```
LinksPlatform's Platform Interfaces Class Library
    ./ICounter[TResult, TArgument].cs
   namespace Platform. Interfaces
       /// <summary>
3
       /// <para>Defines a counter that requires passing an argument to perform a count.</para>
4
       /// <para>Определяет счетчик, который требует передачи аргумента для выполнения
          подсчёта.</para>
        /// </summary>
       /// <typeparam name="TArgument"><para>The argument type.</para><para>Тип
           аргумента.</para></typeparam>
       /// <typeparam name="TResult"><para>The count result type.</para><para>Тип результата
           подсчета.</para></typeparam>
       public interface ICounter<out TResult, in TArgument>
10
            /// <summary>
            /// <para>Performs a count that requires passing an argument.</para>
            /// <para>Выполняет посчёт, для которого требуется передача аргумент.</para>
            /// </summary>
14
            /// <param name="argument"><para>The argument.</para><para>Аргумент.</para></param>
15
            /// <returns><para>The count result.</para><para>Результат подсчёта.</para></returns>
           TResult Count(TArgument argument);
17
18
   ./ICounter[TResult].cs
   namespace Platform.Interfaces
       /// <summary>
3
       /// <para>Defines a counter.</para>
4
       /// <para>Определяет счетчик.</para>
       /// </summary>
       /// <typeparam name="TResult"><para>The count result type.</para><para>Тип результата
           подсчета.</para></typeparam>
       public interface ICounter<out TResult>
            /// <summary>
10
            /// <para>Performs a count.</para>
11
            /// <para>Выполняет посчёт.</para>
           /// </summary>
13
            /// <returns><para>The count result.</para><para>Peзультат подсчёта.</para></returns>
14
           TResult Count();
       }
16
17
1.3
    ./ICriterionMatcher.cs
   namespace Platform. Interfaces
2
       /// <summary>
3
       /// <para>Defines a criterion matcher, that contains a specific method of determining
           whether the argument matches criterion or not.</para>
       /// <para>Определяет объект который проверяет соответствие критерию и содержащий конкретный
        → метод определения, соответствует ли аргумент критерию или нет.</para>
       /// </summary>
       /// <typeparam name="TArgument"><para>Argument type.</para><para>Тип
           аргумента.</para></typeparam>
       public interface ICriterionMatcher<in TArgument>
            /// <summary>
10
           /// <para>Determines whether the argument matches the criterion.</para>
           /// <para>Определяет, соответствует ли аргумент критерию.</para>
12
           /// </summary>
            /// <param name="argument"><para>The argument.</para><para>Аргумент.</para></param>
            /// <returns><para>A value that determines whether the argument matches the
15
            criterion.</para><pаra>Значение, определяющие соответствует ли аргумент
               критерию.</para></returns>
           bool IsMatched(TArgument argument);
16
       }
17
   }
18
1.4
    ./IFactory.cs
   namespace Platform.Interfaces
2
       /// <summary>
3
       /// <para>Defines a factory that produces instances of a specific type.</para>
       /// <para>Определяет фабрику, которая производит экземпляры определенного типа.</para>
```

```
/// </summary>
       /// <typeparam name="TProduct"><para>Туре of produced instances.</para><para>Тип
       → производимых экземпляров.</para></typeparam>
public interface IFactory<out TProduct>
            /// <summary>
10
           /// <para>Creates an instance of TProduct type.</para>
11
           /// <para>Создает экземпляр типа TProduct.</para>
12
           /// </summary>
13
            /// <returns><para>The instance of TProduct type.</para><para>Экземпляр типа
14
            TProduct Create();
15
       }
16
   }
    /IProperties cs
  namespace Platform. Interfaces
1
2
       /// <summary>
3
       /// <para>Defines a properties operator that is able to get or set values of properties of a
4
           object of a specific type.</para>
       /// <para>Определяет оператор свойств, который может получать или установливать значения
           свойств объекта определенного типа.</para>
       /// </summary>
       /// <typeparam name="TObject"><para>Object type.</para>Tип объекта.</para></typeparam>
       /// <typeparam name="TProperty"><para>Property reference type.</para><para>Тип ссылки на
           свойство.</para></typeparam>
       /// <typeparam name="TValue"><para>Property value type.</para><para>Тип значения
           свойства.</para></typeparam>
       public interface IProperties<in TObject, in TProperty, TValue>
10
11
12
           /// <para>Gets the value of the property in the specified object.</para>
13
           /// <para>Получает значение свойства в указанном объекте.</para>
14
           /// </summary>
            /// <param name="object"><para>The object reference.</para><para>Ссылка на

→ объект.</para></param>

            /// <param name="property"><para>The property reference.</para><para>Ссылка на
17
               свойство.</para></param>
            /// <returns><para>The value of the property.</para><para>Значение
