Лабораторная работа

АНАЛИЗ КАЧЕСТВА ГПСЧ

Цель работы: проанализировать равномерность распределения и статистическую независимость чисел на выходе ГПСЧ.

Краткие сведения из теории построения ГПСЧ

Для анализа качества ГПСЧ применяются различные статистические тесты, выявляющие соответствие ГПСЧ двум основным требованиям: равномерности распределения и независимости генерируемых чисел.

Тестирование равномерности распределения

Равномерность распределения можно проверять с помощью частотного теста. Суть этого теста состоит в построении эмпирического распределения чисел r_i и его сравнении с теоретическим, т.е. равномерным распределением.

Для этого интервал (0,1) возможных значений r_i разбивается на k одинаковых подынтервалов, генерируется выборка $r_1,\ldots, r_n$, для каждого подынтервала определяется количество n_j (j=1,…,k) тех псевдослучайных чисел, которые попали в этот подынтервал, и вычисляются относительные частоты $f_j = (n_j/n)$ попаданий. Для идеального генератора при $n \to \infty$ выполняются условия

$$f_j \to 1/k, \qquad j = 1,\ldots,k, \tag{2.1}$$

т.е. частота f_j попадания в интервал сходится к вероятности $1/k$ попадания в него стандартного случайного числа z.

Из формулы (2.1) следует, что при $n \to \infty$

$$kf_j \to 1, \quad j = 1,...,k, \tag{2.2}$$

т.е. для идеального ГПСЧ эмпирическая плотность вероятностей kf_j на каждом подынтервале сходится к теоретической. Для реального ГПСЧ вычисляют величины kf_j для нескольких достаточно больших значений n и проверяют, приближаются ли они с ростом n к 1.

Косвенная проверка равномерности распределения

Косвенная проверка равномерности распределения может быть осуществлена путем оценивания математического ожидания (МО) и дисперсии псевдослучайных чисел $r_1,\ldots, r_n$. При равномерном распределении на (0,1) **математическое ожидание и дисперсия** равны **1/2 и 1/12,** соответственно. Оценки M (математическое ожидание) и D (дисперсия) для выборки $r_1,\ldots, r_n$ рассчитываются по известным статистическим формулам

$$M = (r_1 + \ldots + r_n)/n = S1/n = 1/2; \tag{2.3}$$

$$D = (r_1^2 + \ldots + r_n^2)/n - M^2 = S2/n - M^2 = 1/12\,, \tag{2.4}$$

где суммы $S1$ и $S2$ накапливаются в процессе генерации выборки. Очевидно, если выборка $r_1,\ldots, r_n$ характеризуется равномерным распределением, то M и D с ростом n должны сходиться к 1/2 и 1/12 соответственно. Это условие является необходимым для равномерности распределения чисел $r_1,\ldots, r_n$, но не достаточным. Поэтому его выполнение лишь косвенно подтверждает (но не доказывает) гипотезу о равномерности распределения.

Проверка статистической независимости

Проверка независимости чисел на выходе ГПСЧ обычно производится путем измерения корреляции между r_i и r_{i+g} , где $g > 0$ – некоторое смещение. В случае, если установлена равномерность распределения, оценка R_g коэффициента корреляции между r_i и r_{i+g} может быть получена по формуле

$$R_g = \frac{M[r_i r_{i+g}]-1/4}{1/12} = 12M[r_i r_{i+g}]\text{-}3 \quad (2.5)$$

где $M[r_i r_{i+g}]$ – оценка математического ожидания произведения чисел r_i и r_{i+g} (т.е. среднее значение произведений пар чисел, отстоящих в выборке на g шагов друг от друга),

$$M[r_i r_{i+g}]= \frac{1}{n-g}(r_1 r_{1+g} +\ldots+ r_n r_{n+g}) =\frac{S_g}{n-g}, \qquad (2.6)$$

где сумму S_g можно накапливать в процессе генерации выборки. В случае независимости псевдослучайных чисел для любого g =1,2,...,n должно выполняться условие $R_g \to 0$ при $n \to \infty$, которое означает попарную некоррелированность чисел. Следует обратить внимание на то, что некоррелированность является необходимым, но не достаточным условием независимости случайных величин.

Проверка длины периода

Поскольку последовательность чисел на выходе мультипликативного конгруэнтного ГПСЧ периодическая, она не может характеризоваться равномерным распределением и независимостью чисел в строгом смысле, но при большой длине l периода этот недостаток не приводит к ошибкам более существенным, чем, скажем, ограниченная длина разрядной сетки ЭВМ.

Выявление периода в последовательности $r_1,\ldots, r_n$ или, что то же самое, в последовательности $x_1,\ldots, x_n$ и определение его длины l – непростая задача, поскольку некоторое число первых членов последовательности может и не принадлежать его периодической части.

Определенную информацию о периодичности можно получить, если запомнить x_1 и в процессе генерации выборки последовательно сравнивать с ним числа x_i, i=2,3,…,n. При первом совпадении $x_i = x_1$ определяется длина периода $l=i-1$. Если для $i \le n$, совпадения не произошло, то либо $l \ge n$, либо периодическая часть последовательности начинается при $i > 1$.

Задание

1. Дополните программу из лабораторной работы ГПСЧ следующим функционалом:

– *тестирование равномерности распределения ГПСЧ частотным методом разбиения на 10 интервалов* (*построить два графика*

$f_i(n)$ *для n=100 и n=10000, где* f_i *– частота встречаемости случайной величины на i-м интервале*);

– *проверка равномерности распределения нахождением математического ожидания и дисперсии по формулам* (*2.3*) *и* (*2.4*) *для n=100 и n=10000*;

– *проверка (дополнительно) статистической независимости по формуле* (*2.5*). *Постройте график* $R_g(n)$ *для n от 10 до 1000*.

