```
LinksPlatform's Platform Setters Class Library
    ./csharp/Platform.Setters/SetterBase.cs
   using System.Runtime.CompilerServices;
   using Platform.Interfaces;
2
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
4
   namespace Platform.Setters
6
        /// <summary>
       /// <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;
28
29
            /// <summary>
30
            /// <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>Результирующее 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/SetterExtensions.cs
   using System.Collections.Generic;
   namespace Platform. Setters
       public static class SetterExtensions
5
           public static TDecision SetFirstFromNonNullListAndReturnTrue<TResult, TDecision>(this
               Setter<TResult, TDecision> setter, IList<TResult>? list)
                if (list != null)
                {
                    setter.Set(list[0]);
1.1
12
                return setter.TrueValue;
```

```
14
15
             public static TDecision SetFirstFromNonNullFirstListAndReturnTrue<TResult,
                 TDecision>(this Setter<TResult, TDecision> setter, IList<TResult>? list1,
                 IList<TResult>? list2)
17
                 if (list1 != null)
                 {
19
                      setter.Set(list1[0]);
20
                 return setter.TrueValue;
22
             }
23
24
             public static TDecision SetSecondFromNonNullFirstListAndReturnTrue<TResult,</pre>
                 TDecision>(this Setter<TResult, TDecision> setter, IList<TResult>? list1,
IList<TResult>? list2)
26
                 if (list1 != null)
27
                 {
2.8
                      setter.Set(list1[1]);
29
                 return setter.TrueValue;
31
             }
32
33
             public static TDecision SetThirdFromNonNullFirstListAndReturnTrue<TResult,</pre>
34
                 TDecision>(this Setter<TResult, TDecision> setter, IList<TResult>? list1, IList<TResult>? list2)
35
                 if (list1 != null)
36
                 {
37
                      setter.Set(list1[2]);
39
                 return setter.TrueValue;
40
             }
41
42
             public static TDecision SetFirstFromNonNullSecondListAndReturnTrue<TResult,</pre>
43
                 TDecision>(this Setter<TResult, TDecision> setter, IList<TResult>? list1, IList<TResult>? list2)
44
                 if (list2 != null)
45
                 {
46
                      setter.Set(list2[0]);
47
48
49
                 return setter.TrueValue;
             }
50
51
             public static TDecision SetSecondFromNonNullSecondListAndReturnTrue<TResult,
52
                 TDecision>(this Setter<TResult, TDecision> setter, IList<TResult>? list1,
IList<TResult>? list2)
53
                 if (list2 != null)
54
                 {
55
                      setter.Set(list2[1]);
57
58
                 return setter.TrueValue;
             }
59
             public static TDecision SetThirdFromNonNullSecondListAndReturnTrue<TResult,</pre>
61
                 TDecision>(this Setter<TResult, TDecision> setter, IList<TResult>? list1, IList<TResult>? list2)
62
                 if (list2 != null)
63
                 {
                      setter.Set(list2[2]);
65
66
                 return setter.TrueValue;
67
             }
68
        }
69
70
     ./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
4
   namespace Platform. Setters
6
7
        /// <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
1.0
            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>
1.1
        /// <typeparam name="TResult"><para>The type of result value.</para><para>Тип
12
        \rightarrow результирующего значения.</para></typeparam> /// <typeparam name="TDecision"><para>The type of value which will be used to make the
13
           decision.</para><para>Тип значения на основе которого будет приниматься
            решение.</para></typeparam>
        public class Setter<TResult, TDecision> : SetterBase<TResult>
            public readonly TDecision TrueValue;
public readonly TDecision FalseValue;
16
17
18
            /// <summary>
19
            \label{eq:local_para} \///\ cpara>Initializes a new instance of the Setter class using the passed-in value as
20

→ the default result value.
            /// <para>Инициализирует новый экземпляр класса Setter, используя переданные значения
21
                trueValue, falseValue, defaultValue в качестве результирующего по умолчанию.</para>
            /// </summary>
            /// <param name="defaultValue"><para>The default result
                value.</para><para>Результирующее значение по умолчанию.</para></param>
            [\texttt{MethodImpl}(\bar{\texttt{MethodImpl}}0ptions.AggressiveInlining)]
24
            public Setter(TDecision trueValue, TDecision falseValue, TResult defaultValue)
25
                 : base(defaultValue)
            {
27
                 TrueValue = trueValue;
                FalseValue = falseValue;
29
            }
30
31
            /// <summary>
32
            /// <para>Gets result value.</para>
33
            /// <para>Возвращает результирующее значение.</para>
34
            /// </summary>
35
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
36
            public Setter(TDecision trueValue, TDecision falseValue) : this(trueValue, falseValue,
             \rightarrow default) \{\ \}
38
            /// <summary>
39
            /// <para>Gets result value.</para>
40
            /// <para>Возвращает результирующее значение.</para>
41
            /// </summary>
