**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Теория массового обслуживания»**

**Тема: Моделирование непрерывных и дискретных случайных величин**

ВАРИАНТ 1

1. **Цель работы**

Получить знание о генераторах случайных величин и их практической реализации.

1. **Теоретические положения**

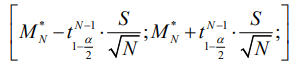
Генератор случайных чисел – математическая модель, позволяющая получить необходимую выборку случайных чисел, которые образуют собой случайную величину.

В работе построили модели генерирования ДСВ и НСВ, которые несколько отличаются друг от друга: для генерации НСВ необходимо использовать метод обратных функций, который заключается в том, что необходимо найти функцию F -1 , обратную заданной функции распределения F, после чего подставить в полученную функцию F -1 выборку значений Z, сгенерированных по нормальному закону. На выходе мы получим множество значений F -1 (Z), которые будут распределены по заданному закону F.

Для задание ДСВ удобно интервал [0,1] разделить на подынтервалы длинами рi, а затем задать выборку нормированных значений Z, тогда эти значения Z будут задавать вероятность выпадения P для каждого Xi значения заданной ДСВ. Необходимо будет соотнести по значению Z к какому интервалу рi будет относиться наш элемент Xi, затем элемент Xi можно «восстановить» по значению его вероятности.

1. **Функции, использованные в программе**

Данная функция реализует интервальную оценку среднего по формуле:

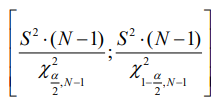


В этом отрезке кода выполняются расчёты интервальных оценок для дисперсии и среднего значения на основе выборок различных размеров (50, 200 и 1000). Вот краткое описание каждого блока:

Код функции:

1. % расчёт интервальной дисперсии для выборки 50
2. Arr\_50\_3\_Disp\_intervals = cell(1,length(alphas));
3. for i=1:length(alphas)
4. chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);
5. chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);
6. distrDisp = var(Arr\_50(1:18))\*(18-1);
7. Arr\_50\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];
8. end
9. % расчёт интервальной оценки среднего для выборки 200
10. Arr\_200\_3\_Mean\_intervals = cell(1,length(alphas));
11. Arr\_200\_18\_Mean = mean(Arr\_200(1:18));
12. for i=1:length(alphas)
13. k\_STD = tinv(1 - alphas(i) / 2, 18 - 1);
15. margin\_of\_error = k\_STD \* (sqrt(var(Arr\_200(1:18))) / sqrt(18));
16. Arr\_200\_3\_Mean\_intervals{i} = [Arr\_200\_18\_Mean - k\_STD \* margin\_of\_error, Arr\_200\_18\_Mean + k\_STD \* margin\_of\_error];
17. end
18. % расчёт интервальной дисперсии для выборки 200
19. Arr\_200\_3\_Disp\_intervals = cell(1,length(alphas));
20. for i=1:length(alphas)
21. chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);
22. chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);
23. distrDisp = var(Arr\_200(1:18))\*(18-1);
24. Arr\_200\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];
25. end
26. % расчёт интервальной оценки среднего для выборки 1000
27. Arr\_1000\_3\_Mean\_intervals = cell(1,length(alphas));
28. Arr\_1000\_18\_Mean = mean(Arr\_1000(1:18));
29. for i=1:length(alphas)
30. k\_STD = tinv(1 - alphas(i) / 2, 18 - 1);
32. margin\_of\_error = k\_STD \* (sqrt(var(Arr\_1000(1:18))) / sqrt(18));
33. Arr\_1000\_3\_Mean\_intervals{i} = [Arr\_1000\_18\_Mean - k\_STD \* margin\_of\_error, Arr\_1000\_18\_Mean + k\_STD \* margin\_of\_error];
34. end

Функция «dispersia» реализует интервальную оценку среднего по формуле:



В данном отрезке кода происходит расчет интервальной дисперсии для трех выборок размером 50, 200 и 1000

Код функции:

% расчёт интервальной дисперсии для выборки 50

Arr\_50\_3\_Disp\_intervals = cell(1,length(alphas));

for i=1:length(alphas)

chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);

chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);

distrDisp = var(Arr\_50(1:18))\*(18-1);