2. По рассчитанным оценкам сделайте вывод о пригодности ГПСЧ. Если ГПСЧ не пригоден, подберите для него более подходящие параметры a, m.

Содержание отчета

1. Цель работы.
2. Формулы для расчета статистических оценок, применяемых для анализа качества ГПСЧ.
3. Результаты расчета оценок (скриншоты), показывающие в графическом виде зависимость этих оценок от длины *n* выборки.
4. Вывод о пригодности ГПСЧ для его использования в статистическом моделировании.
5. Листинг программы.

Порядок выполнения работы

1. Открыть проект с программой в C++ Builder 6 (лабораторная работа ГПСЧ).
2. Добавить на форму объект PageControl с вкладки Win32 (рис. 2.1).

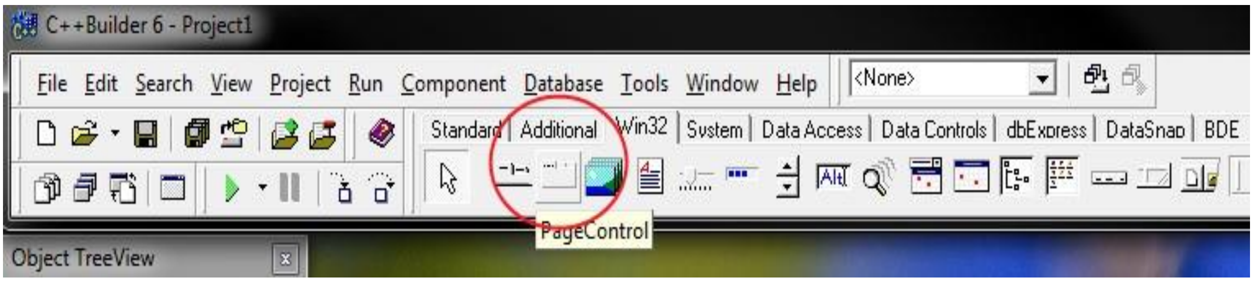

Рис. 2.1. Вкладка с объектами

3. Нажать правой кнопкой мыши на данном объекте и выбрать пункт NewPage (для создания двух вкладок – 2 клика).
4. Растянуть данный объект по размеру формы приложения, оставив свободное место для него перемещением имеющихся объектов в любое свободное пространство на форме (рис. 2.2).

