```
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
     ./csharp/Platform.Random.Tests/RandomExtensionsTests.cs
1.3
   using Xunit;
   namespace Platform.Random.Tests
3
4
        /// <summary>
        /// <para>
6
        /// Represents the random extensions tests.
        /// </para>
        /// <para></para>
9
        /// </summary>
10
        public class RandomExtensionsTests
11
12
            /// <summary>
13
            /// <para>
            /// Tests that next u int 64 test.
            /// </para>
/// <para></para>
16
17
            /// </summary>
18
            [Fact]
19
            public void NextUInt64Test()
20
                 var lastValue = OUL;
22
                 var theSameCount = 0;
23
                for (var i = 0; i < 10; i++)</pre>
24
25
                     var newValue = RandomHelpers.Default.NextUInt64();
                     if (newValue == lastValue)
27
28
                         theSameCount++;
29
                     }
30
                     else
                     {
32
                         lastValue = newValue;
33
                         theSameCount = 0;
34
                     Assert.InRange(RandomHelpers.Default.NextUInt64((OUL, 5UL)), OUL, 5UL);
37
                 Assert.True(theSameCount < 8);
38
            }
39
40
            /// <summary>
41
            /// <para>
42
            /// Tests that next boolean test.
43
            /// </para>
44
            /// <para></para>
45
            /// </summary>
46
            [Fact]
47
            public void NextBooleanTest()
49
                 var trueCount = 0;
50
                var falseCount = 0;
51
                 for (var i = 0; i < 10; i++)</pre>
                     var newValue = RandomHelpers.Default.NextBoolean();
54
                     if (newValue)
55
                     {
56
                         trueCount++;
57
                     }
58
                     else
59
                     {
60
                         falseCount++;
61
                     }
62
63
                 Assert.True(trueCount > 0);
                 Assert.True(falseCount > 0);
65
            }
```

```
}
68
1.4 \quad ./csharp/Platform.Random.Tests/RandomHelpersTests.cs
   using Xunit;
   namespace Platform.Random.Tests
        /// <summary>
/// <para>
/// Represents the random helpers tests.
5
        /// </para>
        /// <para></para>
        public class RandomHelpersTests
{
        /// </summary>
11
12
             /// <summary>
13
             /// <para>
14
             /// Tests that default field test.
15
             /// </para>
             /// <para></para>
/// </summary>
17
18
             [Fact]
19
             public void DefaultFieldTest()
{
20
21
                  Assert.NotNull(RandomHelpers.Default);
             }
        }
^{24}
<sub>25</sub> }
```

Index

- ./csharp/Platform.Random.Tests/RandomExtensionsTests.cs, 2 ./csharp/Platform.Random.Tests/RandomHelpersTests.cs, 3 ./csharp/Platform.Random/RandomExtensions.cs, 1 ./csharp/Platform.Random/RandomHelpers.cs, 1