Arr\_50\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];

end

% расчёт интервальной дисперсии для выборки 200

Arr\_200\_3\_Disp\_intervals = cell(1,length(alphas));

for i=1:length(alphas)

chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);

chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);

distrDisp = var(Arr\_200(1:18))\*(18-1);

Arr\_200\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];

end

% расчёт интервальной дисперсии для выборки 1000

Arr\_1000\_3\_Disp\_intervals = cell(1,length(alphas));

for i=1:length(alphas)

chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);

chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);

distrDisp = var(Arr\_1000(1:18))\*(18-1);

Arr\_1000\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];

end

В данном отрезке кода реализована функция generateDescrete, которая генерирует выборку из дискретного распределения на основе заданных вероятностей.

Код функции:

function list = generateDescrete(f,count)

list = zeros(count, 1);

for j=1:count

z = rand();

S = z;

i = 1;

% 5 - количество возможных значений

while i <= 5

S = S - f(i);

if S < 0

list(j) = i;

break;

end

i = i + 1;

end

end

end

1. **Реализация генератора случайных чисел для НСВ.**

В данном отрезке кода выполняется генерация выборок из распределения, оценка статистических характеристик и расчет интервальных оценок для среднего и дисперсии.

% Функция распределения

F = @(x) integral(f, a, x, 'ArrayValued', true);

F\_inv = @(x) a + (b-a)\*F(x);

% Начало генерации выборки

Arr\_50 = rand(50, 1);

Arr\_200 = rand(200, 1);

Arr\_1000 = rand(1000, 1);

for i=1:length(Arr\_50)

Arr\_50(i) = F\_inv(Arr\_50(i));

end

for i=1:length(Arr\_200)

Arr\_200(i) = F\_inv(Arr\_200(i));

end

for i=1:length(Arr\_1000)

Arr\_1000(i) = F\_inv(Arr\_1000(i));

end

% Конец генерации выборки

% точечные оценки среднего, дисперсии и среднеквадратического отклонения (СКО)

% точечная оценка среднего (математичекое ожидание)

Arr\_50\_Mean = mean(Arr\_50);

% точечная оценка дисперсии

Arr\_50\_Disp = var(Arr\_50);

% СКО

Arr\_50\_Std = std(Arr\_50);

% точечная оценка среднего (математичекое ожидание)

Arr\_200\_Mean = mean(Arr\_200);

% точечная оценка дисперсии

Arr\_200\_Disp = var(Arr\_200);

% СКО

Arr\_200\_Std = std(Arr\_200);

% точечная оценка среднего (математичекое ожидание)

Arr\_1000\_Mean = mean(Arr\_1000);

% точечная оценка дисперсии

Arr\_1000\_Disp = var(Arr\_1000);

% СКО

Arr\_1000\_Std = std(Arr\_1000);

% Расчитайте интервальные оценки среднего и дисперсия

alphas = [0.1, 0.05, 0.01];

% расчёт интервальной оценки среднего для выборки 50

Arr\_50\_3\_Mean\_intervals = cell(1,length(alphas));

Arr\_50\_18\_Mean = mean(Arr\_50(1:18));

for i=1:length(alphas)

k\_STD = tinv(1 - alphas(i) / 2, 18 - 1);

margin\_of\_error = k\_STD \* (sqrt(var(Arr\_50(1:18))) / sqrt(18));

Arr\_50\_3\_Mean\_intervals{i} = [Arr\_50\_18\_Mean - k\_STD \* margin\_of\_error, Arr\_50\_18\_Mean + k\_STD \* margin\_of\_error];

end

% расчёт интервальной дисперсии для выборки 50

Arr\_50\_3\_Disp\_intervals = cell(1,length(alphas));

for i=1:length(alphas)

chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);

chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);

distrDisp = var(Arr\_50(1:18))\*(18-1);

Arr\_50\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];

end

% расчёт интервальной оценки среднего для выборки 200

Arr\_200\_3\_Mean\_intervals = cell(1,length(alphas));

Arr\_200\_18\_Mean = mean(Arr\_200(1:18));

for i=1:length(alphas)

k\_STD = tinv(1 - alphas(i) / 2, 18 - 1);

margin\_of\_error = k\_STD \* (sqrt(var(Arr\_200(1:18))) / sqrt(18));