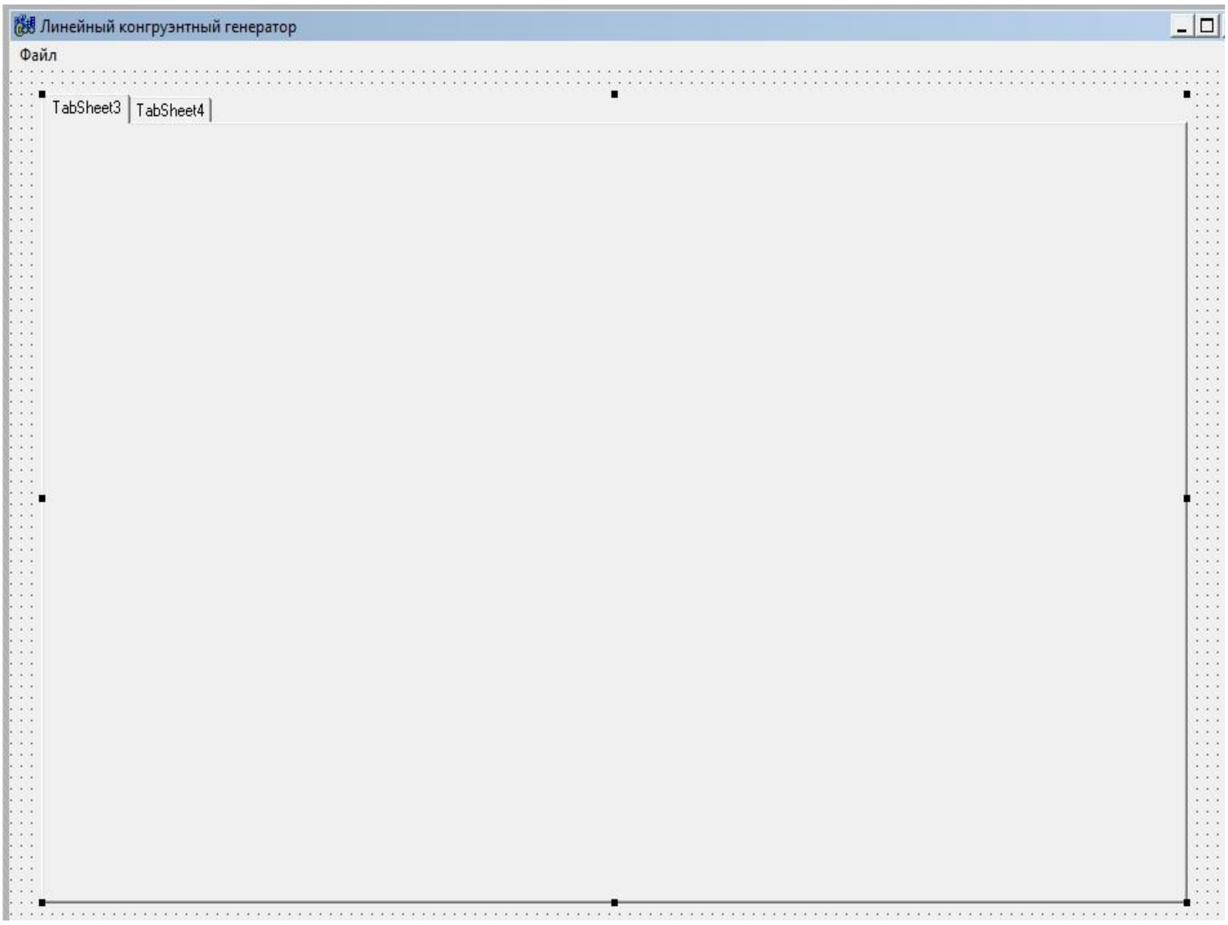

Рис. 2.2. Объект PageControl

5. Переместить все существующие объекты на объект PageControl с использованием выделения и копирования их в буфер обмена сочетанием клавиш ctrl+*x* и вставкой в PageControl с помощью ctrl+*c*.
6. Переименовать вкладки TabSheet1 и TabSheet2 в «Генерация чисел» и «Проверка генератора» соответственно (для этого кликнуть по объекту PageControl и слева в object tree view выбрать TabSheet1, переименовать его в параметре Caption окна Object Inspector на «Генерация чисел») (рис. 2.3).
7. Последующие работы будут добавляться наращиванием функционала данной программы добавлением новых вкладок.

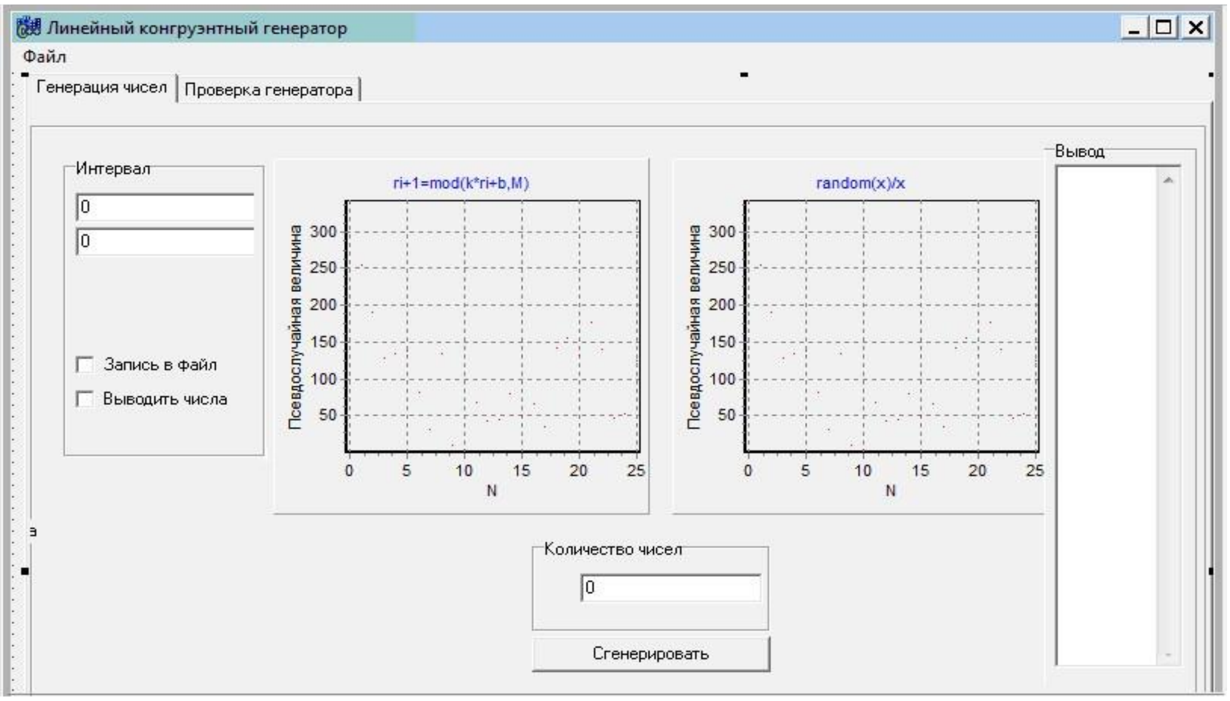

Рис. 2.3. Установка необходимых объектов

8. Перейти на вкладку «Проверка генератора» и установить 3 новых объекта Chart с добавлением в каждый из них по одному графику Series. Дать название каждому из графиков, как указано на рис. 2.4 (двойной клик на объекте Chart, на вкладке Titles).

