

## LinksPlatform's Platform.Incrementers Class Library

### 1.1 ./csharp/Platform.Incrementers/IIncrementer.cs

```
1 using System.Runtime.CompilerServices;
2
3 namespace Platform.Incrementers
4 {
5     /// <summary>
6     /// <para>Defines an incrementer that increments any object.</para>
7     /// <para>Определяет инкрементер, который выполняет приращение любого объекта.</para>
8     /// </summary>
9     public interface IIncrementer
10     {
11         /// <summary>
12         /// <para>Increments the object by a specific value.</para>
13         /// <para>Выполняет приращение объекта на определённое значение.</para>
14         /// </summary>
15         [MethodImpl(MethodImplOptions.AggressiveInlining)]
16         void Increment();
17     }
18 }
```

### 1.2 ./csharp/Platform.Incrementers/IIncrementer[TNumber].cs

```
1 using System.Runtime.CompilerServices;
2
3 namespace Platform.Incrementers
4 {
5     /// <summary>
6     /// <para>Defines an incrementer that increments any number with a specific type.</para>
7     /// <para>Определяет инкрементер, который выполняет приращение любого числа определённого
8     ↪ типа.</para>
9     /// </summary>
10    /// <typeparam name="TNumber"><para>Type of incremented number.</para><para>Тип
11    ↪ увеличиваемого числа.</para></typeparam>
12    public interface IIncrementer<TNumber>
13    {
14        /// <summary>
15        /// <para>Increments the number by a specific value.</para>
16        /// <para>Увеличивает число на определённое значение.</para>
17        /// </summary>
18        /// <param name="number"><para>The number to be incremented.</para><para>Увеличиваемое
19        ↪ число.</para></param>
20        /// <returns><para>The incremented number.</para><para>Увеличенное
21        ↪ число.</para></returns>
22        [MethodImpl(MethodImplOptions.AggressiveInlining)]
23        TNumber Increment(TNumber number);
24    }
25 }
```

### 1.3 ./csharp/Platform.Incrementers/Incrementer.cs

```
1 using System.Runtime.CompilerServices;
2
3 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
4
5 namespace Platform.Incrementers
6 {
7     /// <remarks>
8     /// Must be class, not struct (in order to persist access to Result field value).
9     /// </remarks>
10    public class Incrementer : IIncrementer
11    {
12        protected ulong _result;
13
14        public ulong Result
15        {
16            [MethodImpl(MethodImplOptions.AggressiveInlining)]
17            get => _result;
18        }
19
20        [MethodImpl(MethodImplOptions.AggressiveInlining)]
21        public Incrementer(ulong initialValue) => _result = initialValue;
22
23        [MethodImpl(MethodImplOptions.AggressiveInlining)]
24        public Incrementer() { }
25
26        [MethodImpl(MethodImplOptions.AggressiveInlining)]
27        public void Increment() => _result++;
28    }
29 }
```

#### 1.4 ./csharp/Platform.Incrementers/Incrementer[TValue, TDecision].cs

```
1 using System.Runtime.CompilerServices;
2
3 #pragma warning disable IDE0060 // Remove unused parameter
4 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
5
6 namespace Platform.Incrementers
7 {
8     public class Incrementer<TValue, TDecision> : Incrementer
9     {
10         private readonly TDecision _trueValue;
11
12         [MethodImpl(MethodImplOptions.AggressiveInlining)]
13         public Incrementer(ulong initialValue, TDecision trueValue) : base(initialValue) =>
14             ↪ _trueValue = trueValue;
15
16         [MethodImpl(MethodImplOptions.AggressiveInlining)]
17         public Incrementer(TDecision trueValue) => _trueValue = trueValue;
18
19         [MethodImpl(MethodImplOptions.AggressiveInlining)]
20         public Incrementer() { }
21
22         [MethodImpl(MethodImplOptions.AggressiveInlining)]
23         public TDecision IncrementAndReturnTrue()
24         {
25             _result++;
26             return _trueValue;
27         }
28
29         [MethodImpl(MethodImplOptions.AggressiveInlining)]
30         public TDecision IncrementAndReturnTrue(TValue value)
31         {
32             _result++;
33             return _trueValue;
34         }
35     }
36 }
```

#### 1.5 ./csharp/Platform.Incrementers/Incrementer[TValue].cs

```
1 using System.Runtime.CompilerServices;
2
3 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
4
5 namespace Platform.Incrementers
6 {
7     public class Incrementer<TValue> : Incrementer<TValue, bool>
8     {
9         [MethodImpl(MethodImplOptions.AggressiveInlining)]
10         public Incrementer(ulong initialValue) : base(initialValue, true) { }
11
12         [MethodImpl(MethodImplOptions.AggressiveInlining)]
13         public Incrementer() : base(true) { }
14     }
15 }
```

#### 1.6 ./csharp/Platform.Incrementers.Tests/IncrementerTests.cs

```
1 using Xunit;
2
3 namespace Platform.Incrementers.Tests
4 {
5     public class IncrementerTests
6     {
7         [Fact]
8         public void ParameterlessConstructedSetterTest()
9         {
10             Incrementer<int> incrementer = new Incrementer<int>();
11             Assert.Equal(default, incrementer.Result);
12         }
13
14         [Fact]
15         public void ConstructedWithDefaultValueSetterTest()
16         {
17             Incrementer<int> incrementer = new Incrementer<int>(9UL);
18             Assert.Equal(9UL, incrementer.Result);
19         }
20
21         [Fact]
22         public void MethodsWithBooleanReturnTypeTest()
23         {
24             Incrementer<int> incrementer = new Incrementer<int>();
25         }
26     }
27 }
```

```
25         incrementer.Increment();
26         Assert.Equal(1UL, incrementer.Result);
27         Assert.True(incrementer.IncrementAndReturnTrue());
28         Assert.Equal(2UL, incrementer.Result);
29     }
30
31     [Fact]
32     public void MethodsWithIntegerReturnTypeTest()
33     {
34         Incrementer<int, int> incrementer = new Incrementer<int, int>(1);
35         incrementer.Increment();
36         Assert.Equal(1UL, incrementer.Result);
37         Assert.Equal(1, incrementer.IncrementAndReturnTrue());
38         Assert.Equal(2UL, incrementer.Result);
39     }
40 }
41 }
```

## Index

[./csharp/Platform.Incrementers.Tests/IncrementerTests.cs](#), 2  
[./csharp/Platform.Incrementers/IIncrementer.cs](#), 1  
[./csharp/Platform.Incrementers/IIncrementer\[TNumber\].cs](#), 1  
[./csharp/Platform.Incrementers/Incrementer.cs](#), 1  
[./csharp/Platform.Incrementers/Incrementer\[TValue, TDecision\].cs](#), 1  
[./csharp/Platform.Incrementers/Incrementer\[TValue\].cs](#), 2