LinksPlatform's Platform.Comparers Class Library

1.1 ./csharp/Platform.Comparers/Comparer.cs

```csharp
using System;
using System.Collections;

#pragma warning disable CS1591 // Missing XML comment for publicly visible type or member

namespace Platform.Comparers
{
    public class Comparer : IComparer
    {
        private readonly Func<object, object, int> _compare;
        public Comparer(IComparer comparer) : this(comparer.Compare) { }
        public Comparer(Func<object, object, int> compare) => _compare = compare;
        public int Compare(object x, object y) => _compare(x, y);
        public static implicit operator Comparer(Func<object, object, int> compare) => new
          ↪  Comparer(compare);
        public static implicit operator Func<object, object, int>(Comparer comparer) =>
          ↪  comparer._compare;
    }
}
```

1.2 ./csharp/Platform.Comparers/Comparer[T].cs

```csharp
using System;
using System.Collections.Generic;

#pragma warning disable CS1591 // Missing XML comment for publicly visible type or member

namespace Platform.Comparers
{
    public class Comparer<T> : IComparer<T>
    {
        private readonly Func<T, T, int> _compare;
        public Comparer(IComparer<T> comparer) : this(comparer.Compare) { }
        public Comparer(Func<T, T, int> compare) => _compare = compare;
        public int Compare(T x, T y) => _compare(x, y);
        public static implicit operator Comparer<T>(Func<T, T, int> compare) => new
          ↪  Comparer<T>(compare);
        public static implicit operator Func<T, T, int>(Comparer<T> comparer) =>
          ↪  comparer._compare;
    }
}
```

1.3 ./csharp/Platform.Comparers.Tests/ComparerTests.cs

```csharp
using System;
using Xunit;

namespace Platform.Comparers.Tests
{
    public class ComparerTests
    {
        [Fact]
        public void FunctionConversionTest()
        {
            Func<object, object, int> function = (object x, object y) =>
              ↪  x.GetHashCode().CompareTo(y.GetHashCode());
            Comparer comparer = function;
            var x = new object();
            var y = new object();
            Assert.Equal(function(x, y), comparer.Compare(x, y));
        }
    }
}
```

Index