```
LinksPlatform's Platform Timestamps Class Library
    ./csharp/Platform.Timestamps/Timestamp.cs
   using System;
1
   using System.Runtime.CompilerServices;
2
   namespace Platform. Timestamps
4
5
       /// <summary>
6
       /// <para>Represents a timestamp.</para>
7
       /// <para>Представляет метку времени.</para>
       /// </summary>
       /// <remarks>
10
       /// <para>To make this timestamp truly unique, it is recommended to use <see
11
           cref="UniqueTimestampFactory"/>.</para>
       /// <para>Чтобы эта метка времени была действительно уникальна рекомендуется использовать
12
           <see cref="UniqueTimestampFactory"/>.</para>
       /// </remarks>
       public struct Timestamp : IEquatable<Timestamp>
14
15
            /// <summary>
16
           /// <para>Returns a string containing the default DateTime format for Timestamp.</para>
17
            /// <para>Возвращает строку, содержащую формат даты и времени по умолчанию для метки
18

→ времени.</para>
/// </summary>
           public static readonly string DefaultFormat = "yyyy.MM.dd hh:mm:ss.ffffffff";
20
21
            /// <summary>
22
           /// <para>Gets or sets the number of ticks that represent the date and time in
23
               UTC.</para>
            /// <para>Возвращает или устанавливает количество тиков, которые представляют дату и
               время в UTC.</para>
            /// </summary>
25
           public readonly ulong Ticks;
27
            /// <summary>
28
            /// <para>Creates a timestamp.</para>
29
           /// <para>Создаёт метку времени.</para>
30
           /// </summary>
31
            /// <param name="ticks"><para>A number representing the number of
            🕁 ticks.</para><para>Число представляющие количество тиков.</para></param>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
           public Timestamp(ulong ticks) => Ticks = ticks;
34
            /// <summary>
36
           /// <para>Defines an implicit conversion of a DateTime to a Timestamp.</para>
37
            /// <para>Определяет неявное преобразование DateTime в метку времени.</para>
            /// </summary>
39
            /// <param name="dateTime"><para>The DateTime struct.</para><para>Структура
40
               DateTime.</para></param>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
41
           public static implicit operator Timestamp(DateTime dateTime) => new
            → Timestamp((ulong)dateTime.ToUniversalTime().Ticks);
43
            /// <summary>
44
            /// <para>Defines an implicit conversion of a Timestamp to a DateTime.</para>
            /// <para>Определяет неявное преобразование метки времени в DateTime.</para>
46
            /// </summary>
47
            /// <param name="timestamp"><para>The Timestamp.</para><para>Отметка
48
            \hookrightarrow времени.</para></param>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
           public static implicit operator DateTime(Timestamp timestamp) => new
50
            → DateTime((long)timestamp.Ticks, DateTimeKind.Utc);
            /// <summary>
52
            /// <para>Defines an implicit conversion of a 64-bit unsigned integer to a
53
               Timestamp.</para>
            /// <para>Определяет неявное преобразование 64-разрядного целого числа без знака в метку
54
           /// <param name="ticks"><para>The number of ticks represented as a 64-bit
56
               integer.</para><para>Количество тиков представленное в виде 64-разрядного целого
               числа.</para></param>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
           public static implicit operator Timestamp(ulong ticks) => new Timestamp(ticks);
59
           /// <summary>
60
```

```
/// <para>Defines an implicit conversion of a Timestamp to a 64-bit unsigned
                integer.</para>
            /// <para>Определяет неявное преобразование метки времени в 64-разрядное целое число без
62
                знака.</para>
            /// </summary>
            /// <param name="timestamp"><para>The Timestamp </para><para>Отметка
64
               времени.</para></param>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
6.5
            public static implicit operator ulong(Timestamp timestamp) => timestamp.Ticks;
67
            /// <summarv>
68
            /// <para>Returns a string that represents the current Timestamp.</para>
69
            /// <para>Возвращает строку, которая представляет текущую метку времени.</para>
70
            /// </summary>
71
            /// <returns><para>A string that represents the current Timestamp.</para><para>CTpoka,
            → представляющая текущую метку времени.</para></returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
73
            public override string ToString() => ((DateTime)this).ToString(DefaultFormat);
74
75
            /// <summary>
76
            /// <para>Определяет, равна ли текущая отметка времени другой отметке времени.</para>
77
            /// <para>Indicates whether the current Timestamp is equal to another Timestamp.</para>
            /// </summary>
79
            /// <param name="other"><para>Other Timestamp.</para><para>Другая отметка
80
            _{\hookrightarrow} времени.</para></param> /// <returns><para>True if the current Timestamp is equal to the other Timestamp;
            _{
ightharpoonup} otherwise, false.</para>Чстину, если текущая отметка времени равна другой
               отметке времени; иначе ложь.</para></returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public bool Equals(Timestamp other) => Ticks == other.Ticks;
83
            /// <summary>
85
            /// <para>Determines whether the specified object is equal to the current object.</para>
86
            /// <para>Определяет, равен ли указанный объект текущему объекту.</para>
            /// </summary>
            /// <param name="obj"><para>The object to compare with the current
89
                object.</para><para>Объект для сравнения с текущим объектом.</para></param>
            /// <returns><para>True if the specified object is equal to the current object;
90
               otherwise, false.</para><para>Истину, если указанный объект равен текущему объекту;
               иначе ложь. </para></returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public override bool Equals(object obj) => obj is Timestamp timestamp ?
