```
LinksPlatform's Platform Setters Class Library
    ./csharp/Platform.Setters/SetterBase.cs
   using System.Runtime.CompilerServices;
   using Platform.Interfaces;
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
4
   namespace Platform.Setters
6
       /// <summary>
       /// < \verb|para>Represents| a base implementation for an setter that allows you to set a passed
9
           value as the result value.</para>
       /// <para>Представляет базовую реализацию для установщика, который позволяет установить
10
       \rightarrow переданное ему значение в качестве результирующего значения.
11
       /// <typeparam name="TResult"><para>The type of result value.</para><para>Тип
12
           результирующего значения.</para></typeparam>
       /// <remarks>
13
       /// Must be class, not struct (in order to persist access to Result property value).
14
       /// </remarks>
15
       public abstract class SetterBase<TResult> : ISetter<TResult>
16
17
            /// <summary>
18
           /// <para>Represents the result value.</para>
19
           /// <para>Представляет результирующие значение.</para>
20
            /// </summary>
           protected TResult _result;
22
23
           /// <summary>
24
           /// <para>Gets result value.</para>
25
            /// <para>Возвращает результирующее значение.</para>
           /// </summary>
27
           public TResult Result => _result;
29
            /// <summary>
            /// <para>Initializes a new instance of the SetterBase class.</para>
31
            /// <para>Инициализирует новый экземпляр класса SetterBase.</para>
32
            /// </summary>
33
34
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
           protected SetterBase() { }
3.5
36
            /// <summary>
           /// <para>Initializes a new instance of the SetterBase class using the passed-in value
38

→ as the default result value.
            /// <para>Инициализирует новый экземпляр класса SetterBase, используя переданное
39
               значение в качестве результирующего по умолчанию. </para>
            /// </summary>
40
            /// <param name="defaultValue"><para>The default result
               value.</para><para>Pезультирующее значение по умолчанию.</para></param>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
42
           protected SetterBase(TResult defaultValue) => _result = defaultValue;
43
44
            /// <summary>
45
            /// <para>Sets the passed value as the result.</para>
46
            /// <para>Устанавливает переданное значение в качестве результирующего.</para>
47
            /// </summary>
48
            /// <param name="value"><para>The result value.</para><para>Результирующее
49
               значение.</para></param>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
           public void Set(TResult value) => _result = value;
51
       }
52
   }
53
    ./csharp/Platform.Setters/Setter[TResult, TDecision].cs
   using System.Collections.Generic;
   using System.Runtime.CompilerServices;
2
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
   namespace Platform.Setters
       /// <summary>
       /// <para>Represents implementation for an setter that allows you to set a passed value as
           the result value. This setter implementation has additional methods that, simultaneously
           with setting the result value, return <typeparamref name="TDecision"/> values indicating
           true or false.</para>
```

```
/// <para>Представляет реализацию для установщика, который позволяет установить переданное
10
            ему значение в качестве результирующего значения. В этой реализации установщика есть
            дополнительные методы, которые одновременно с установкой результирующего значения
           возвращают значения типа <typeparamref name="TDecision"/>, обозначающие истину или
           ложь.</para>
        /// </summary>
        /// <typeparam name="TResult"><para>The type of result value.</para><para>Тип
12
           результирующего значения.</para></typeparam>
        /// <typeparam name="TDecision"><para>The type of value which will be used to make the
13
           decision.</para><pаra>Тип значения на основе которого будет приниматься
           решение.</para></typeparam>
        public class Setter<TResult, TDecision> : SetterBase<TResult>
15
            private readonly TDecision _trueValue;
private readonly TDecision _falseValue;
17
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
19
            public Setter(TDecision trueValue, TDecision falseValue, TResult defaultValue)
20
                : base(defaultValue)
22
                 _trueValue = trueValue;
                 _falseValue = falseValue;
24
            }
25
26
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
27
            public Setter(TDecision trueValue, TDecision falseValue) : this(trueValue, falseValue,
               default) { }
29
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
30
            public Setter(TResult defaultValue) : base(defaultValue) { }
32
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
33
            public Setter() { }
34
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
36
            public TDecision SetAndReturnTrue(TResult value)
37
38
                 result = value:
39
                return _trueValue;
41
42
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
43
            public TDecision SetAndReturnFalse(TResult value)
44
45
                 _result = <mark>value</mark>
46
                return _falseValue;
47
            }
48
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
50
            public TDecision SetFirstAndReturnTrue(IList<TResult> list)
51
52
                 result = list[0];
                return _trueValue;
54
            }
56
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
57
            public TDecision SetFirstAndReturnFalse(IList<TResult> list)
59
                _result = list[0];
60
61
                return _falseValue;
62
        }
63
64
     ./csharp/Platform.Setters/Setter[TResult].cs
1.3
   using System.Runtime.CompilerServices;
2
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
4
   namespace Platform.Setters
5
6
        public class Setter<TResult> : Setter<TResult, bool>
9
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public Setter(TResult defaultValue) : base(true, false, defaultValue) { }
10
11
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
12
            public Setter() : base(true, false) { }
14
   }
15
```

```
./csharp/Platform.Setters.Tests/SetterTests.cs
   using Xunit;
2
   namespace Platform.Setters.Tests
4
        public class SetterTests
5
6
            [Fact]
            public void ParameterlessConstructedSetterTest()
                Setter<int> setter = new Setter<int>();
10
                Assert.Equal(default, setter.Result);
11
            }
12
13
            [Fact]
            public void ConstructedWithDefaultValueSetterTest()
15
16
                Setter<int> setter = new Setter<int>(9);
17
                Assert.Equal(9, setter.Result);
19
            [Fact]
21
            public void MethodsWithBooleanReturnTypeTest()
22
23
                Setter<int> setter = new Setter<int>();
24
                Assert.True(setter.SetAndReturnTrue(1));
25
                Assert.Equal(1, setter.Result);
26
                Assert.False(setter.SetAndReturnFalse(2));
                Assert.Equal(2, setter.Result);
28
                Assert.True(setter.SetFirstAndReturnTrue(new int[] { 3 }));
29
                Assert.Equal(3, setter.Result);
30
                Assert.False(setter.SetFirstAndReturnFalse(new int[] { 4 }));
31
                Assert.Equal(4, setter.Result);
32
            }
34
            [Fact]
35
            public void MethodsWithIntegerReturnTypeTest()
37
                Setter<int, int> setter = new Setter<int, int>(1, 0);
38
                Assert.Equal(1, setter.SetAndReturnTrue(1));
39
40
                Assert.Equal(1, setter.Result);
                Assert.Equal(0, setter.SetAndReturnFalse(2));
41
                Assert.Equal(2, setter.Result);
Assert.Equal(1, setter.SetFirstAndReturnTrue(new int[] { 3 }));
42
43
                Assert.Equal(3, setter.Result);
44
                Assert.Equal(0, setter.SetFirstAndReturnFalse(new int[] { 4 }));
45
                Assert.Equal(4, setter.Result);
46
            }
47
        }
48
49
```

## Index

- ./csharp/Platform.Setters.Tests/SetterTests.cs, 3 ./csharp/Platform.Setters/SetterBase.cs, 1 ./csharp/Platform.Setters/Setter[TResult, TDecision].cs, 1 ./csharp/Platform.Setters/Setter[TResult].cs, 2