

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     /// <para>Contains extension methods for Random class.</para>
8     /// <para>Содержит методы расширения для класса Random.</para>
9     /// </summary>
10    public static class RandomExtensions
11    {
12        /// <summary>
13        /// <para>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.</para>
15        /// <para>Возвращает случайное 64-разрядное целое число без знака, которое больше или
16        ///     → равно ulong.MinValue и меньше или равно ulong.MaxValue.</para>
17        /// </summary>
18        /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
19        ///     → псевдослучайных чисел.</para></param>
20        /// <returns>
21        /// <para>A 64-bit unsigned integer that is greater than or equal to ulong.MinValue, and
22        ///     → less than or equal to ulong.MaxValue.</para>
23        /// <para>64-разрядное целое число без знака, которое больше или равно ulong.MinValue и
24        ///     → меньше или равно ulong.MaxValue.</para>
25        /// </returns>
26        [MethodImpl(MethodImplOptions.AggressiveInlining)]
27        public static ulong NextUInt64(this System.Random random) =>
28            → random.NextUInt64((ulong.MinValue, ulong.MaxValue));
29
30        /// <summary>
31        /// <para>Returns a random 64-bit unsigned integer that is greater than or equal to
32        ///     → minimum of specified range, and less than or equal to maximum of specified
33        ///     → range.</para>
34        /// <para>Возвращает случайное 64-разрядное целое число без знака, которое больше или
35        ///     → равно минимуму указанного диапазона и меньше или равно максимуму указанного
36        ///     → диапазона.</para>
37        /// </summary>
38        /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
39        ///     → псевдослучайных чисел.</para></param>
40        /// <param name="range"><para>The range of possible values.</para><para>Диапазон
41        ///     → возможных значений.</para></param>
42        /// <returns>
43        /// <para>A 64-bit unsigned integer that is greater than or equal to the minimum of
44        ///     → specified range, and less than or equal to the maximum of the specified range.</para>
45        /// <para>64-разрядное целое число без знака, которое больше или равно минимуму
46        ///     → указанного диапазона и меньше или равно максимуму указанного диапазона.</para>
47        /// </returns>
48        [MethodImpl(MethodImplOptions.AggressiveInlining)]
49        public static ulong NextUInt64(this System.Random random, Range<ulong> range) =>
50            → (ulong)(random.NextDouble() * range.Difference()) + range.Minimum;
51
52        /// <summary>
53        /// <para>Return a random boolean value.</para>
54        /// <para>Возвращает случайное булево значение.</para>
55        /// </summary>
56        /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
57        ///     → псевдослучайных чисел.</para></param>
58        /// <returns><para>A random boolean value.</para><para>Случайное булево
59        ///     → значение.</para></returns>
60        [MethodImpl(MethodImplOptions.AggressiveInlining)]
61        public static bool NextBoolean(this System.Random random) => random.Next(2) == 1;
62    }
63 }
```

./RandomHelpers.cs

```
1 namespace Platform.Random
2 {
3     /// <summary>
4     /// <para>Contains field-helper for Random class.</para>
5     /// <para>Содержит вспомогательное поле для класса Random.</para>
6     /// </summary>
7     public static class RandomHelpers
8     {
9         /// <summary>
```

```
10      /// <para>Returns the pseudorandom number generator that is using the time of the first
    ↳ access to this field as seed.</para>
11      /// <para>Возвращает генератор псевдослучайных чисел использующий в качестве seed время
    ↳ первого обращения к этому полю.</para>
12      /// </summary>
13      public static readonly System.Random Default = new
    ↳ System.Random(System.DateTime.UtcNow.Ticks.GetHashCode());
14  }
15 }
```

Index

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