            93
            /// <summary>
            /// <para>Serves as the default hash function.</para>
95
            /// <para>Служит в качестве хэш-функции по умолчанию.</para>
96
            /// </summary>
            /// <returns><para>A hash code for the current object.</para><para>Хеш-код для текущего
98
               объекта.</para></returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
99
            public override int GetHashCode() => Ticks.GetHashCode();
100
101
            /// <summary>
102
            /// <para>Determines if the specified timestamp is equal to the current timestamp.</para>
            /// <para>Определяет, равна ли указанная метка времени текущей метке времени.</para>
104
            /// </summary>
105
            /// <param name="left"><para>The current timestamp.</para><para>Текущая метка
106
               времени.</para></param>
            /// <param name="right"><para>A timestamp to compare with this
                timestamp.</para><para>Метка времени для сравнения с этой меткой
                времени.</para></param>
            /// <returns><para>True if the current timestamp is equal to the other timestamp;
108
                otherwise, false.</para><para>True, если текущий метка времени равна другой метке
                времени; иначе false.</para></returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
109
            public static bool operator ==(Timestamp left, Timestamp right) => left.Equals(right);
111
            /// <summary>
112
            /// <para>Determines if the specified timestamp is not equal to the current
                timestamp.</para>
            /// <para>Определяет, не равна ли указанная метка времени текущей метке времени.</para>
114
            /// </summary>
115
            /// <param name="left"><para>The current timestamp.</para><para>Текущая метка
116
               времени.</para></param>
```

```
/// <param name="right"><para>A timestamp to compare with this
117
                timestamp.</para><para>Метка времени для сравнения с этой меткой
                времени.</para></param>
            /// <returns><para>True if the current timestamp is not equal to the other timestamp;
118
                otherwise, false.</para><para>True, если текущий метка времени не равна другой метке
                времени; иначе false.</para></returns>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
119
            public static bool operator !=(Timestamp left, Timestamp right) => !(left == right);
120
    }
122
     ./csharp/Platform.Timestamps/UniqueTimestampFactory.cs
1.2
    using System;
    using System.Runtime.CompilerServices;
    using Platform.Interfaces;
 3
    namespace Platform. Timestamps
 5
 6
        /// <summary>
 7
        /// <para>Represents a factory for creating unique timestamps.</para>
 8
        /// <para>Представляет фабрику по созданию уникальных отметок времени.</para>
        /// </summary>
10
        public class UniqueTimestampFactory : IFactory<Timestamp>
11
12
            /// <summary>
13
            /// <para>
14
            /// The last ticks.
            /// </para>
            /// <para></para>
17
            /// </summary>
18
            private ulong _lastTicks;
19
            /// <summary>
21
            /// <para>Creates a timestamp corresponding to the current UTC date and time or next
22
                unique timestamp.</para>
            /// <para>Создаёт отметку времени соответствующую текущей дате и времени по UTC или
23
                следующую уникальную отметку времени.</para>
            /// </summary>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
25
            public Timestamp Create()
26
                var utcTicks = (ulong)DateTime.UtcNow.Ticks;
                 _lastTicks = utcTicks > _lastTicks ? utcTicks : _lastTicks + 1;
29
                return new Timestamp(_lastTicks);
            }
31
        }
32
    }
     ./csharp/Platform.Timestamps.Tests/UniqueTimestampFactoryTests.cs
   using Xunit;
 2
    namespace Platform. Timestamps. Tests
 3
 4
        /// <summary>
        /// <para>
        /// Represents the unique timestamp factory tests.
        /// </para>
 8
        /// <para></para>
 9
        /// </summary>
10
        public class UniqueTimestampFactoryTests
11
12
            /// <summary>
13
            /// <para>
14
            /// Tests that unique timestamp test.
15
            /// </para>
16
            /// <para></para>
17
            /// </summary>
18
            [Fact]
            public void UniqueTimestampTest()
20
21
                var factory = new UniqueTimestampFactory();
22
                var timestamp1 = factory.Create();
23
                var timestamp2 = factory.Create();
24
                Assert.NotEqual(timestamp1, timestamp2);
            }
        }
27
    }
28
```

Index

- ./csharp/Platform.Timestamps.Tests/UniqueTimestampFactoryTests.cs, 3 ./csharp/Platform.Timestamps/Timestamp.cs, 1 ./csharp/Platform.Timestamps/UniqueTimestampFactory.cs, 3