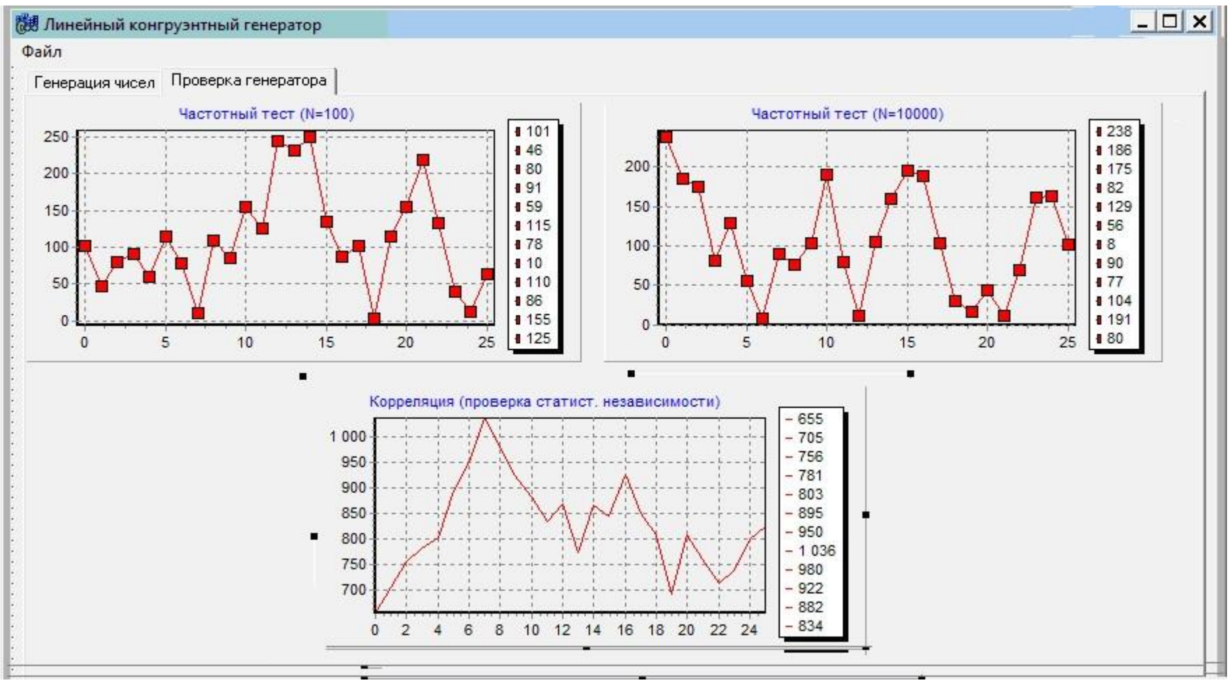

Рис. 2.4. Создание графиков

9. Графики будут соответствовать следующим результатам: первые 2 графика для проверки частотным методом, один график – зависимость частоты встречаемости случайной величины на каждом из интервалов разбиения от номера интервала для N=100, другой – для

N=10000. Третий график будет соответствовать зависимости коэффициента корреляции R_g от N (N следует брать от 10 до 10000).

10. Добавить на форму кнопку Button, в поле caption дать ей название («Начать тестирование ГПСЧ»). Добавить 2 компонента Memo

(рис. 2.5).

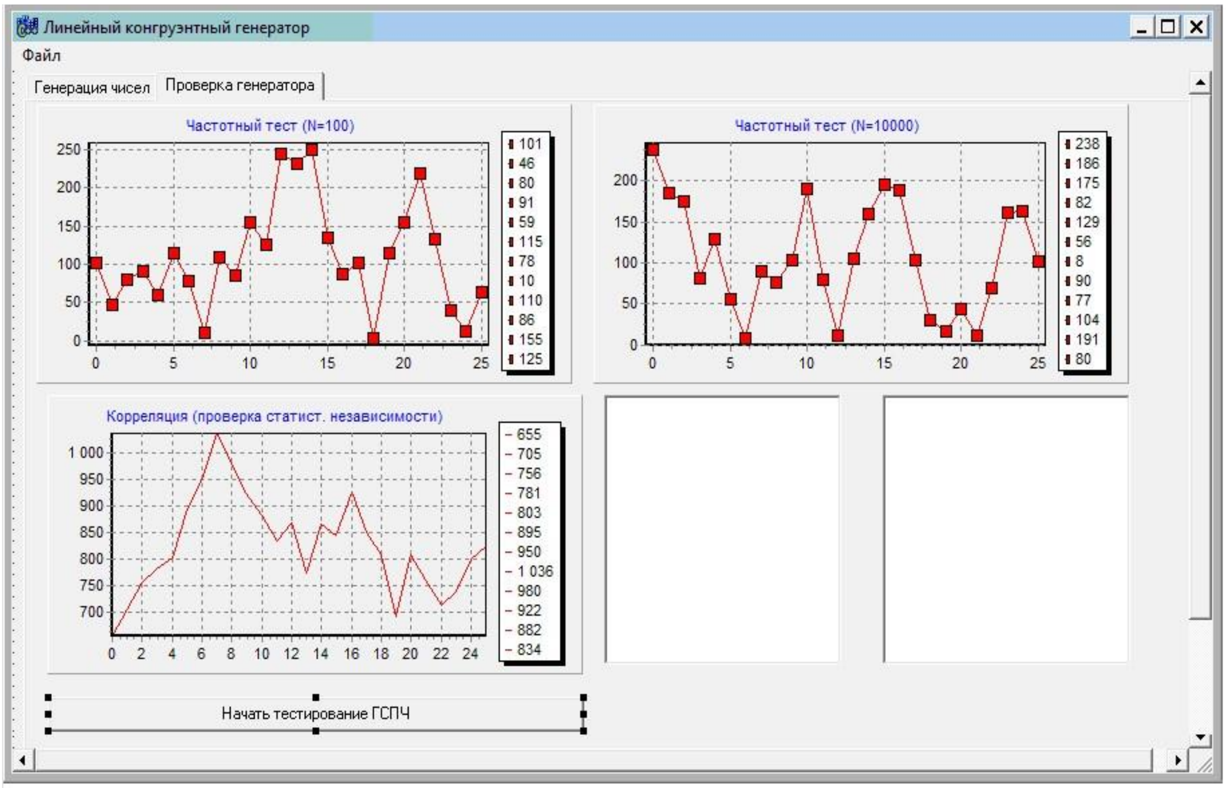

Рис. 2.5. Вывод текста

11. Вам необходимо нажать на кнопку «Начать тестирование ГПСЧ», назначить необходимые действия для заполнения всех трех графиков.

12. Первое, что должна выполнить программа, – сгенерировать первые 100 псевдослучайных чисел. Код можно скопировать с кнопки, генерирующей числа из первой лабораторной работы в кнопку «Начать тестирование ГПСЧ», но с изменением одного параметра – генерации фиксированного количества чисел, равного 100.

