Цель работы

Выполнить программную реализацию генератора дискретной случайной величины.

1. Ход выполнения работы

1.1. Первые 30 значений x_i:

Таблица 1 – Случайные величины

i	x_i	i	x_i	i	x_i
1	-11.3	11	-11.3	21	42.4
2	-80.1	12	-11.6	22	79.1
3	42.4	13	-11.6	23	-80.1
4	79.1	14	42.4	24	36.2
5	79.1	15	42.4	25	-11.6
6	-11.3	16	36.2	26	-11.3
7	-11.3	17	79.1	27	-11.3
8	-11.3	18	36.2	28	-80.1
9	79.1	19	-80.1	29	42.4
10	79.1	20	-77.2	30	42.4

1.2. Результаты эмпирических и теоретических значений M и D:

Математическое ожидание M и дисперсия D дискретной СВ определяются по формулам:

$$M(x) = \sum_{j=1}^{K} p_j x_j ;$$

$$D(x) = \sum_{j=1}^{K} p_j x_j^2 - M^2(x) ,$$

где K – число возможных значений дискретной СВ.

Таблица 2 - Сравнение результатов

Теоретическое	Практическое
$M(x) = 5.5672$ $D(x) = 2533.815$	$M(x) = 5.50089$ $D(x) = 2579.42735$

1.3. Гистограммы распределения эмпирических и теоретических вероятностей дискретной СВ:

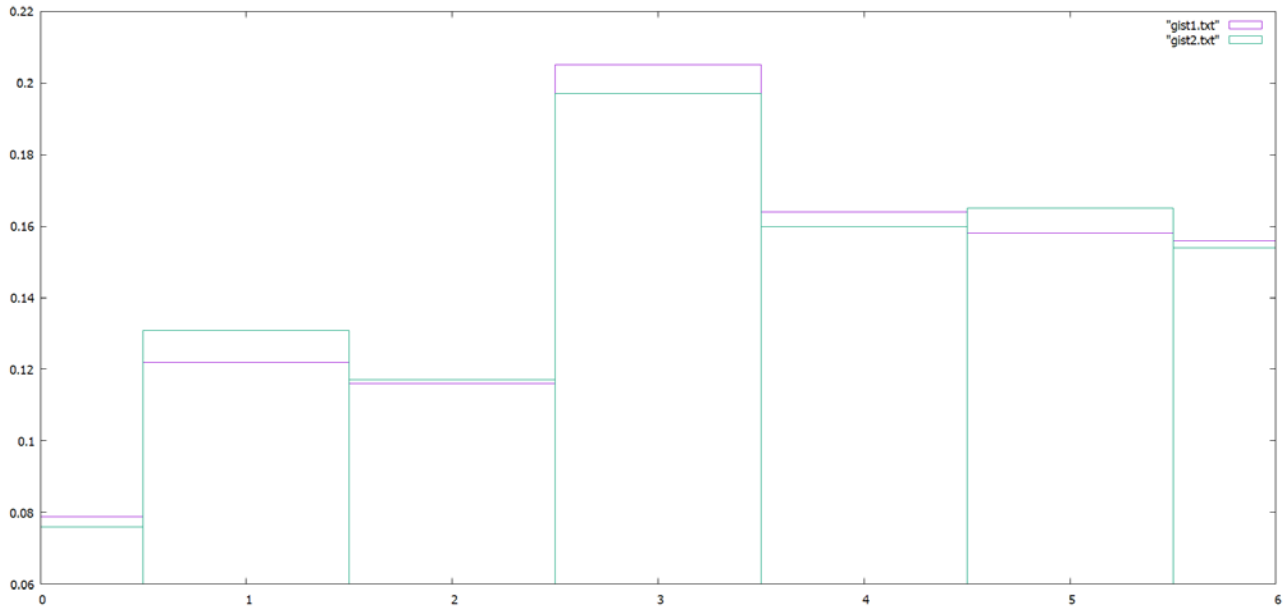

График 1 - Гистограмма распределения вероятностей дискретной СВ

Вывод

В ходе выполнения лабораторной работы был реализован генератор дискретной случайной величины. Из гистограммы видно, что теоретическая вероятность дискретной СВ хоть и отличается от практической, но не очень значительно.

Приложение А

Листинг программы

```
import java.io.IOException;
import java.util.LinkedList;

public class Main {
    public static void main(String[] args) throws IOException {
        Fibonacci fibonacci = new Fibonacci();
        Files files = new Files();
        LinkedList<Double> rand = fibonacci.ZRand();
        LinkedList<Double> Pmas = new LinkedList<>();
        LinkedList<Double> Xmas = new LinkedList<>();
        for (int i = 0; i < 7; i++) {
            Pmas.add(0.0);
        }

        LinkedList<Double> xj = new LinkedList<>();
        xj.add(-80.1);
        xj.add(-77.2);
        xj.add(-11.6);
        xj.add(-11.3);
        xj.add(36.2);
        xj.add(42.4);
```

```
        xj.add(79.1);

        LinkedList<Double> pj = new LinkedList<>();
        pj.add(0.0);
        pj.add(0.079);
        pj.add(0.122);
        pj.add(0.116);
        pj.add(0.205);
        pj.add(0.164);
        pj.add(0.158);
        pj.add(0.156);
        files.gistFile2(pj,1);

        double exp = 0;
        int K = pj.size();
        for (int i = 1; i < K; i++) {
            exp += pj.get(i) * xj.get(i-1);
        }

        double variance = 0;
        for (int i = 1; i < K; i++) {
            variance += (pj.get(i) * Math.pow(xj.get(i-1), 2)) - Math.pow(exp,
2);
        }

        for (int i = 1; i < pj.size(); i++) {
            pj.set(i, (pj.get(i) + (pj.get(i - 1))));
        }

        for (int i = 0; i < rand.size(); i++) {
            for (int j = 0; j < pj.size() - 1; j++) {
                if (pj.get(j) < rand.get(i) && rand.get(i) < pj.get(j + 1)) {
                    Pmas.set(j, Pmas.get(j) + 1);
                    Xmas.add(xj.get(j));
                }
            }
        }

        for (int i = 0; i < Pmas.size(); i++) {
            Pmas.set(i, Pmas.get(i) / rand.size());
        }
        System.out.println(Pmas);
        files.gistFile2(Pmas,2);
        System.out.println();

        double realExp = 0;

        double sum = 0;
        for (Double xma : Xmas) {
            sum += xma;
        }
        realExp = sum / Xmas.size();
        System.out.println("Мат. ожидание равно: " + realExp);
        System.out.println("Норма равна: " + exp);
        System.out.println("Разница: " + (realExp - exp));
        System.out.println();


        double realVariance = 0;

        sum = 0;
        for (Double xma : Xmas) {
            sum += Math.pow(xma - realExp, 2);
        }
        realVariance = sum / Xmas.size();
        System.out.println("Дисперсия равна: " + realVariance);
        System.out.println("Норма равна: " + variance);
        System.out.println("Разница: " + (realVariance - variance));
```

```
        for (int i = 0; i < 30; i++) {
            System.out.print(Xmas.get(i) + ",  ");
            if (i % 10 == 0) {
                System.out.println();
            }
        }
    }
}
```