18
               свойства.</para></returns>
           TValue GetValue(TObject @object, TProperty property);
19
20
           /// <summary>
2.1
            /// <para>Sets the value of a property in the specified object.</para>
            /// <para>Устанавливает значение свойства в указанном объекте.</para>
23
            /// </summary>
24
           /// <param name="object"><para>The object reference.</para><para>Ссылка на
25
               объект.</para></param>
            /// <param name="property"><para>The property reference.</para><para>Ссылка на
               свойство.</para></param>
            /// <param name="value"><para>The value.</para><para>Значение</para></param>
           void SetValue(TObject @object, TProperty property, TValue value);
28
       }
29
   }
     ./IProperty.cs
1.6
   namespace Platform. Interfaces
1
2
       /// <summary>
3
       /// <para>Defines a specific property that is able to get or set values of that
4
           property.</para>
       /// <para>Определяет определённого свойства, который может получать или установливать его
           значения.</para>
       /// </summary>
       /// <typeparam name="TObject"><para>Object type.</para><para>Тип объекта.</para></typeparam>
       /// <typeparam name="TValue"><para>Property value type.</para><para>Тип значения
           свойства.</para></typeparam>
       public interface IProperty in TObject, TValue : ISetter TValue, TObject : IProvider TValue,
9
           TObject>
        \hookrightarrow
10
11
   }
12
```

```
../IProvider[TProvided, TArgument].cs
   namespace Platform.Interfaces
1
2
        /// <summary>
3
       /// <para>Defines the provider of objects for which an argument must be specified.</para>
4
       /// <para>Определяет поставщика объектов, для получения которых необходимо указать
           аргумент.</para>
        /// </summary>
       /// <typeparam name="TProvided"><para>Type of provided objects.</para><para>Тип
       → предоставляемых объектов.</para></typeparam>
/// <typeparam name="TArgument"><para>Argument type.</para><para>Тип
           аргумента.</para></typeparam>
       public interface IProvider out TProvided, in TArgument>
10
            /// <summary>
            /// <para>Provides an object.</para>
12
            /// <para>Предоставляет объект. </para>
13
            /// </summary>
14
            /// <param name="argument"><para>The argument required to acquire the
                object.</para><para>Аргумент, необходимый для получения объекта.</para></param>
            /// <returns><para>The object.</para><para>Объект.</para></returns>
16
            TProvided Get(TArgument argument);
17
       }
18
   }
19
1.8
    ./IProvider[TProvided].cs
   namespace Platform. Interfaces
1
2
        /// <summary>
3
       /// <para>Defines the provider of objects.</para>
4
       /// <para>Определяет поставщика объектов.</para>
        /// <\bar{\formary>
       /// <typeparam name="TProvided"><para>Type of provided object.</para><para>Тип
           предоставляемого объекта.</para></typeparam>
       public interface IProvider out TProvided>
9
            /// <summary>
10
            /// <para>Provides an object.</para>
11
            /// </summary>
            /// <returns><para>The provided object.</para></returns>
13
            TProvided Get();
14
       }
15
16
1.9
     ./ISetter[TValue, TArgument].cs
   namespace Platform.Interfaces
1
2
        /// <summary>
3
       /// <para>Defines an setter that requires an argument to set the passed value as a new
4
           state.</para>
       /// <para>Определяет установщик, которому для установки переданного значения в качестве
       \rightarrow нового состояния требуется аргумент.
       /// <typeparam name="TValue"><para>Type of setted value.</para><para>Тип устанавливаемого
            значения.</para></typeparam>
       /// <typeparam name="TArgument"><para>The argument type.</para>Тип
            аргумента.</para></typeparam>
       public interface ISetter<in TValue, in TArgument>
1.0
            /// <summary>
11
            /// <para>Sets the value of a specific property in the specified object.</para>
12
            /// <para>Устанавливает значение определённого свойства в указанном объекте.</para>
13
            /// </summary>
14
            /// <param name="argument"><para>The argument.</para><para>Aргумент.</para></param>
            /// <param name="value"><para>The value.</para><para>Значение.</para></param>
            void Set(TArgument argument, TValue value);
17
       }
18
      ./ISetter[TValue].cs
1.10
   namespace Platform.Interfaces
1
2
       /// <summary>
       /// <para>Defines an setter that sets the passed value as a new state.</para>
4
       /// <para>Определяет установщик, который устанавливает переданное значение в качестве нового
           состояния.</para>
       /// </summary>
```

Index ./ICounter[TResult, TArgument].cs, 1 ./ICounter[TResult].cs, 1 ./ICriterionMatcher.cs, 1 ./IFactory.cs, 1 ./IProperties.cs, 2 ./IProperty.cs, 2 ./IProvider[TProvided, TArgument].cs, 2 ./IProvider[TProvided].cs, 3 ./ISetter[TValue, TArgument].cs, 3 ./ISetter[TValue].cs, 3