13. Код необходимо модифицировать так, чтобы сгенерированные числа записывались в массив. Инициализация массива на 100 чисел в С++ выглядит следующим образом: *float A*[100].

14. По нажатию кнопки заполнение массива псевдослучайными числами можно реализовать следующим образом:

```
…
for (int i=0;
i<100; i++) {
  //Здесь вы генерируете случайное число X с помощью кода из лабораторной работы № 1//
A[i]= X; // присваиваем i-му элементу массива значение X
 }
        ….
```

15. Проверить, правильно ли заполнился массив *A*[*i*] выводом его содержимого в компонент Memo:

```
for (int i=0; i<100; i++)

      Memo2–>Lines->Add(A[i]);
```

16. Следующий этап – подсчет количества случайных величин в интервалах равной длины, составляющих всю сгенерированную последовательность. Разобьем последовательность на 10 интервалов: 0-0,1; 0,1-0,2; 0,2-0,3; 0,3-0,4; 0,4-0,5; 0,5-0,6; 0,6-0,7; 0,7-0,8; 0,8-0,9; 0,9-1,0. Реализация подсчета числа попавших чисел в интервалы может быть получена путем нахождения целого числа от

умножения элемента *A*[*i*] (случайная величина) на 10 и прибавлением единицы. Например, если случайное число 0,25 умножить на 10 и отбросить дробную часть, то мы получим число 2, добавив единицу, и номер интервала, равный 3 (действительно, 0,25 находится в интервале 0,2-0,3) (рис. 2.6). Для проверки правильности подсчета установить еще один компонент Memo и выводить в него значения номеров интервалов. Проинициализировав новый массив B[10], содержащий информацию о каждом из интервалов, можно реализовать подсчет случайных чисел в каждом из интервалов следующим образом (выделение целой части можно сделать указанием *(int)* перед выводом числа): *for* (*int i*=0;*i*<100;*i*++)

{

Memo2–>Lines–>Add(*A*[*i*]); *//выводим случайную величину*

buff=*A*[*i*]*10+1; *// определяем номер интервала, которому она принадлежит*

Memo3–>Lines–>Add((int)buff); //выводим номер этого интервала

B[(int)buff]+=1*; //добавляем к счетчику единицу*

}

Случайная величина *Номер интервала*

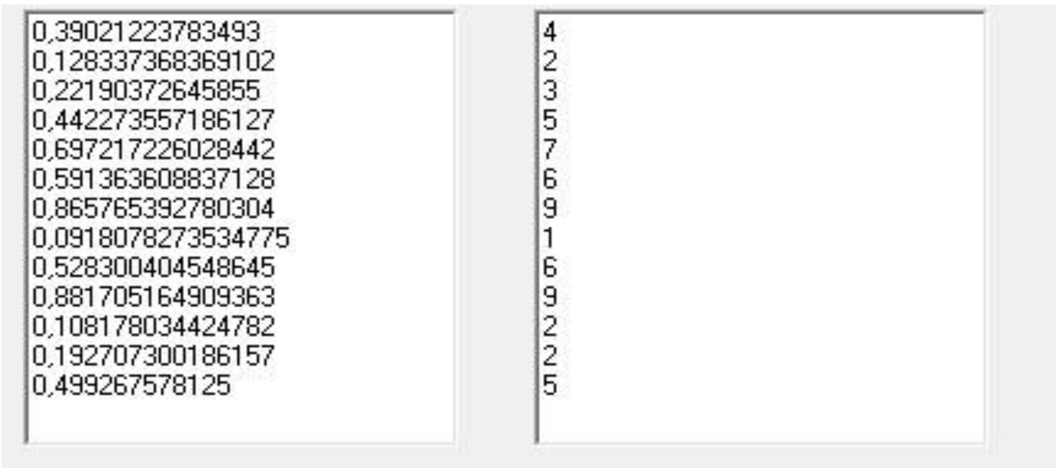

Рис. 2.6. Вывод значений случайной величины

17. В массиве *B*[*i*] теперь подсчитано количество попавших чисел для каждого из интервалов. Выведем на первый график значения из массива *B*:

for (*int i*=0;*i*<10;*i*++)

Series4->AddXY(i,B[*i*]/100); *// отношение попавших в интервал значений к общему их числу(определение f_i).*

18. Сделать копию блока кода для *N*=100 и поместить его следующим за текущим.
19. В скопированном блоке внести изменения, которые позволят генерировать 10000 значений и подсчитать их число в каждом из десяти интервалов (заменить в циклах 100 на 10000).
20. Отобразить на втором графике частоты встречаемости случайных величин на каждом из 10 интервалов (рис. 2.7).
21. Как видим, при переходе к *N*=10000 частотная составляющая приближается к вероятности попадания на каждый из интервалов, а именно 1/*k*=0,1.

