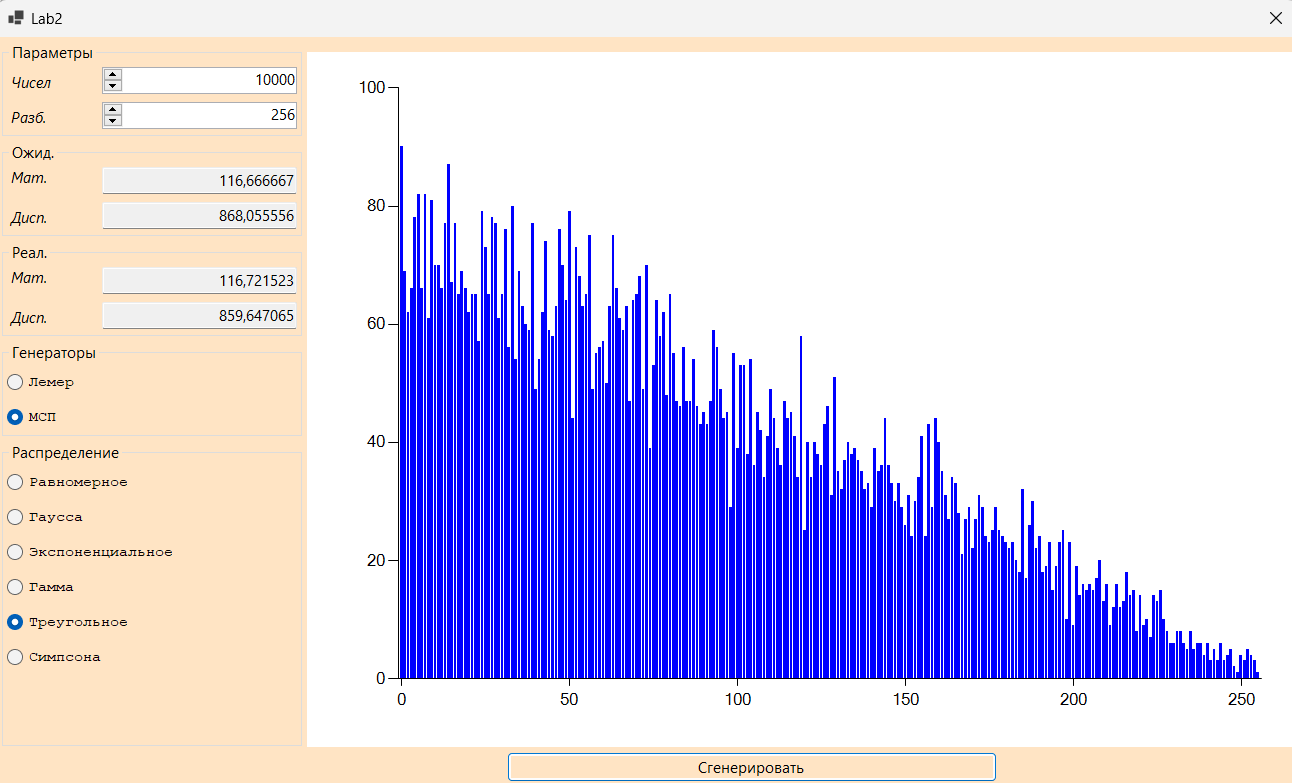
Генератор с выборкой равной n = 10000

