```
LinksPlatform's Platform.Communication.Protocol.Lino Class Library
     ./Platform.Communication.Protocol.Lino/IListExtensions.cs
   using Platform.Collections;
   using System;
2
   using System.Collections.Generic;
   using System.Linq;
4
   using System.Runtime.CompilerServices;
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
   namespace Platform.Communication.Protocol.Lino
9
   {
10
        public static class IListExtensions
11
12
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
13
            public static string Format(this IList<Link> links) => string.Join(Environment.NewLine,
14

→ links.Select(l => l.ToString()));
15
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
16
            public static string Format(this IList<Link> links, bool lessParentheses)
17
18
                if (lessParentheses == false)
19
                {
20
                    return links.Format();
21
                }
22
                else
24
                     return string.Join(Environment.NewLine, links.Select(1 =>
25
                        1.ToString().TrimSingle('(').TrimSingle(')')));
                }
            }
27
        }
28
   }
     ./Platform.Communication.Protocol.Lino/Link.cs
   using System;
   using System.Collections.Generic;
2
   using System.Runtime.CompilerServices;
   using System. Text;
   using Platform.Collections;
   using Platform.Collections.Lists;
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
9
10
   namespace Platform.Communication.Protocol.Lino
11
        public struct Link : IEquatable<Link>
13
            public string Id
14
15
                 [MethodImpl(MethodImplOptions.AggressiveInlining)]
16
                [MethodImpl(MethodImplOptions.AggressiveInlining)]
18
19
                set;
            }
20
            public IList<Link> Values
22
23
                 [MethodImpl(MethodImplOptions.AggressiveInlining)]
24
25
                [MethodImpl(MethodImplOptions.AggressiveInlining)]
27
                set;
            }
28
29
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
30
            public Link(string id, IList<Link> values) => (Id, Values) = (id, values);
32
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
33
            public Link(IList<Link> values) : this(null, values) { }
34
3.5
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public Link(string id) : this(id, null) { }
37
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
39
            public override string ToString() => Values.IsNullOrEmpty() ? $\$"({Id})" :
40

→ GetLinkValuesString();

            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            private string GetLinkValuesString() => Id == null ? $\(\frac{1}{3}\) (\(\frac{1}{3}\) GetValuesString()\(\frac{1}{3}\))":
43
                |$|"({Id}: {GetValuesString()})";
```

```
44
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public string GetValuesString()
46
47
                var sb = new StringBuilder();
                for (int i = 0; i < Values.Count; i++)</pre>
49
50
                    if (i > 0)
51
                         sb.Append(' ');
53
54
                    sb.Append(GetValueString(Values[i]));
55
                return sb.ToString();
57
            }
58
59
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
60
            public static string GetValueString(Link value) => value.ToLinkOrIdString();
62
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
63
            public string ToLinkOrIdString() => Values.IsNullOrEmpty() ? Id : ToString();
64
6.5
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public static implicit operator Link(string value) => new Link(value);
67
68
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
69
            public static implicit operator Link(ValueTuple<string, IList<Link>> value) => new
70

→ Link(value.Item1, value.Item2);
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
72
            public override bool Equals(object obj) => obj is Link link ? Equals(link) : false;
73
74
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
75
            public override int GetHashCode() => (Id, Values.GenerateHashCode()).GetHashCode();
76
77
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
78
            public bool Equals(Link other) => Id == other.Id && Values.EqualTo(other.Values);
79
80
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public static bool operator ==(Link left, Link right) => left.Equals(right);
82
83
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
84
            public static bool operator !=(Link left, Link right) => !(left == right);
85
86
   }
    ./Platform.Communication.Protocol.Lino.Tests/ParserTests.cs
   using Xunit;
2
3
   namespace Platform.Communication.Protocol.Lino.Tests
4
        public static class ParserTests
            [Fact]
            public static void ParseAndStringifyTest()
                var source = @"(papa (lovesMama: loves mama))
10
    (son lovesMama)
11
    (daughter lovesMama)
12
    (all (love mama))";
                var parser = new Parser();
14
                var links = parser.Parse(source);
15
                var target = links.Format();
16
                Assert. Equal(source, target);
17
            }
18
19
            [Fact]
20
            public static void ParseAndStringifyWithLessParenthesesTest()
21
22
                var source = @"lovesMama: loves mama
23
   papa lovesMama
^{24}
   son lovesMama
   daughter lovesMama
26
   all (love mama)";
27
                var parser = new Parser();
28
                var links = parser.Parse(source);
29
                var target = links.Format(lessParentheses: true);
30
                Assert.Equal(source, target);
31
            }
32
```

```
33
              [Fact]
^{34}
             public static void SignificantWhitespaceTest()
35
36
                  var source = @"
37
38
   users
      user1
39
        id
43
41
         name
42
         first
43
             John
44
           last
Williams
45
46
         location
New York
47
48
      age
23
user2
49
50
51
         id
52
           56
54
        name
          first
55
           Igor
middle
56
57
             Petrovich
           last
59
             Ivanov
60
         location
61
          Moscow
62
         age
20";
63
64
65
67
              }
68
         }
69
   }
70
```

## Index

- ./Platform.Communication.Protocol.Lino.Tests/ParserTests.cs, 2 ./Platform.Communication.Protocol.Lino/IListExtensions.cs, 1 ./Platform.Communication.Protocol.Lino/Link.cs, 1