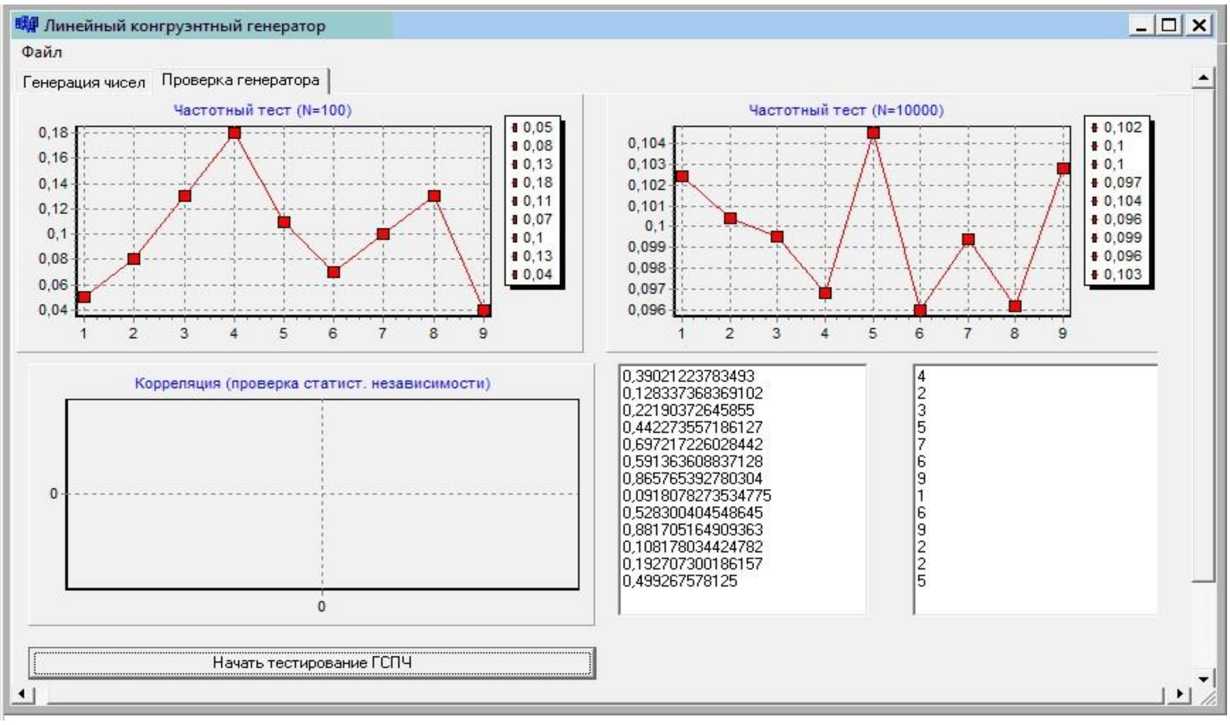

Рис. 2.7. Интервалы

22. Поместить на форму 4 компонента Edit для вывода значений математического ожидания и дисперсии для N=100 и N=10000
(рис. 2.8).

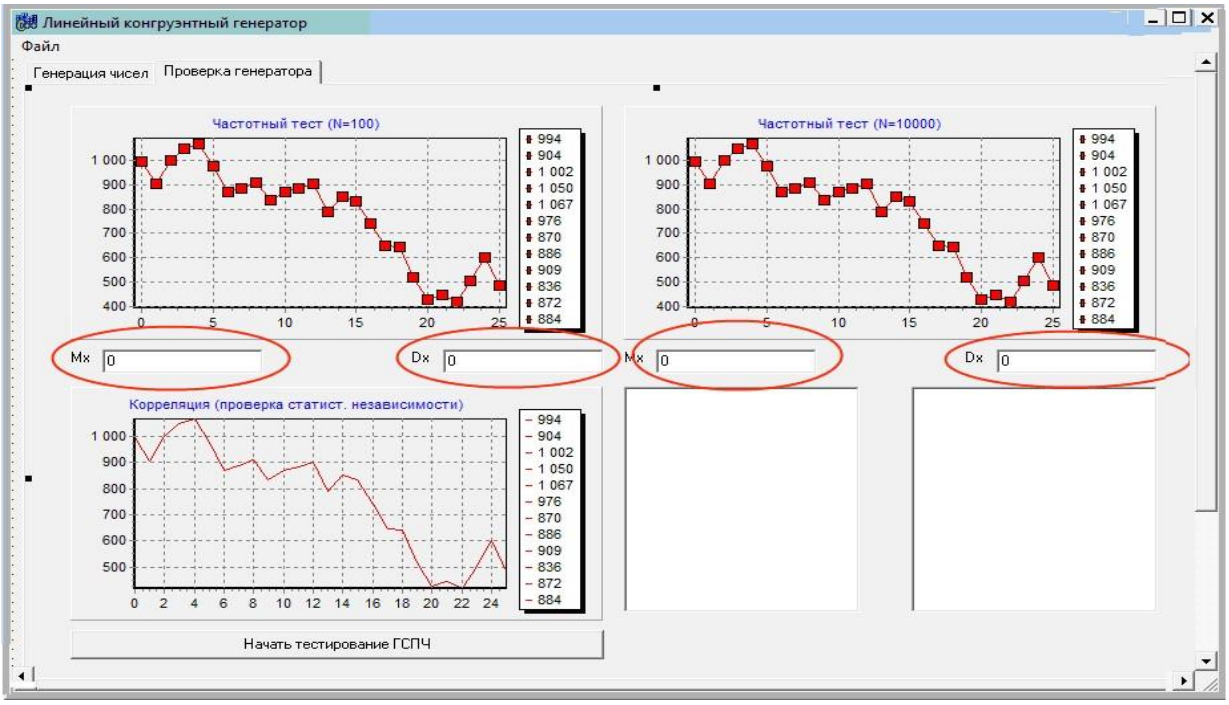

Рис. 2.8. Вывод значений математического ожидания и дисперсии

23. Найти значения математического ожидания для N=100 и N=10000 по формуле (2.3). Выводить полученные значения можно следующим образом:

Edit–9–>Text=Mx; // Mx – найденное математическое ожидание для 100 *чисел, Edit может быть под другим номером*

Edit10–>Text=Mx;// найденное математическое ожидание для 10000 *чисел, Edit может быть под другим номером.*