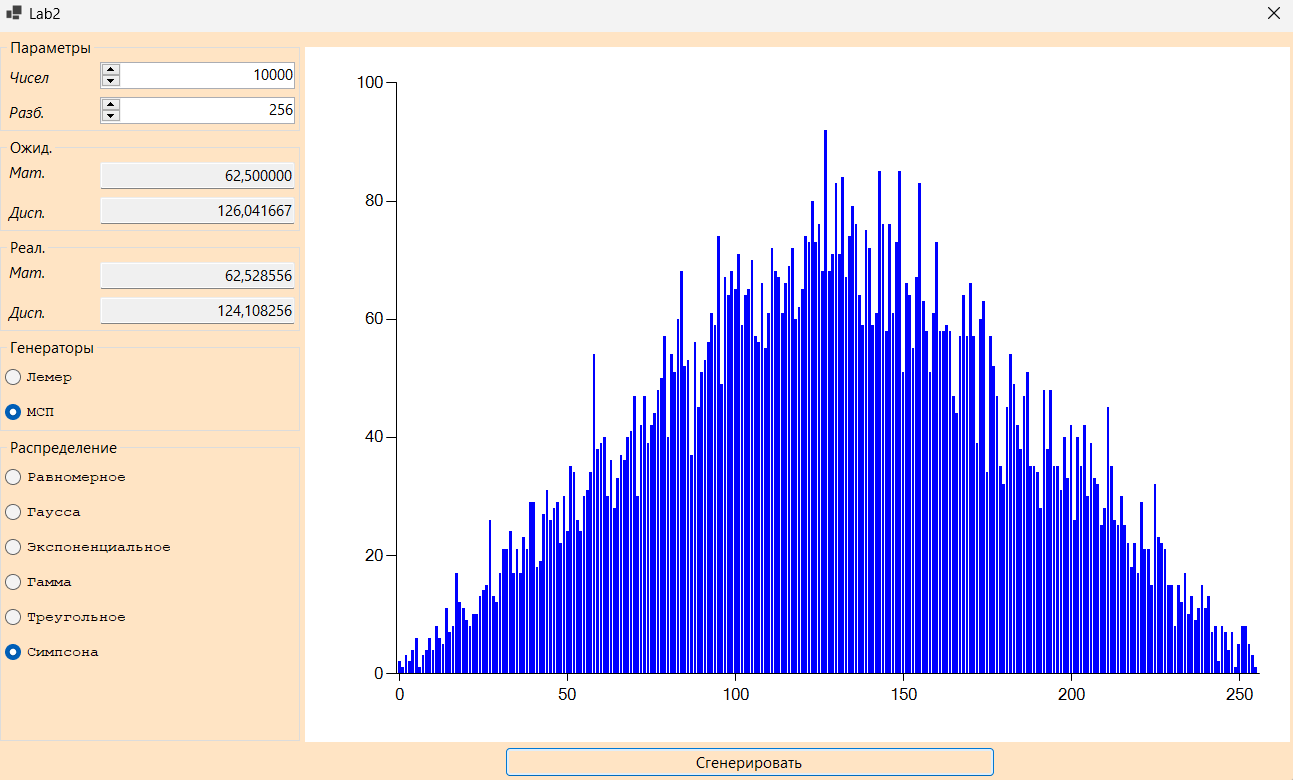




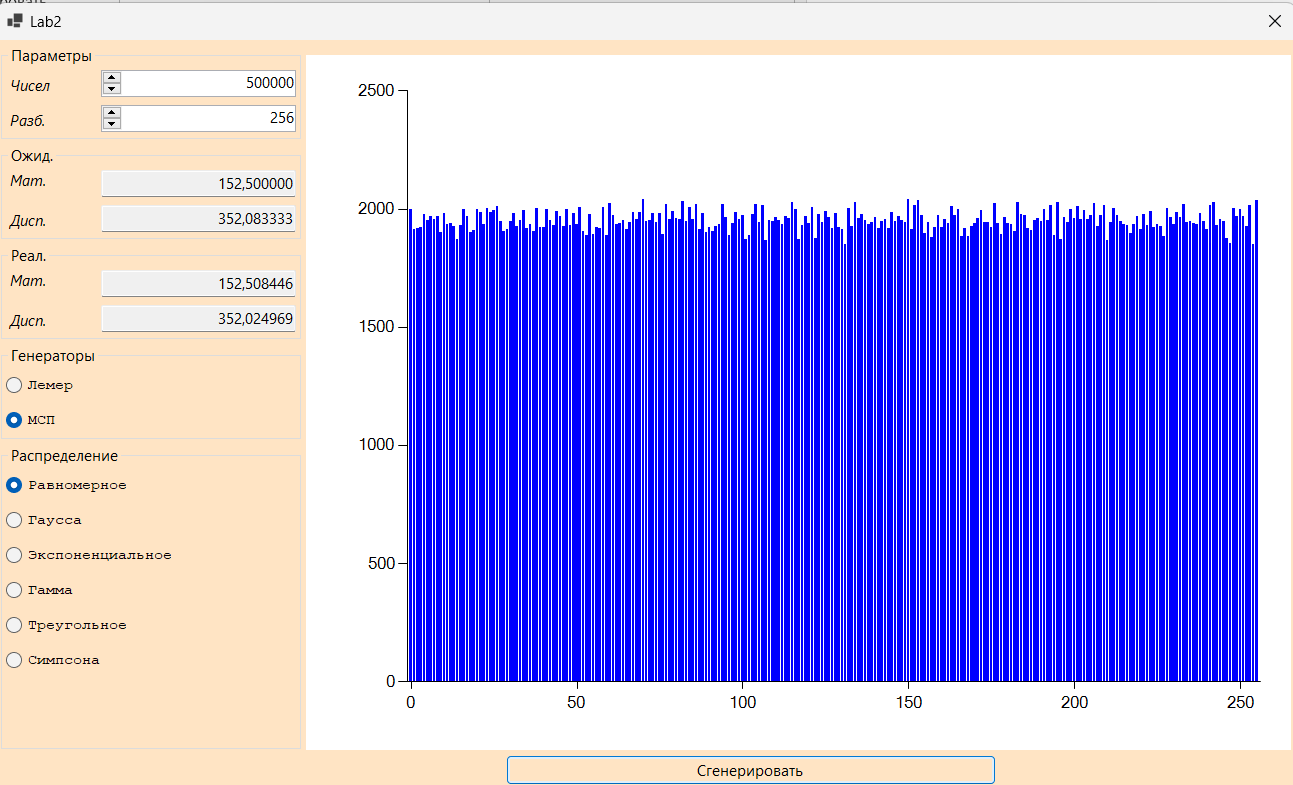


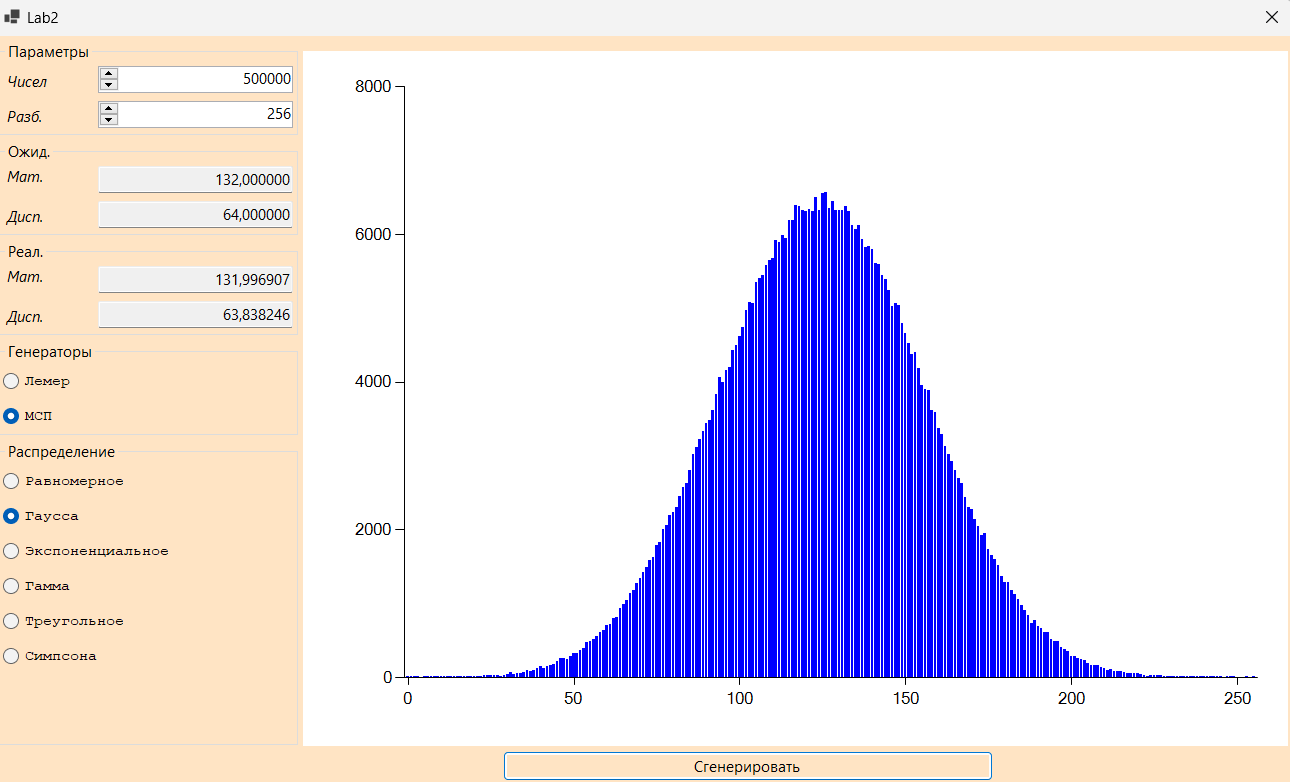