Arr\_200\_3\_Mean\_intervals{i} = [Arr\_200\_18\_Mean - k\_STD \* margin\_of\_error, Arr\_200\_18\_Mean + k\_STD \* margin\_of\_error];

end

% расчёт интервальной дисперсии для выборки 200

Arr\_200\_3\_Disp\_intervals = cell(1,length(alphas));

for i=1:length(alphas)

chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);

chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);

distrDisp = var(Arr\_200(1:18))\*(18-1);

Arr\_200\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];

end

% расчёт интервальной оценки среднего для выборки 1000

Arr\_1000\_3\_Mean\_intervals = cell(1,length(alphas));

Arr\_1000\_18\_Mean = mean(Arr\_1000(1:18));

for i=1:length(alphas)

k\_STD = tinv(1 - alphas(i) / 2, 18 - 1);

margin\_of\_error = k\_STD \* (sqrt(var(Arr\_1000(1:18))) / sqrt(18));

Arr\_1000\_3\_Mean\_intervals{i} = [Arr\_1000\_18\_Mean - k\_STD \* margin\_of\_error, Arr\_1000\_18\_Mean + k\_STD \* margin\_of\_error];

end

% расчёт интервальной дисперсии для выборки 1000

Arr\_1000\_3\_Disp\_intervals = cell(1,length(alphas));

for i=1:length(alphas)

chiSQLeft = chi2inv((alphas(i)) / 2, 18 - 1);

chiSQRight = chi2inv((1 - alphas(i)) / 2, 18 - 1);

distrDisp = var(Arr\_1000(1:18))\*(18-1);

Arr\_1000\_3\_Disp\_intervals{i} = [distrDisp / chiSQLeft, distrDisp / chiSQRight];

end

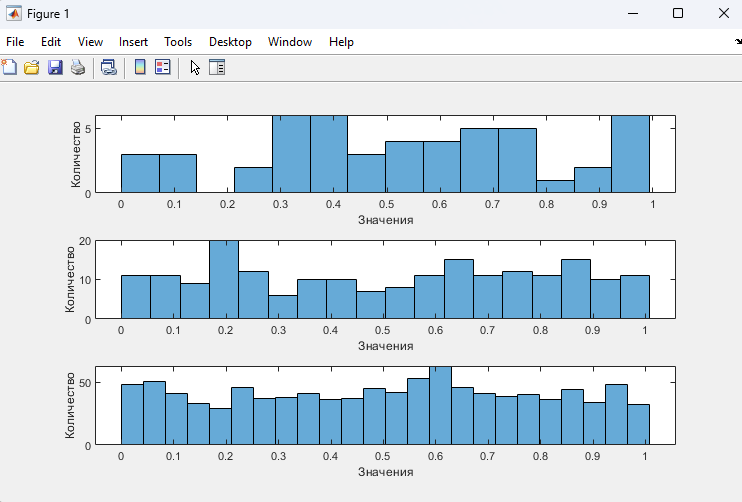
**5. Графики и таблицы программы**

Рисунок 1. График №1. Задание №6.

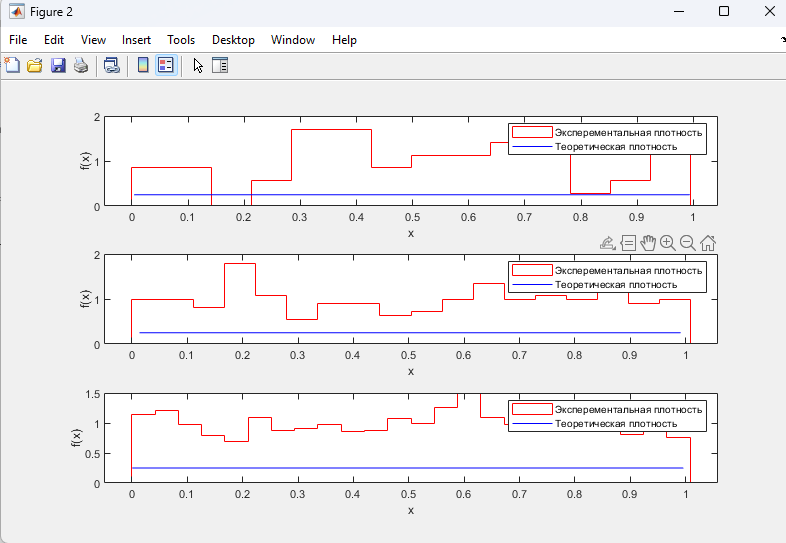


Рисунок 2. График №2. Задание №7.

