Random class

https://csci-1301.github.io/about#authors

February 23, 2022 (11:03:26 AM)

Contents

The `Random` class from the C# standard library can be used to generate random numbers in any given range. In this lab you will practice using `Random` class.

1 Generating random numbers

Start by creating a new project, then practice generating different random numbers and display them at the screen:

1. Generate any random integer
2. Generate a random integer between -10 and 10 including these boundary values
3. Generate a random double

Note you only need 1 instance of `Random` class to generate these numbers.

Execute the program a few times to make sure the outputs are different each time.

Once you have successfully generated the 3 random numbers described above, add the following enhancements to the program:

1. Generate any random integer *5 times*
2. Generate a random integer between -10 and 10 including these boundary values, *10 times*
3. Generate a random double, *5 times*

Execute the program again, a few times, to make sure these values change on each execution.

2 Manipulating two arrays

This problem combines random number generation with arrays. Using a Random object, write a program that:

1. declares two arrays of `int` of size `8`,
2. initializes the values of the first array with random numbers between 0 and 9,
3. initializes the values of second array with random numbers between 0 and 9,
4. displays the contents of the two arrays in a table, and, for each index, a letter indicating whether the first array "won" or "lost" a contest with the second array:
   - `"W"` if the value in the first array is greater than the value in the second array
   - `"T"` if they are equal, and
   - `"L"` if it is less

An example execution of this program would display:

```
0       8       L
5       3       W
3       3       T
1       2       L
3       1       W
9       0       W
9       0       W
1       5       L
```

In this example, the first array contains "0 5 3 1 3 9 9 1" and the second contains "8 3 3 2 1 0 0 5".

3 "Higher or Lower" Game

Write a program that:

1. Starts by having computer choose a random number between 0 and 100. Store that number at a variable.
2. Asks the user to enter a numerical value between 0 and 100 and stores the user's answer in a variable.
3. Add an if statement that displays on the screen `"You guessed correctly"` if the number entered by the user matches the number selected by computer.
4. Add an if statement that displays on the screen `"Too high!"` if the number entered by the user is strictly greater than the number selected by computer.
5. Add an if statement that displays on the screen `"Too low!"` if the number entered by the user is strictly smaller than the number selected by computer.
6. Add an if statement that displays on the screen `"You found a multiple !"` if the number entered by the user is a multiple of the number selected by computer, but different from it.
7. When user enters a number that does not match computers selection, the user should be able to enter another number. This continues until user guesses correctly.
8. Keep count of how many guesses it takes for the user to arrive to the correct answer. After user guesses correctly, display that count, for example:

```
You guessed correctly.
That was 7 guesses!
```

You can adjust your program so that e.g. if the user enters a number that is at the same time higher and a multiple of computer's number, only one message is displayed.

4 Pushing Further (Optional)

4.1 Cryptographically secure random numbers

Random number generation is only pseudo-random, meaning these are algorithmically generated numbers that approximate a sequence of truly random numbers. Using the default `Random` class is not recommended for applications that need cryptographically secure random numbers, for example, to generate suggested passwords.

When an application needs cryptographically secure random numbers, RandomNumberGenerator class should be used instead. It works as follows:

```
using System;
using System.Security.Cryptography; // include definition!

class Program
{
    static void Main()
    {

        // choose secure (!) random integer
        // between 0 (inclusive) and 100 (exclusive)
        int secureRandom = RandomNumberGenerator.GetInt32(100);

        // display cryptographically secure int
        Console.WriteLine(secureRandom);

        // choose secure (!) random integer
        // between 50 (inclusive) and 500 (exclusive)
        int anotherSecureRandom = RandomNumberGenerator.GetInt32(50, 500);

        // display cryptographically secure int
        Console.WriteLine(anotherSecureRandom);
    }
}
```

You can learn more about secure random numbers by reading through:

- the technical description of RandomNumberGenerator class[1]
- discussion on how to choose which Random generator to use[2]

[1] https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.randomnumbergenerator
[2] https://stackoverflow.com/questions/1257299/why-use-the-c-sharp-class-system-random-at-all-instead-of-system-security-crypto