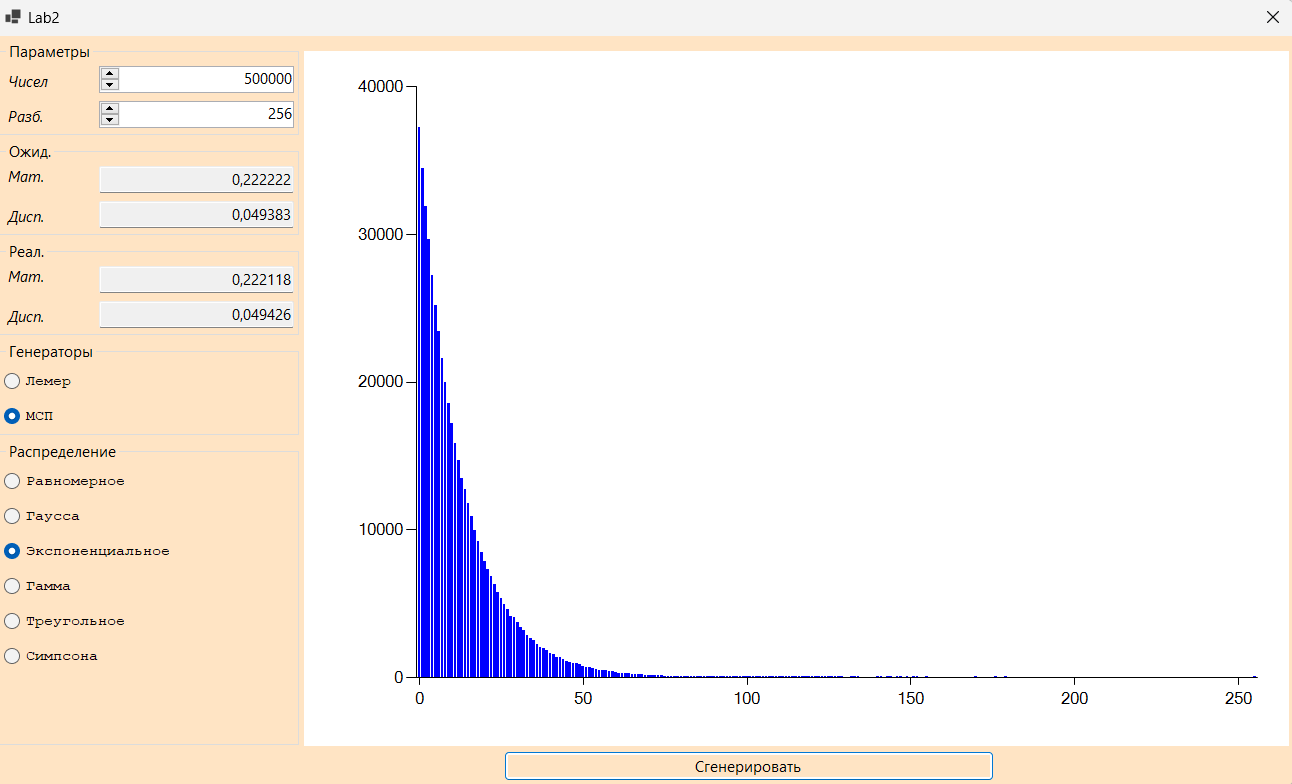


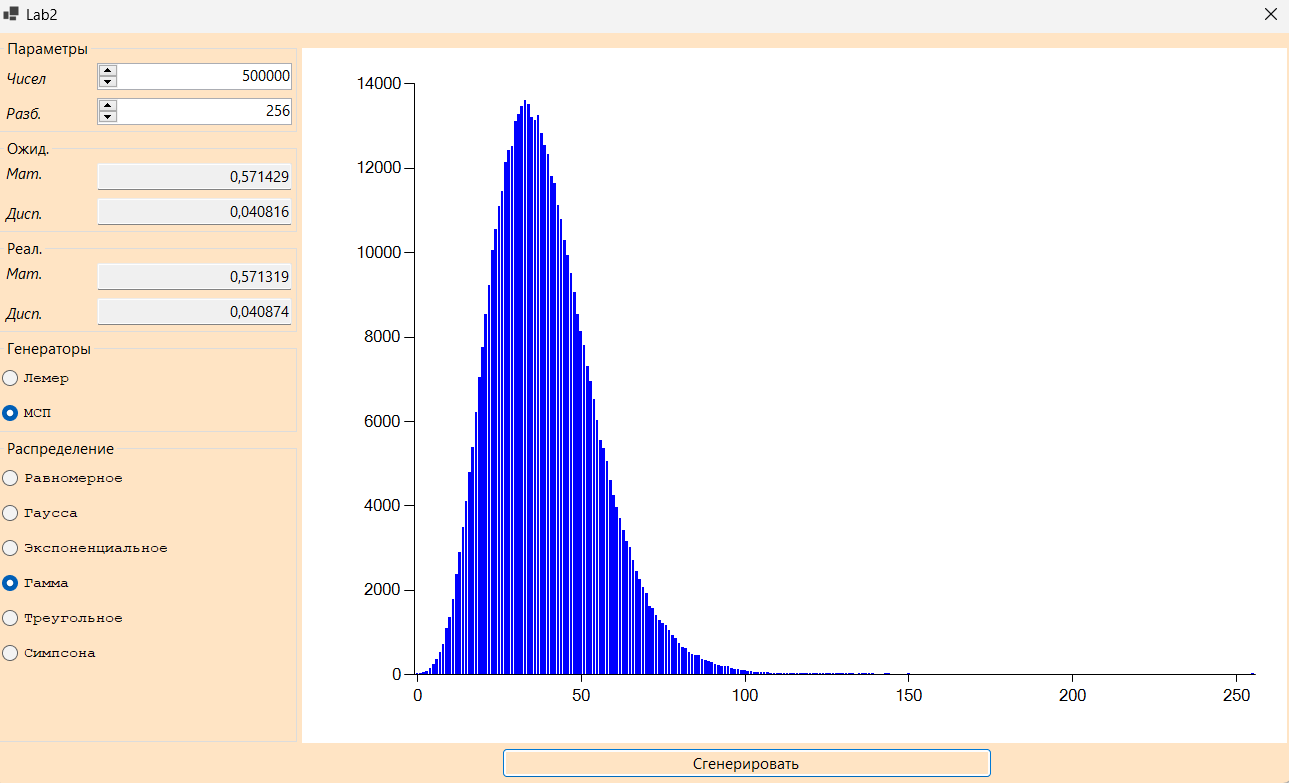


