

LinksPlatform's Platform.Random Class Library

./RandomExtensions.cs

```
1 using System.Runtime.CompilerServices;
2 using Platform.Ranges;
3
4 namespace Platform.Random
5 {
6     /// <summary>
7     /// Contains extension methods for Random class.
8     /// Содержит методы расширения для класса Random.
9     /// </summary>
10    public static class RandomExtensions
11    {
12        /// <summary>
13        /// Returns a random 64-bit unsigned integer that is greater than or equal to
14        /// → ulong.MinValue, and less than or equal to ulong.MaxValue.
15        /// Возвращает случайное 64-разрядное целое число без знака, которое больше или равно
16        /// → ulong.MinValue и меньше или равно ulong.MaxValue.
17        /// </summary>
18        /// <param name="rnd">A pseudo-random number generator. Генератор псевдослучайных
19        /// → чисел.</param>
20        /// <returns>A 64-bit unsigned integer that is greater than or equal to ulong.MinValue,
21        /// → and less than or equal to ulong.MaxValue. 64-разрядное целое число без знака,
22        /// → которое больше или равно ulong.MinValue и меньше или равно ulong.MaxValue.</returns>
23        [MethodImpl(MethodImplOptions.AggressiveInlining)]
24        public static ulong NextUInt64(this System.Random rnd) => rnd.NextUInt64(new
25        → Range<ulong>(ulong.MinValue, ulong.MaxValue));
26
27        /// <summary>
28        /// Returns a random 64-bit unsigned integer that is greater than or equal to minimum of
29        /// → specified range, and less than or equal to maximum of specified range.
30        /// Возвращает случайное 64-разрядное целое число без знака, которое больше или равно
31        /// → минимуму указанного диапазона и меньше или равно максимуму указанного диапазона.
32        /// </summary>
33        /// <param name="rnd">A pseudo-random number generator. Генератор псевдослучайных
34        /// → чисел.</param>
35        /// <param name="range">The range of possible values. Диапазон возможных
36        /// → значений.</param>
37        /// <returns>A 64-bit unsigned integer that is greater than or equal to the minimum of
38        /// → specified range, and less than or equal to the maximum of the specified range.
39        /// → 64-разрядное целое число без знака, которое больше или равно минимуму указанного
40        /// → диапазона и меньше или равно максимуму указанного диапазона.</returns>
41        [MethodImpl(MethodImplOptions.AggressiveInlining)]
42        public static ulong NextUInt64(this System.Random rnd, Range<ulong> range) =>
43        → (ulong)(rnd.NextDouble() * (range.Maximum - range.Minimum)) + range.Minimum;
44
45        /// <summary>
46        /// Return a random boolean value.
47        /// Возвращает случайное булево значение.
48        /// </summary>
49        /// <param name="rnd">A pseudo-random number generator. Генератор псевдослучайных
50        /// → чисел.</param>
51        /// <returns>A random boolean value. Случайное булево значение.</returns>
52        [MethodImpl(MethodImplOptions.AggressiveInlining)]
53        public static bool NextBoolean(this System.Random rnd) => rnd.Next(2) == 1;
54    }
55 }
```

./RandomHelpers.cs

```
1 namespace Platform.Random
2 {
3     /// <summary>
4     /// Contains field-helper for Random class.
5     /// Содержит вспомогательное поле для класса Random.
6     /// </summary>
7     public static class RandomHelpers
8     {
9         /// <summary>
10        /// Returns the pseudorandom number generator that is using the time of the first access
11        /// → to this field as seed.
12        /// Возвращает генератор псевдослучайных чисел использующий в качестве seed время
13        /// → первого обращения к этому полю.
14        /// </summary>
15        public static readonly System.Random Default = new
16        → System.Random(System.DateTime.UtcNow.Ticks.GetHashCode());
17    }
18 }
```

Index

- ./RandomExtensions.cs, 1
- ./RandomHelpers.cs, 1