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
43
            public Setter(TResult defaultValue) : base(defaultValue) { }
44
            /// <summary>
46
            /// <para>Gets result value.</para>
47
            /// <para>Возвращает результирующее значение.</para>
48
            /// </summary>
49
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
50
            public Setter() { }
51
52
            /// <summary>
53
            /// <para>Gets result value.</para>
            /// <para>Возвращает результирующее значение.</para>
55
            /// </summary>
56
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
57
            public TDecision SetAndReturnTrue(TResult value)
59
                 _result = value;
                 return TrueValue;
61
            }
62
63
            /// <summary>
64
            /// <para>Gets result value.</para>
            /// <para>Возвращает результирующее значение.</para>
66
            /// </summary>
67
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
68
            public TDecision SetAndReturnFalse(TResult value)
69
70
                 _result = value;
                 return FalseValue;
72
```

```
7.3
74
             /// <summary>
75
             /// <para>Gets result value.</para>
             /// <para>Возвращает результирующее значение.</para>
77
             /// </summarv>
78
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
79
            public TDecision SetFirstAndReturnTrue(IList<TResult>? list)
80
81
                 if (list != null)
82
83
                     _result = list[0];
85
86
                 return TrueValue;
             }
87
             /// <summary>
89
             /// <para>Gets result value.</para>
90
             /// <para>Возвращает результирующее значение.</para>
91
             /// </summary>
92
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
93
            public TDecision SetFirstAndReturnFalse(IList<TResult>? list)
94
                 if (list != null)
96
                 {
97
                     _result = list[0];
98
99
                 return FalseValue;
100
             }
        }
102
103
     ./csharp/Platform.Setters/Setter[TResult].cs
1.4
   using System.Runtime.CompilerServices;
    #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
 3
    namespace Platform.Setters
 5
 6
        /// <summary>
        /// <para>
 8
        /// Represents the setter.
        /// </para>
        /// <para></para>
11
        /// </summary>
12
        /// <seealso cref="Setter{TResult, bool}"/>
13
        public class Setter<TResult> : Setter<TResult, bool>
14
15
             /// <summary>
             /// <para>
17
             /// Initializes a new <see cref="Setter"/> instance.
18
19
             /// </para>
             /// <para></para>
20
             /// </summary>
21
             /// <param name="defaultValue">
22
             /// <para>A default value.</para>
             /// <para></para>
/// </param>
24
25
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
26
            public Setter(TResult defaultValue) : base(true, false, defaultValue) { }
27
28
             /// <summary>
             /// <para>
30
             /// Initializes a new <see cref="Setter"/> instance.
31
             /// </para>
32
             /// <para></para>
33
             /// </summary>
34
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
35
            public Setter() : base(true, false) { }
        }
37
    }
38
    ./csharp/Platform.Setters.Tests/SetterTests.cs
   using Xunit;
    namespace Platform.Setters.Tests
 3
        public class SetterTests
```

```
[Fact]
            public void ParameterlessConstructedSetterTest()
                Setter<int> setter = new Setter<int>();
                Assert.Equal(default, setter.Result);
11
12
13
            [Fact]
14
            public void ConstructedWithDefaultValueSetterTest()
15
16
                Setter<int> setter = new Setter<int>(9);
                Assert.Equal(9, setter.Result);
18
            }
19
20
            [Fact]
21
            public void MethodsWithBooleanReturnTypeTest()
23
                Setter<int> setter = new Setter<int>();
24
                Assert.True(setter.SetAndReturnTrue(1));
                Assert.Equal(1, setter.Result);
26
                Assert.False(setter.SetAndReturnFalse(2));
                Assert.Equal(2, setter.Result);
28
                Assert.True(setter.SetFirstAndReturnTrue(new int[] { 3 }));
                Assert.Equal(3, setter.Result);
30
                Assert.False(setter.SetFirstAndReturnFalse(new int[] { 4 }));
31
32
                Assert.Equal(4, setter.Result);
            }
33
            [Fact]
            public void MethodsWithIntegerReturnTypeTest()
36
38
                Setter<int, int> setter = new Setter<int, int>(1, 0);
                Assert.Equal(1, setter.SetAndReturnTrue(1));
                Assert.Equal(1, setter.Result);
40
                Assert.Equal(0, setter.SetAndReturnFalse(2));
                Assert.Equal(2, setter.Result);
                Assert.Equal(1, setter.SetFirstAndReturnTrue(new int[] { 3 }));
43
                Assert.Equal(3, setter.Result);
Assert.Equal(0, setter.SetFirstAndReturnFalse(new int[] { 4 }));
44
                Assert.Equal(4, setter.Result);
46
            }
47
        }
48
   }
49
```

27

37

39

41

Index

- ./csharp/Platform.Setters.Tests/SetterTests.cs, 4
 ./csharp/Platform.Setters/SetterBase.cs, 1
 ./csharp/Platform.Setters/SetterExtensions.cs, 1
 ./csharp/Platform.Setters/Setter[TResult, TDecision].cs, 2
 ./csharp/Platform.Setters/Setter[TResult].cs, 4