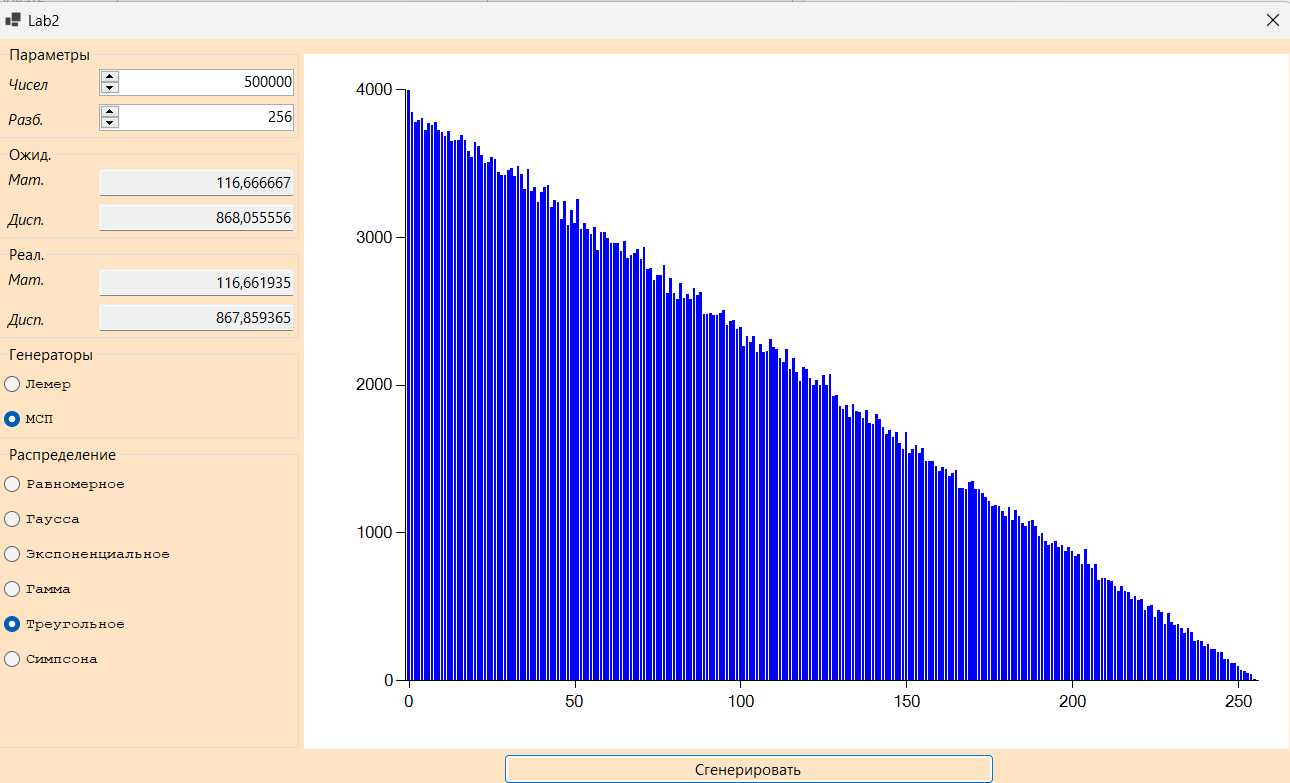
Генератор с выборкой равной n = 500000