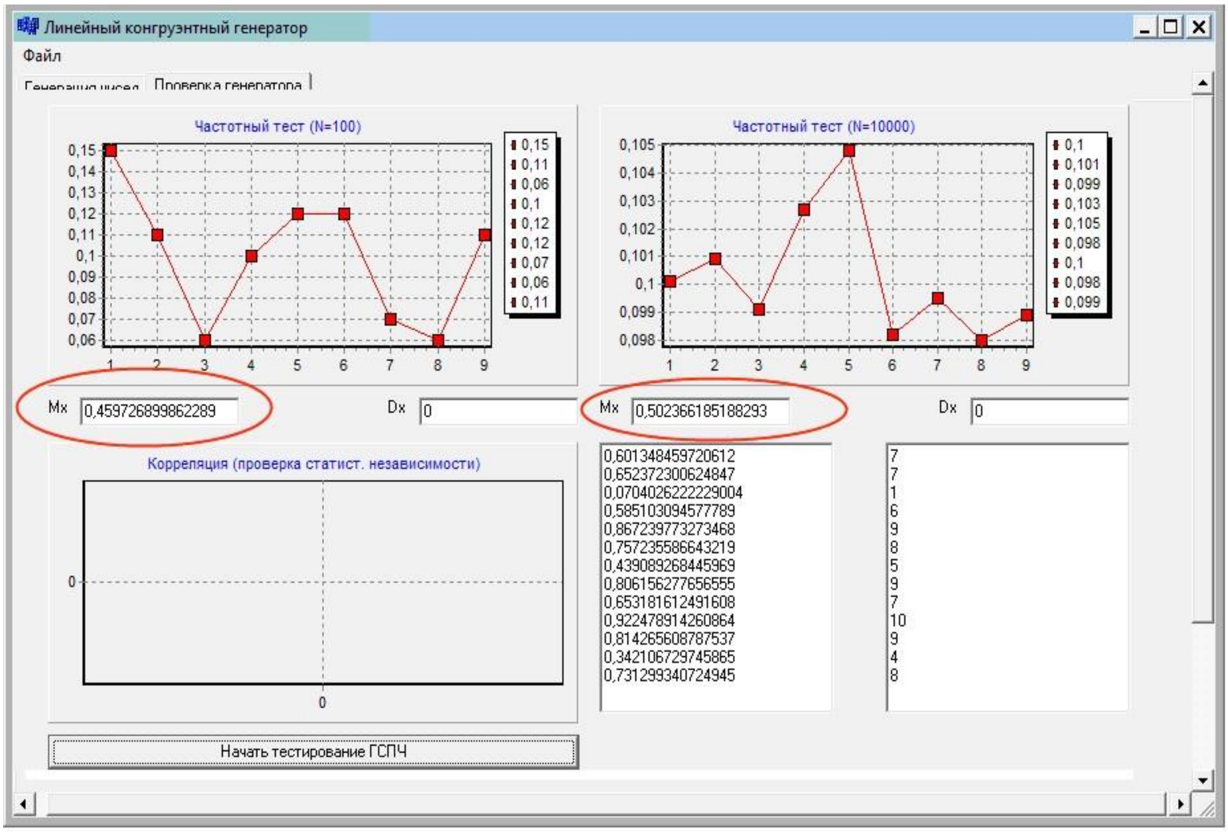

Рис. 2.9. Вывод значений математического ожидания

24. Сгенерировать несколько выборок и следить за получаемыми значениями *Mx*. Для *N*=10000 они должны быть ближе к 0,5, чем при *N*=100 (рис. 2.9).

25. В остальные 2 поля выводить значения дисперсии для *N*=100 и *N*=10000 по формуле (2.4).

26. Значение дисперсии с ростом *N* должно стремиться к

0,083333333 (1/12) и быть ближе к нему у выборки с *N*=10000 (рис. 2.10).

