

## LinksPlatform's Platform.Random Class Library

### 1.1 ./csharp/Platform.Random/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 <see cref="System.Random"/> class.</para>
8     /// <para>Содержит методы расширения для класса <see cref="System.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 <see
14        → cref="ulong.MinValue"/>, and less than or equal to <see
15        → cref="ulong.MaxValue"/>.</para>
16        /// <para>Возвращает случайное 64-разрядное целое число без знака, которое больше или
17        → равно <see cref="ulong.MinValue"/> и меньше или равно <see
18        → cref="ulong.MaxValue"/>.</para>
19        /// </summary>
20        /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
21        → псевдослучайных чисел.</para></param>
22        /// <returns>
23        /// <para>A 64-bit unsigned integer that is greater than or equal to <see
24        → cref="ulong.MinValue"/>, and less than or equal to <see
25        → cref="ulong.MaxValue"/>.</para>
26        /// <para>64-разрядное целое число без знака, которое больше или равно <see
27        → cref="ulong.MinValue"/> и меньше или равно <see cref="ulong.MaxValue"/>.</para>
28        /// </returns>
29        [MethodImpl(MethodImplOptions.AggressiveInlining)]
30        public static ulong NextUInt64(this System.Random random) =>
31        → random.NextUInt64(Range.UInt64);
32
33        /// <summary>
34        /// <para>Returns a random 64-bit unsigned integer that is greater than or equal to
35        → minimum of specified range, and less than or equal to maximum of specified
36        → range.</para>
37        /// <para>Возвращает случайное 64-разрядное целое число без знака, которое больше или
38        → равно минимуму указанного диапазона и меньше или равно максимуму указанного
39        → диапазона.</para>
40        /// </summary>
41        /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
42        → псевдослучайных чисел.</para></param>
43        /// <param name="range"><para>The range of possible values.</para><para>Диапазон
44        → возможных значений.</para></param>
45        /// <returns>
46        /// <para>A 64-bit unsigned integer that is greater than or equal to the minimum of
47        → specified range, and less than or equal to the maximum of the specified range.</para>
48        /// <para>64-разрядное целое число без знака, которое больше или равно минимуму
49        → указанного диапазона и меньше или равно максимуму указанного диапазона.</para>
50        /// </returns>
51        [MethodImpl(MethodImplOptions.AggressiveInlining)]
52        public static ulong NextUInt64(this System.Random random, Range<ulong> range) =>
53        → (ulong)(random.NextDouble() * range.Difference()) + range.Minimum;
54
55        /// <summary>
56        /// <para>Return a random <see cref="bool"/> value.</para>
57        /// <para>Возвращает случайное значение <see cref="bool"/>.</para>
58        /// </summary>
59        /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
60        → псевдослучайных чисел.</para></param>
61        /// <returns><para>A random <see cref="bool"/> value.</para><para>Случайное значение
62        → <see cref="bool"/>.</para></returns>
63        [MethodImpl(MethodImplOptions.AggressiveInlining)]
64        public static bool NextBoolean(this System.Random random) => random.Next(2) == 1;
65    }
66 }
```

### 1.2 ./csharp/Platform.Random/RandomHelpers.cs

```
1 namespace Platform.Random
2 {
3     /// <summary>
4     /// <para>Contains field-helper for <see cref="System.Random"/> class.</para>
5     /// <para>Содержит вспомогательное поле для класса <see cref="System.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
11        ↪ access to this field as seed.</para>
12        /// <para>Возвращает генератор псевдослучайных чисел использующий в качестве seed время
13        ↪ первого обращения к этому полю.</para>
14        /// </summary>
15        public static readonly System.Random Default = new
16        ↪ System.Random(System.DateTime.UtcNow.Ticks.GetHashCode());
17    }
18 }

```

### 1.3 ./csharp/Platform.Random.Tests/RandomExtensionsTests.cs

```

1  using Xunit;
2
3  namespace Platform.Random.Tests
4  {
5      public class RandomExtensionsTests
6      {
7          [Fact]
8          public void NextUInt64Test()
9          {
10             var lastValue = 0UL;
11             var theSameCount = 0;
12             for (var i = 0; i < 10; i++)
13             {
14                 var newValue = RandomHelpers.Default.NextUInt64();
15                 if (newValue == lastValue)
16                 {
17                     theSameCount++;
18                 }
19                 else
20                 {
21                     lastValue = newValue;
22                     theSameCount = 0;
23                 }
24                 Assert.InRange(RandomHelpers.Default.NextUInt64((0UL, 5UL)), 0UL, 5UL);
25             }
26             Assert.True(theSameCount < 8);
27         }
28
29         [Fact]
30         public void NextBooleanTest()
31         {
32             var trueCount = 0;
33             var falseCount = 0;
34             for (var i = 0; i < 10; i++)
35             {
36                 var newValue = RandomHelpers.Default.NextBoolean();
37                 if (newValue)
38                 {
39                     trueCount++;
40                 }
41                 else
42                 {
43                     falseCount++;
44                 }
45             }
46             Assert.True(trueCount > 0);
47             Assert.True(falseCount > 0);
48         }
49     }
50 }

```

### 1.4 ./csharp/Platform.Random.Tests/RandomHelpersTests.cs

```

1  using Xunit;
2
3  namespace Platform.Random.Tests
4  {
5      public class RandomHelpersTests
6      {
7          [Fact]
8          public void DefaultFieldTest()
9          {
10             Assert.NotNull(RandomHelpers.Default);
11         }
12     }
13 }

```

## Index

- ./csharp/Platform.Random.Tests/RandomExtensionsTests.cs, 2
- ./csharp/Platform.Random.Tests/RandomHelpersTests.cs, 2
- ./csharp/Platform.Random/RandomExtensions.cs, 1
- ./csharp/Platform.Random/RandomHelpers.cs, 1