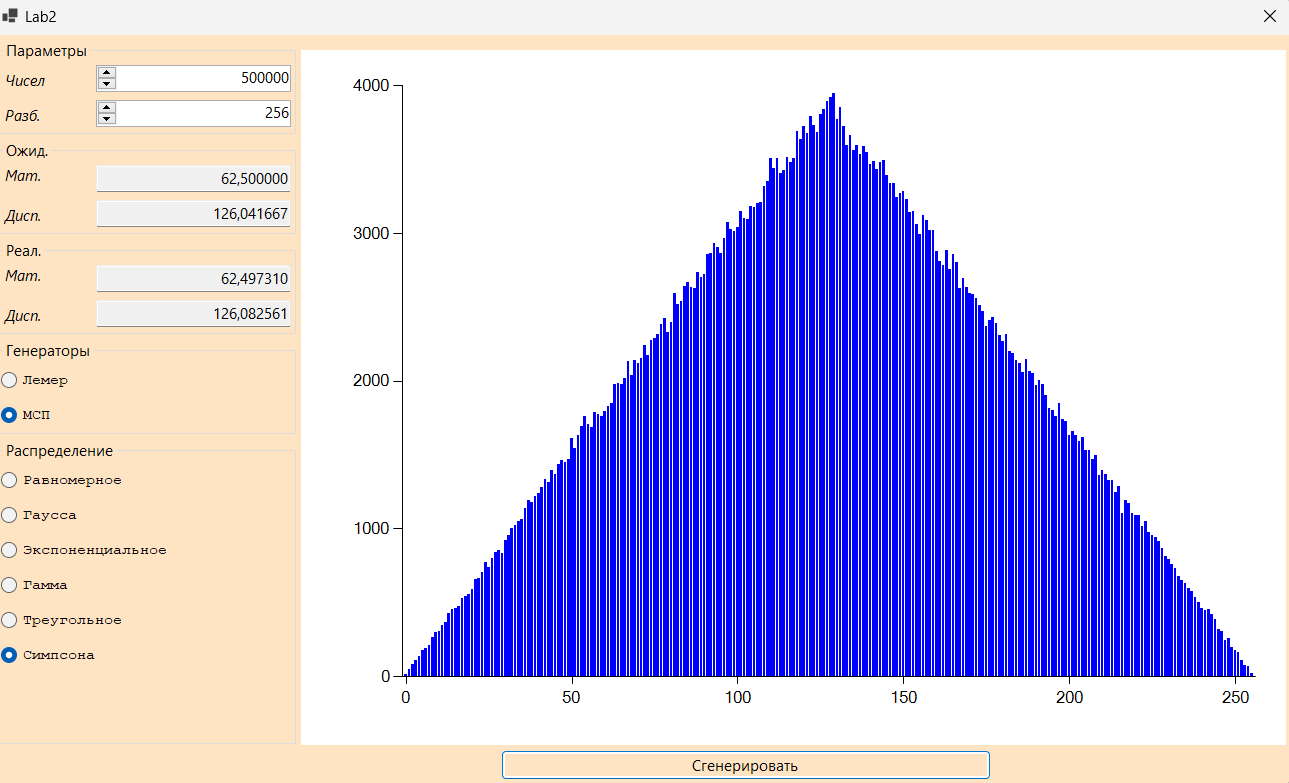




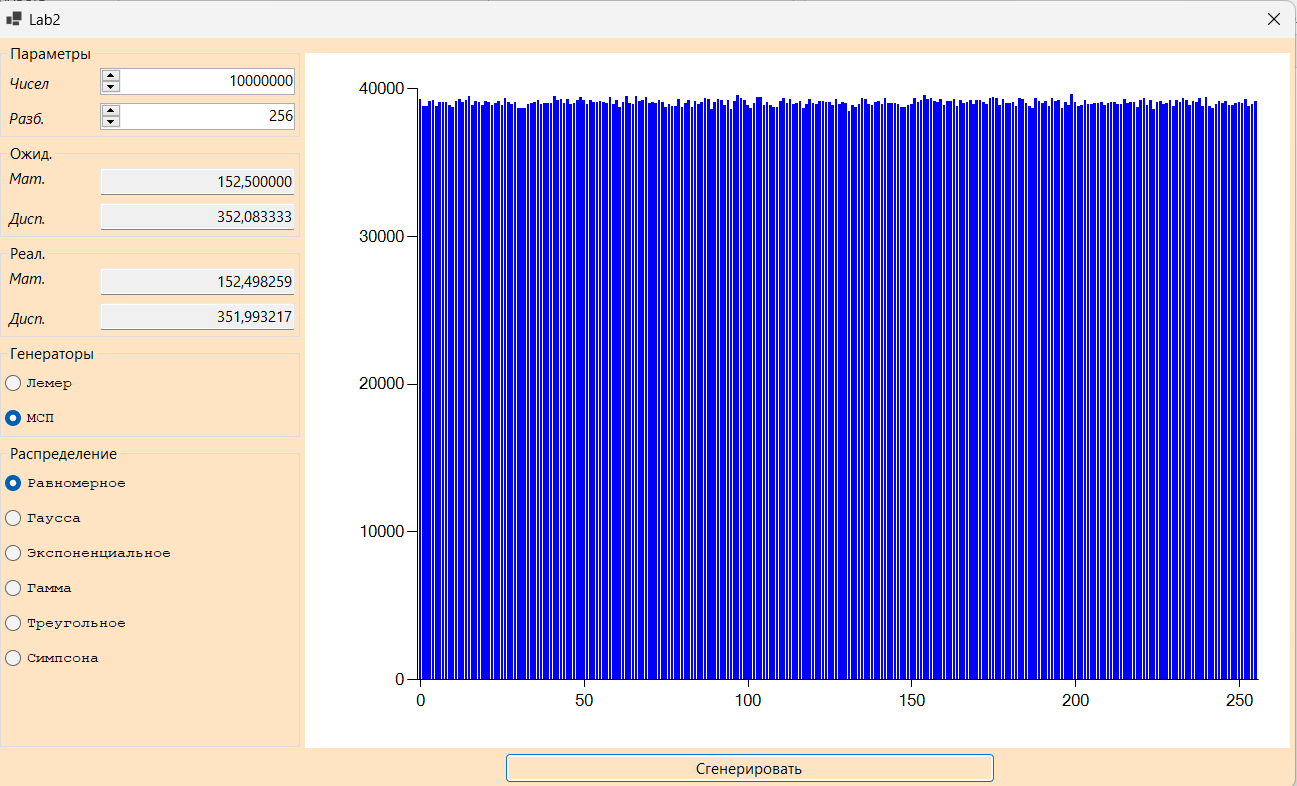


