

[Описание программы](#)

[Код программы](#)

[Код с подробными комментариями](#)

[Советы по улучшению и расширению программы](#)

Описание программы

Насколько хорош ваш генератор случайных чисел? Для повседневных нужд хватает, но... Прямо скажем - стандартный генератор, включенный в наш родной C++ Builder для моделирования квантовых систем будет мало пригоден. Данная программа демонстрирует, насколько тот или иной генератор случайных чисел отличается от идеального (на примере функции `random` из библиотеки `stdlib`).

Рассмотрим подробнее, что эта программа делает. Первым делом мы моделируем случайное бросание камешка в одну из клеток квадратного поля, в данном случае - размером 15*15. По идее, если работа поручена очень хорошему генератору случайных чисел (ГСЧ), то при большом числе бросаний количество попаданий в каждую из клеток должно быть приблизительно одинаковым и с ростом числа бросков различие должно становиться все меньше. Так это или нет в случае вполне конкретной функции, реализующей ГСЧ, нам покажет наша программа на первом этапе своего отчета о проделанной работе (рис. 1). Впрочем, если среднестатистический пользователь применяет для обработки этих значений только свою голову, то он сможет получить из них лишь общее представление - мол, даже при миллионе бросаний числа явно не одинаковые. Поэтому после того, как этот самый пользователь достаточно проникнется важностью циферок и нажмет любую клавишу, программа выдаст ему более наглядное представление о "шероховатостях" псевдослучайного распределения (рис. 2). Этот скромный график показывает распределение числа попаданий в какую-либо клетку. В идеале во всех клетках должно быть примерно одинаковое число попаданий и этот график должен иметь ярко выраженный максимум (практически нет клеток, в которые бы камешки попадали чаще или реже, чем в остальные).

Ну и что же мы видим на самом деле? Число попаданий в клетку, конечно, тяготеет к некоему среднему числу, но разброс все же очень сильный. График охватывает полосу шириной в 500 "попаданий" при среднем 4444 при данных параметрах программы. Получаем среднее число попаданий (4444 +/- 250), т.е. разброс порядка 5%. Такого распределения неравномерностей в генерации псевдослучайных чисел у функции `random` в моей версии C++ Builder'a. Для генерации небольшого числа псевдослучайных значений в обычных задачах - вполне достаточно, однако если перед вами будет стоять задача по моделированию сложных стохастических процессов, то вам придется искать специальный *хороший* ГСЧ.

Тест для генератора случайных чисел

Автор: Андрей

22.01.2010 16:09 - Обновлено 25.01.2010 22:04

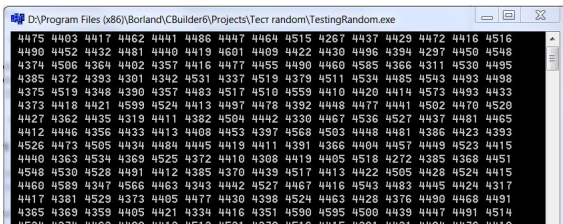


Рис. 1

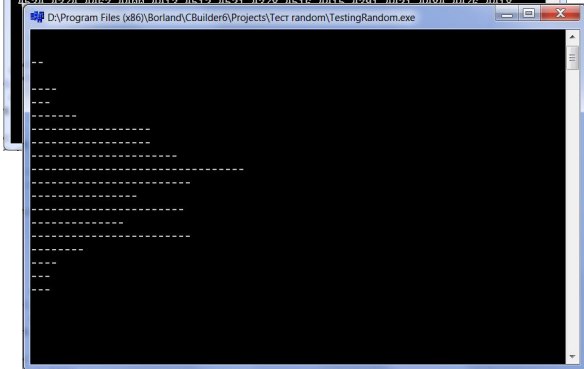


Рис. 2

Код программы

```
#include <iostream>
#include <stdlib.h>
#include <conio.h>
```

```
int main()
{
    const int NX = 15;
    const int NY = 15;
    int Nums[NX][NY];
```

```
    for (int i = 0; i < NX; i++)
        for (int j = 0; j < NY; j++)
            Nums[i][j] = 0;

    randomize();
    int a, b;
    for (int i = 0; i < NX; i++)
    {
        a = random(NX);
        b = random(NY);
        Nums[a][b]++;
    }

    for (int i = 0; i < NX; i++)
    {
        for (int j = 0; j < NY; j++)
        {
            std::cout.width(5);
            std::cout
```