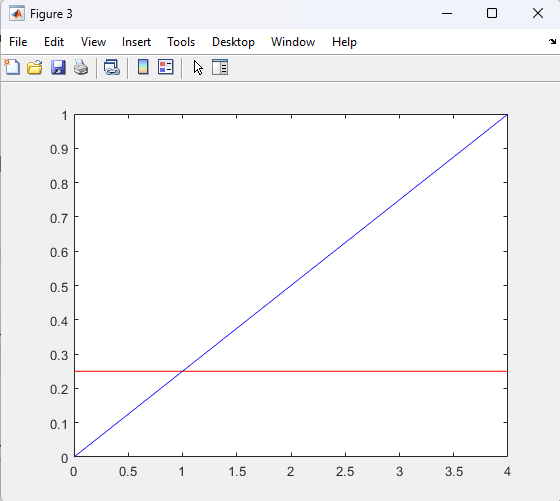


Рисунок 3. График №3. Задание №8.

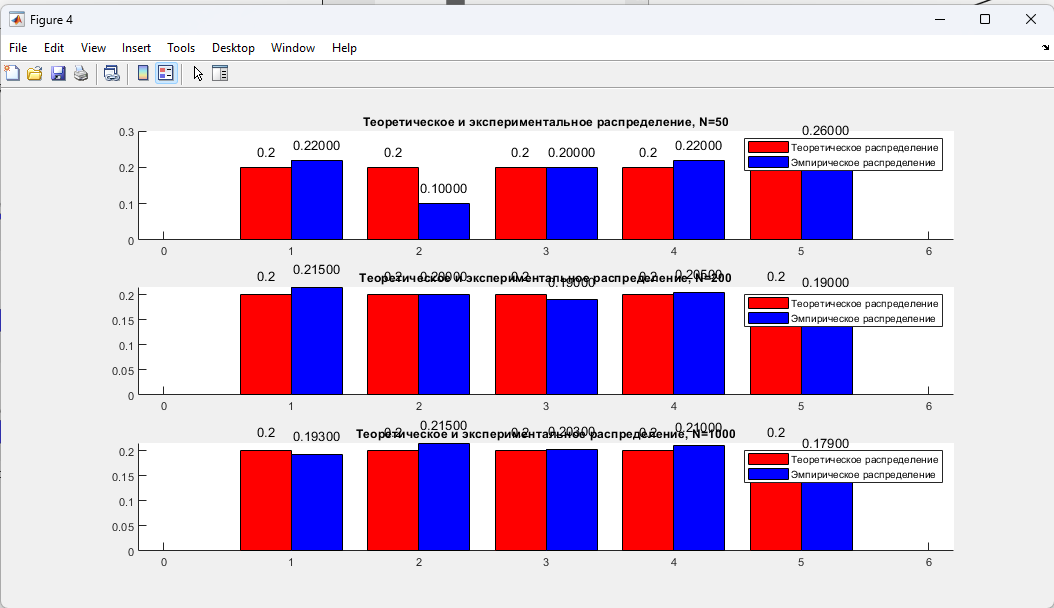


Рисунок 4. График №4. Дискретно распределенная случайная величина.