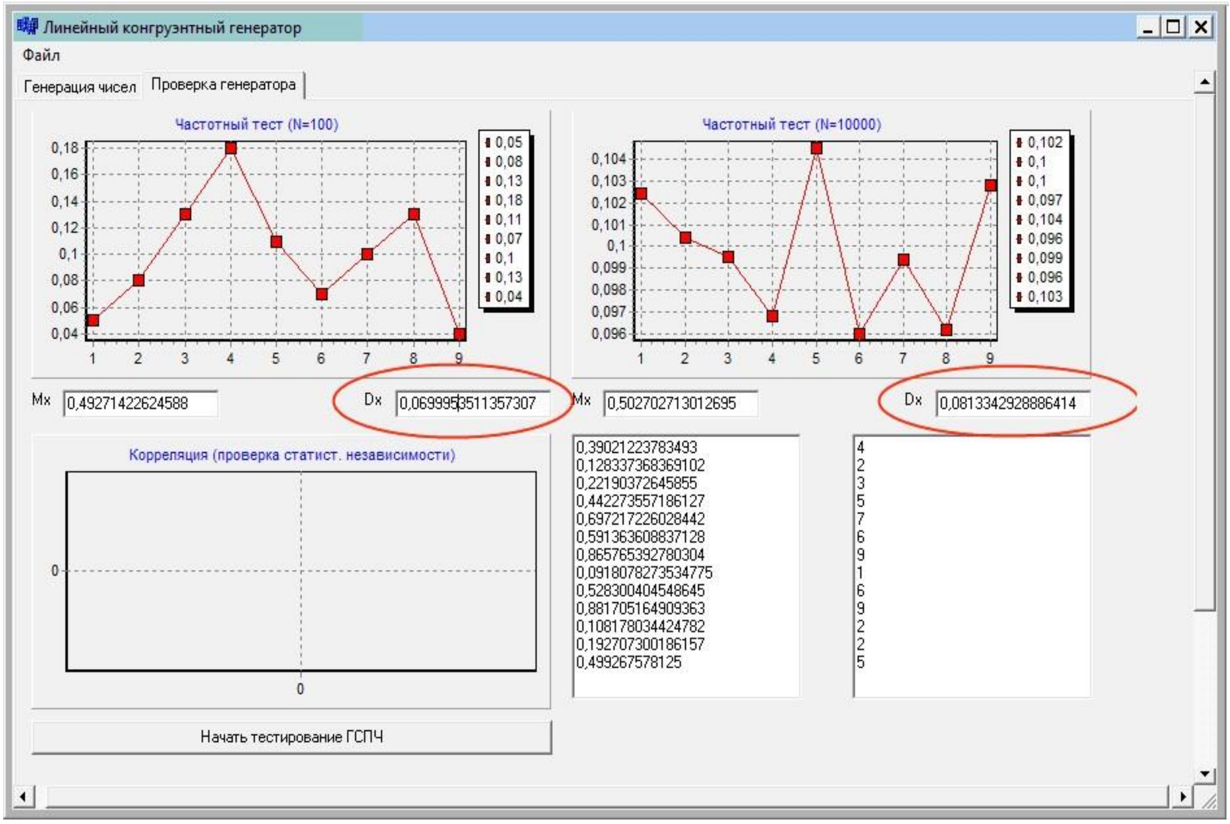

Рис. 2.10. Вывод значений дисперсии

27. Рассчитать (дополнительно) значение R_g корреляции между парами чисел из выборки (g взять равным единице) при N от 10 до 1000 [сгенерировать 10 чисел, рассчитать R_g по формуле (2.5), вывести это значение на график, потом сгенерировать 11 чисел, рассчитать R_g и вывести на график, потом 12, 13 чисел и т.д. до 1000]. Полученная кривая должна отражать приближение коэффициента корреляции R_g к нулю с ростом N (рис. 2.11).

28. Подписать оси на вкладке Axis каждого из графиков, как указано на рис. 2.11.

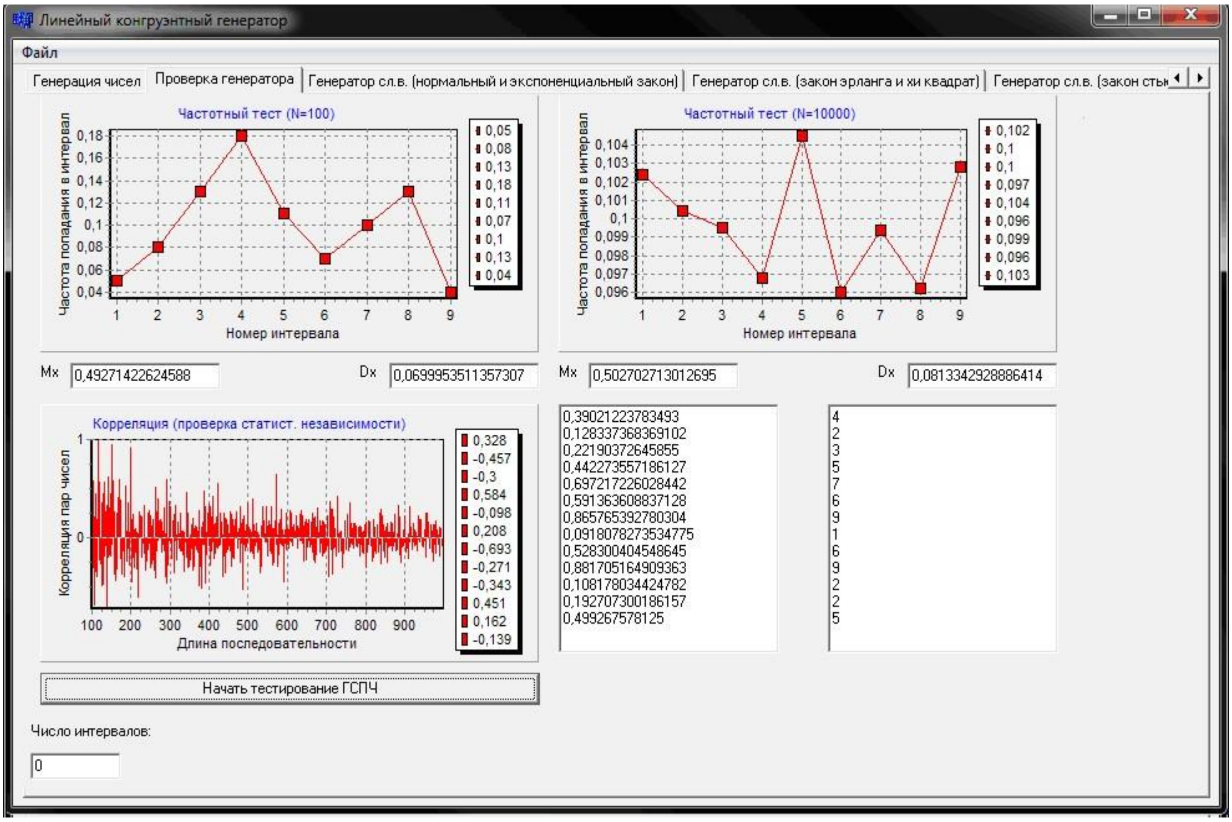

Рис. 2.11. Коэффициент корреляции. Итоговый результат

Контрольные вопросы

1. Как по выборке случайной величины рассчитываются оценки ее математического ожидания и дисперсии?
2. Как проверяется равномерность распределения чисел с помощью частотного теста?
3. Как проверяется статистическая независимость чисел на выходе ГПСЧ?