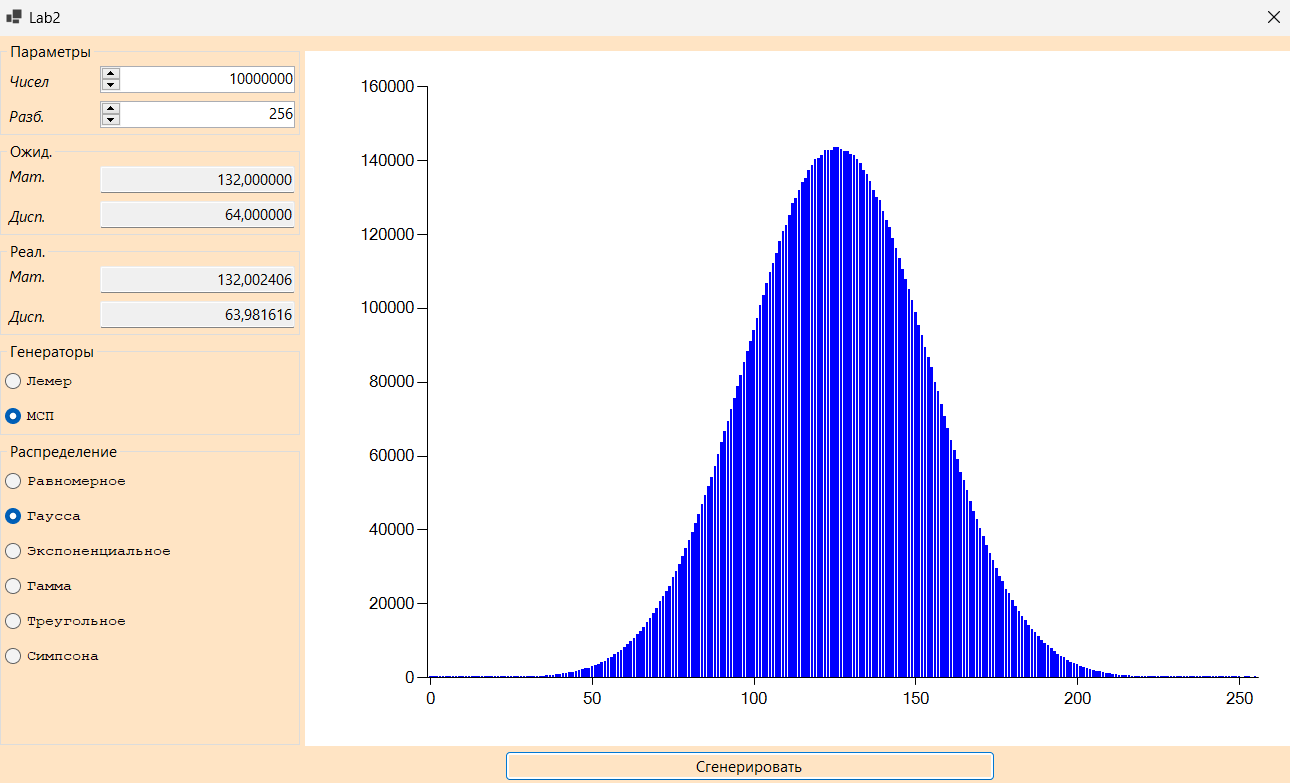


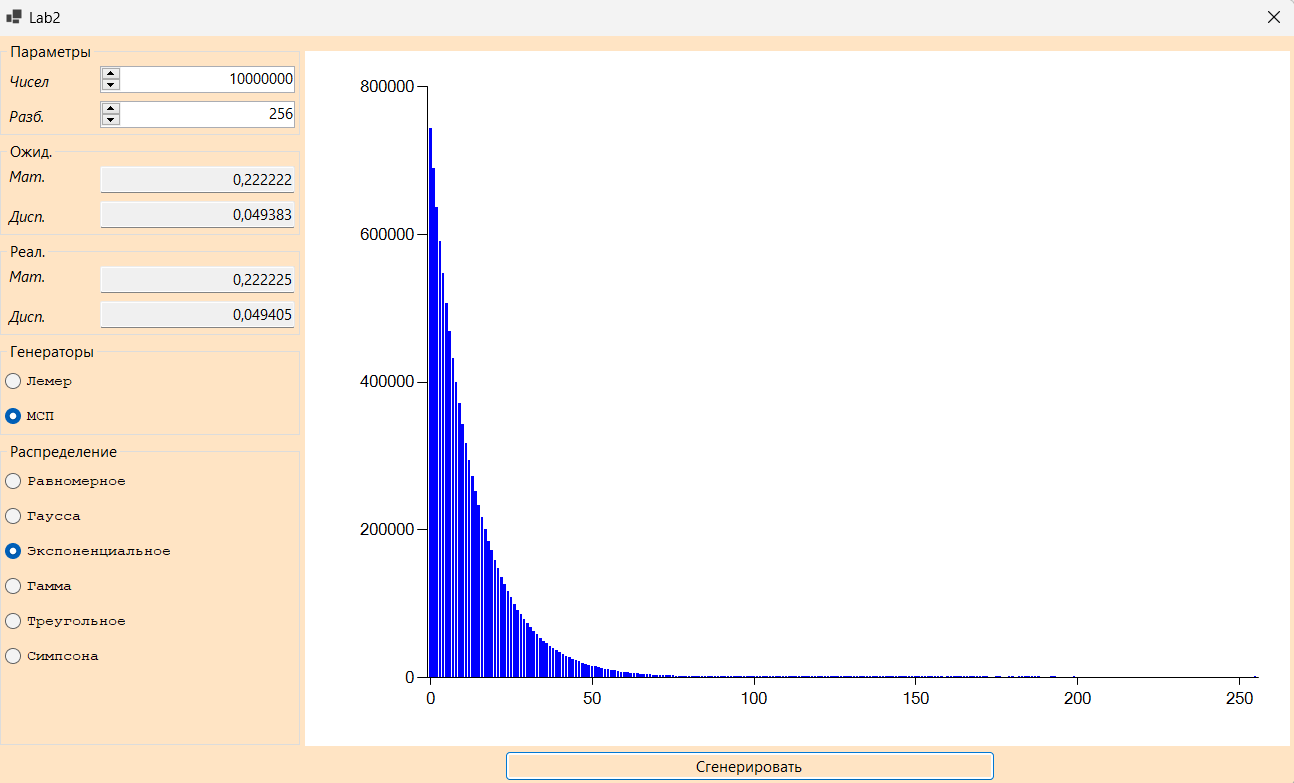


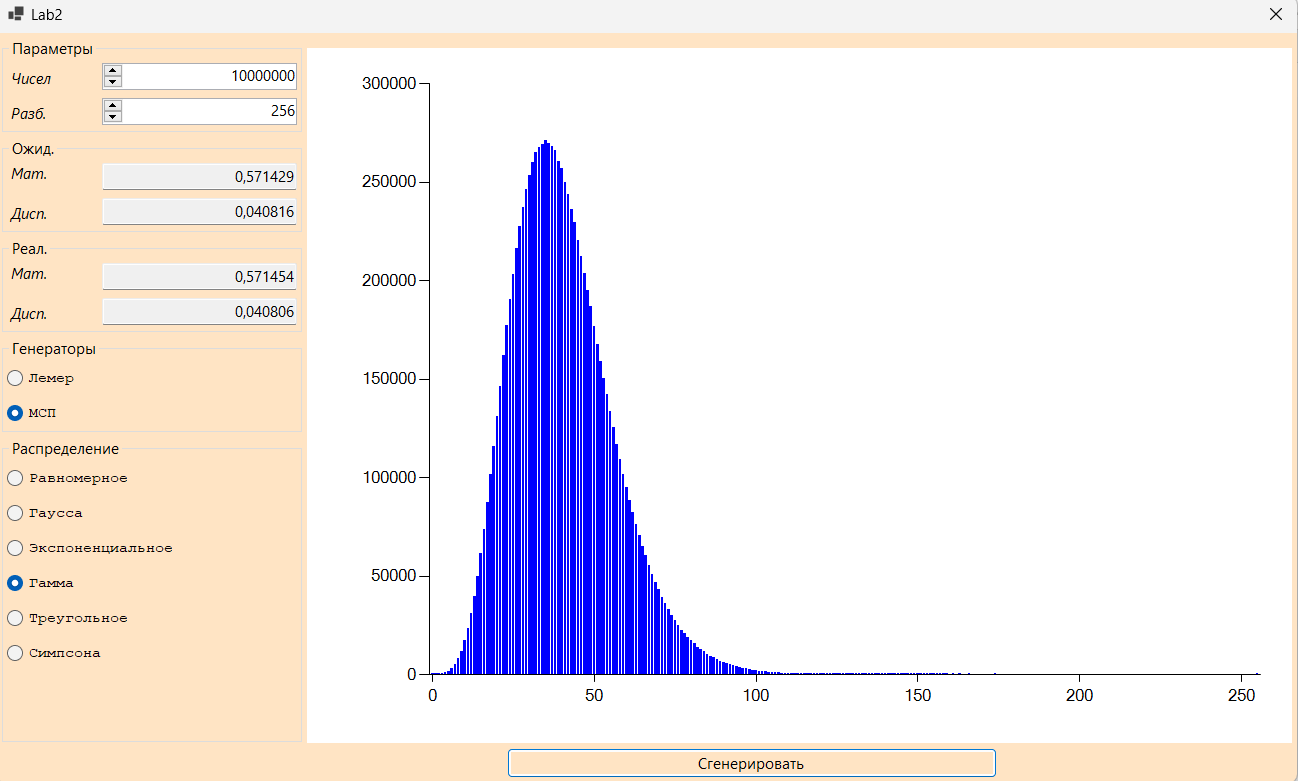