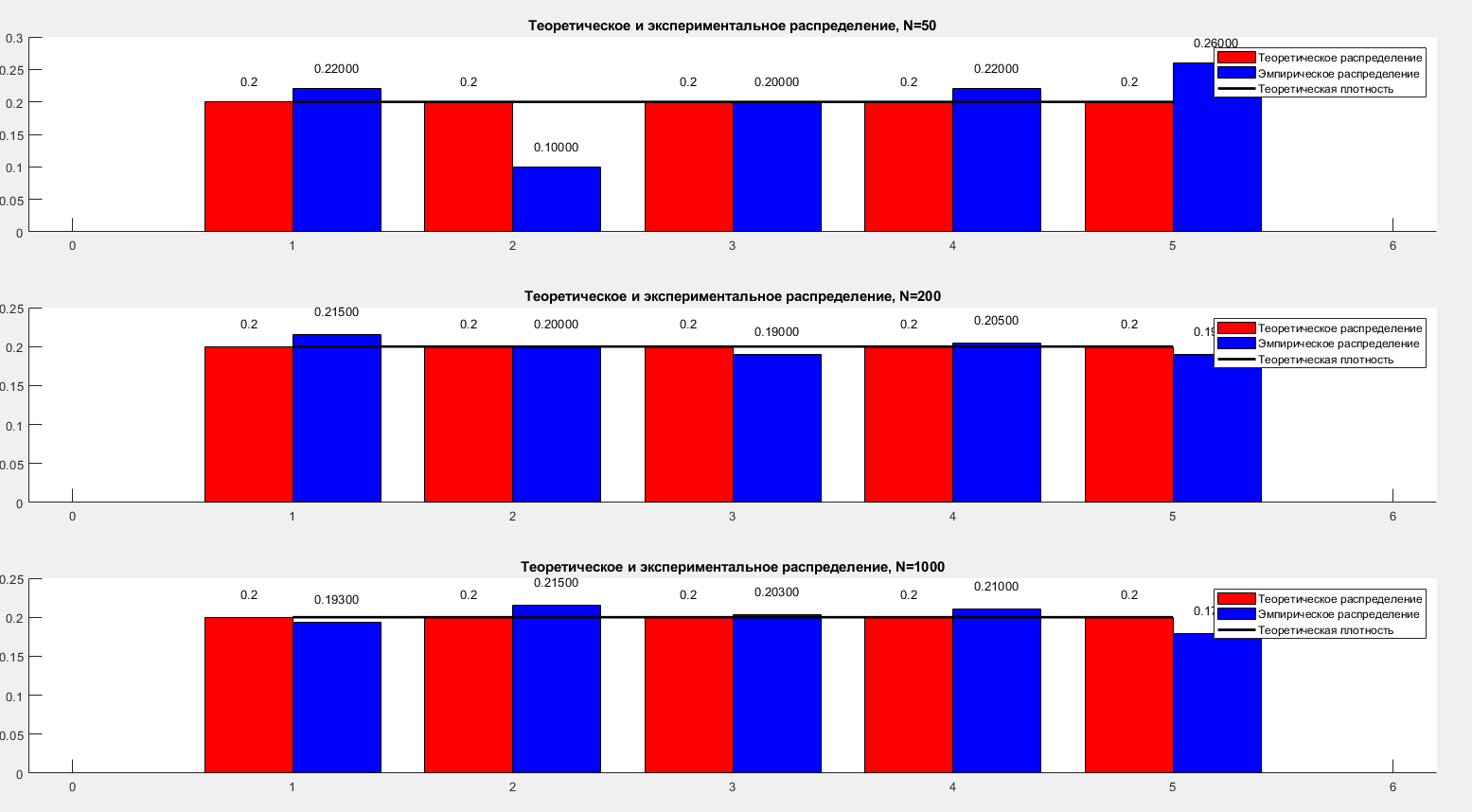


Рисунок 5. График №5. Дискретно распределенная случайная величина.

