```
LinksPlatform's Platform.Random Class Library
    ./csharp/Platform.Random/RandomExtensions.cs
   using System.Runtime.CompilerServices;
   using Platform.Ranges;
   namespace Platform.Random
4
5
        /// <summary>
6
       /// <para>Contains extension methods for <see cref="System.Random"/> class.</para>
       /// <para>Содержит методы расширения для класса <see cref="System.Random"/>.</para>
       /// </summary>
       public static class RandomExtensions
10
11
            /// <summary>
12
           /// <para>Returns a random 64-bit unsigned integer that is greater than or equal to <see
13
               cref="ulong.MinValue"/>, and less than or equal to <see</pre>
               cref="ulong.MaxValue"/>.</para>
            /// <para>Возвращает случайное 64-разрядное целое число без знака, которое больше или
            _{
ightarrow} равно <see cref="ulong.MinValue"/> и меньше или равно <see
               cref="ulong.MaxValue"/>.</para>
            /// </summary>
            /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
16
               псевдослучайных чисел.</para></param>
            /// <returns>
17
            /// <para>A 64-bit unsigned integer that is greater than or equal to <see
18
            _{
ightharpoonup} cref="ulong.MinValue"/>, and less than or equal to <see
               cref="ulong.MaxValue"/>.</para>
            /// <para>64-разрядное целое число без знака, которое больше или равно <see
               cref="ulong.MinValue"/> и меньше или равно <see cref="ulong.MaxValue"/>.</para>
            /// </returns>
2.0
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
           public static ulong NextUInt64(this System.Random random) =>
            → random.NextUInt64(Range.UInt64);
23
            /// <summary>
24
            /// <para>Returns a random 64-bit unsigned integer that is greater than or equal to
               minimum of specified range, and less than or equal to maximum of specified
               range.</para>
            /// <para>Возвращает случайное 64-разрядное целое число без знака, которое больше или
26
            🔾 равно минимуму указанного диапазона и меньше или равно максимуму указанного
               диапазона.</para>
            /// </summary>
            /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
               псевдослучайных чисел.</para></param>
            /// <param name="range"><para>The range of possible values.</para><para>Диапазон
               возможных значений.</para></param>
            /// <returns>
30
           /// <para>A 64-bit unsigned integer that is greater than or equal to the minimum of
               specified range, and less than or equal to the maximum of the specified range. </para>
            /// <para>64-разрядное целое число без знака, которое больше или равно минимуму
               указанного диапазона и меньше или равно максимуму указанного диапазона. </para>
            /// </returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
34
           public static ulong NextUInt64(this System.Random random, Range<ulong> range) =>
35
               (ulong)(random.NextDouble() * range.Difference()) + range.Minimum;
36
            /// <summary>
37
            /// <para>Return a random <see cref="bool"/> value.</para>
            /// <para>Возвращает случайное значение <see cref="bool"/>.</para>
            /// </summary>
40
            /// <param name="random"><para>A pseudo-random number generator.</para><para>Генератор
41
               псевдослучайных чисел.</para></param>
            /// <returns><para>A random <see cref="bool"/> value.</para><para>Случайное значение
42
               <see cref="bool"/>.</para></returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
           public static bool NextBoolean(this System.Random random) => random.Next(2) == 1;
44
       }
45
1.2
     ./csharp/Platform.Random/RandomHelpers.cs
   namespace Platform.Random
1
2
       /// <summary>
3
       /// <para>Contains field-helper for <see cref="System.Random"/> class.</para>
4
       /// <para>Содержит вспомогательное поле для класса <see cref="System.Random"/>.</para>
       /// </summary>
       public static class RandomHelpers
```

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

→ System.Random(System.DateTime.UtcNow.Ticks.GetHashCode());

        }
14
15
1.3
     ./csharp/Platform.Random.Tests/RandomExtensionsTests.cs
   using Xunit;
   namespace Platform.Random.Tests
3
4
        public class RandomExtensionsTests
5
6
            [Fact]
            public void NextUInt64Test()
9
                var lastValue = OUL;
10
                var theSameCount = 0;
11
                for (var i = 0; i < 10; i++)</pre>
12
                     var newValue = RandomHelpers.Default.NextUInt64();
14
                     if (newValue == lastValue)
15
                         theSameCount++;
17
                     }
18
                     else
19
20
                         lastValue = newValue;
21
                         theSameCount = 0;
23
                     Assert.InRange(RandomHelpers.Default.NextUInt64((OUL, 5UL)), OUL, 5UL);
24
25
                Assert.True(theSameCount < 8);
26
            }
27
            [Fact]
29
            public void NextBooleanTest()
30
31
                var trueCount = 0;
32
                var falseCount = 0;
                for (var i = 0; i < 10; i++)
34
35
                     var newValue = RandomHelpers.Default.NextBoolean();
36
                     if (newValue)
37
                     {
38
                         trueCount++;
39
                     }
40
                     else
                     {
42
                         falseCount++;
                     }
44
45
                Assert.True(trueCount > 0);
46
                Assert.True(falseCount > 0);
47
            }
48
49
   }
    ./csharp/Platform.Random.Tests/RandomHelpersTests.cs
   using Xunit;
   namespace Platform.Random.Tests
3
4
        public class RandomHelpersTests
6
            [Fact]
            public void DefaultFieldTest()
9
                Assert.NotNull(RandomHelpers.Default);
10
            }
        }
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