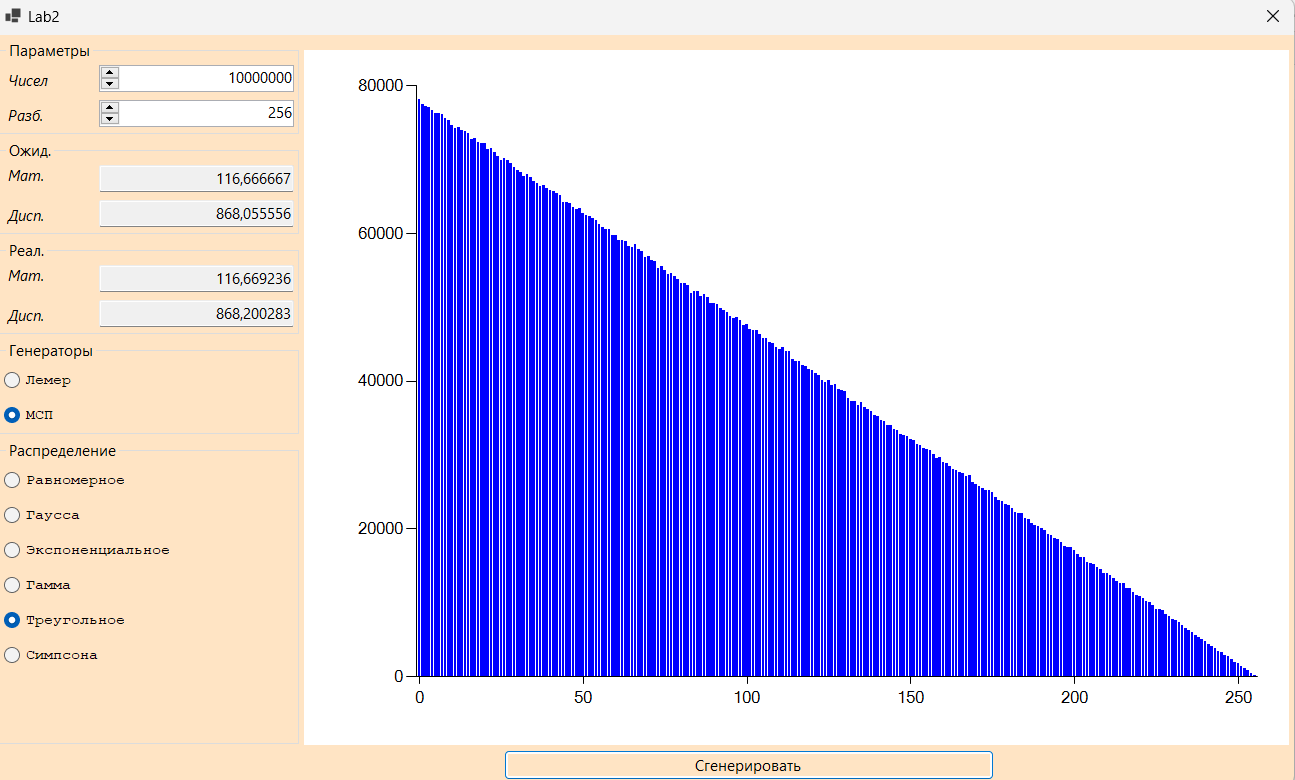
Генератор с выборкой равной n = 100000000