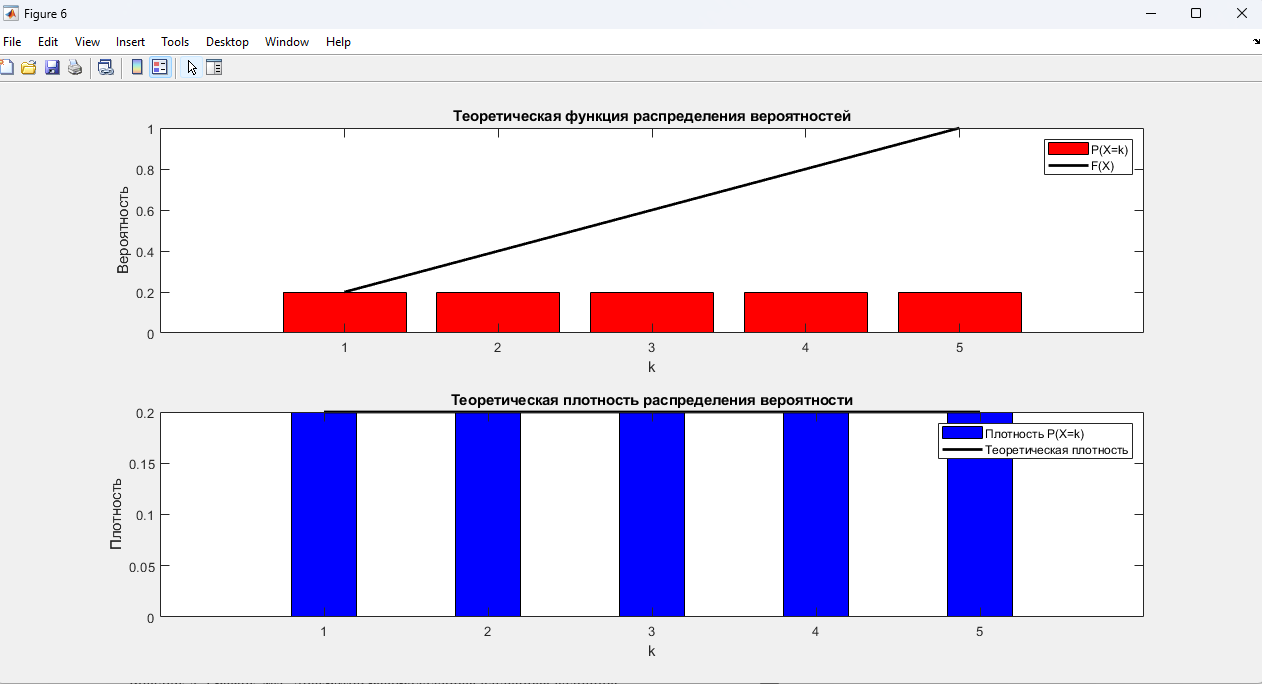
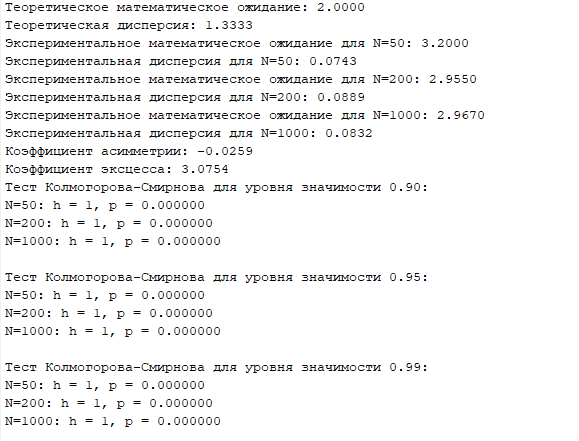


Рисунок 6. График №6. Дискретно распределенная случайная величина.

Таблицы интервальных оценок среднего и дисперсии для выборок:



**6.Дополнительные задания**

****

**7.Выводы**

В данной программе был реализован математический инструмент для генерации случайных величин. Подход к созданию непрерывных случайных величин (НСВ) и дискретных случайных величин (ДСВ) отличался, однако основная идея заключалась в том, что случайные величины, распределенные по определённому закону, получались косвенно. Для НСВ применили метод обратных функций: выразили аргумент функции распределения и подставили в качестве значения сгенерированную величину, следуя нормальному распределению.

При генерации ДСВ использовали нормированную выборку значений, распределённых по нормальному закону, которая задавала вероятность p\_i. Это значение соотносилось с вероятностью p\_k, соответствующей заданной ДСВ. На основе значений p\_k была сформирована выборка искомой ДСВ.

В отчёте представлены гистограммы, иллюстрирующие закон распределения ДСВ: теоретический закон и практические распределения, полученные в процессе генерации.

Для ряда сгенерированных выборок были рассчитаны точечные оценки дисперсии, математического ожидания и стандартного отклонения, а также интервальные оценки среднего и дисперсии для различных уровней значимости. Все данные собраны в таблице и отражены в отчёте. Также были построены гистограммы распределения величин. Интервалы определялись по формуле Стерджесса. На графиках, представленных в отчёте, видно, что распределение сгенерированных случайных величин соответствует теоретическим распределениям по заданному закону.