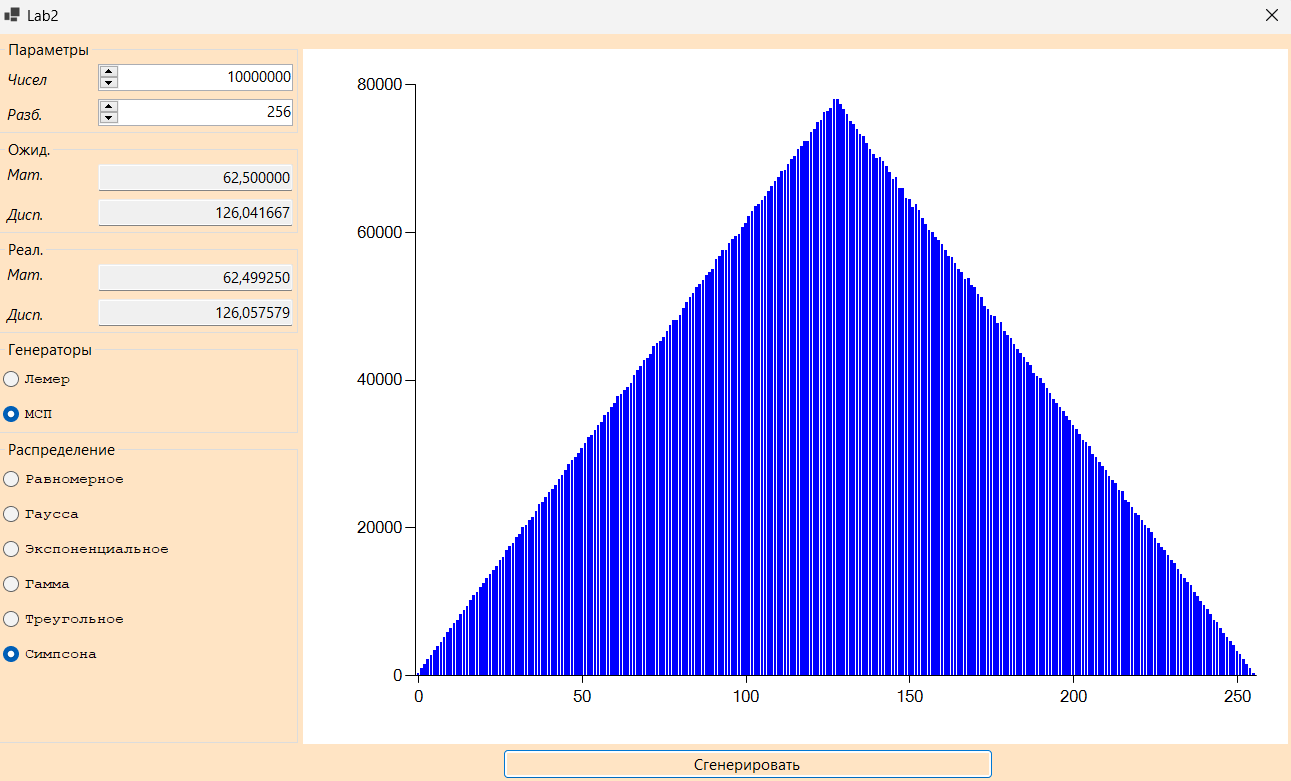












**4 Листинг кода**

Генератор Лемера:

public class Lehmer : IRandom

{

private BigInteger A { get; }

private BigInteger M { get; }

private BigInteger X { get; set; }

public Lehmer(ulong a, ulong m, ulong x)

{

A = a;

M = m;

X = x;

}

public ulong Next()

{

X = A \* X % M;

return (ulong)(X & 0xFFFF\_FFFF\_FFFF\_FFFF);

}

}

Создание экземпляра данного генератора происходит со следующими коэффициентами:

1. A = 16\_769\_023;
2. M = ulong.MaxValue - 1\_678;
3. X = 5\_542\_985\_019\_387.

Генератор серединных произведений:

public class Mpm : IRandom

{

private BigInteger R0 { get; set; }

private BigInteger R1 { get; set; }

public Mpm(ulong r0, ulong r1)

{

R0 = r0;

R1 = r1;

}

public ulong Next()

{

(R0, R1) = (R1, (R0 \* R1 >> 32) & 0xFFFF\_FFFF\_FFFF\_FFFF);

return (ulong)R1;

}

}

Создание экземпляра данного генератора происходит со следующими коэффициентами:

1. R0 = 20\_214\_752;
2. R1 = 9\_842\_349\_174\_754.