

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text.RegularExpressions;
5
6 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
7
8 namespace Platform.RegularExpressions.Transformer.CSharpToCpp
9 {
10     public class CSharpToCppTransformer : Transformer
11     {
12         public static readonly IList<ISubstitutionRule> FirstStage = new List<SubstitutionRule>()
13         {
14             // #pragma warning disable CS1591 // Missing XML comment for publicly visible type
15             // or member
16             //
17             (new Regex(@"^~\s*?~#pragma[~sa-zA-Z0-9~\./]+~$"), "", null, 0),
18             // [MethodImpl(MethodImplOptions.AggressiveInlining)]
19             //
20             (new Regex(@"~$~\s+~[MethodImpl~(MethodImplOptions~.AggressiveInlining~)\~]~"), "",
21              null, 0),
22             // [Fact]
23             //
24             (new Regex(@"~$~\s+~[Fact~]~"), "", null, 0),
25             // {
26             (new Regex(@"~{~\s+~[\r~\n~]+~"), "{" + Environment.NewLine, null, 0),
27             // Platform.Collections.Methods.Lists
28             // Platform::Collections::Methods::Lists
29             (new Regex(@"(namespace[~\r~\n~]+?)~.([~\r~\n~]+?)~"), "$1::~$2", null, 20),
30             // public abstract class
31             // class
32             (new Regex(@"(public abstract|static) class"), "class", null, 0),
33             // class GenericCollectionMethodsBase {
34             // class GenericCollectionMethodsBase { public:
35             (new Regex(@"class ([a-zA-Z0-9~]+)~(\s+){~}"", "class $1$2{" + Environment.NewLine + "
36             ~ public:", null, 0),
37             // class GenericCollectionMethodsBase<TElement> {
38             // template <typename TElement> class GenericCollectionMethodsBase { public:
39             (new Regex(@"class ([a-zA-Z0-9~]+)~<([a-zA-Z0-9~]+)~>([~{~}~+){~}"", "template <typename $2>
40             ~ class $1$3{" + Environment.NewLine + "      public:", null, 0),
41             // static void
42             ~ TestMultipleCreationsAndDeletions<TElement>(SizedBinaryTreeMethodsBase<TElement>
43             ~ tree, TElement* root)
44             // template<typename T> static void
45             ~ TestMultipleCreationsAndDeletions<TElement>(SizedBinaryTreeMethodsBase<TElement>
46             ~ tree, TElement* root)
47             (new Regex(@"static ([a-zA-Z0-9~]+) ([a-zA-Z0-9~]+)~<([a-zA-Z0-9~]+)~>~\((([~\r~\n~]+)~)\~)"",
48              ~ "template <typename $3> static $1 $2($4)", null, 0),
49             // (this
50             // (
51             (new Regex(@"~\((this ~)"", "(" , null, 0),
52             // Func<TElement> treeCount
53             // TElement(*treeCount)()
54             (new Regex(@"Func<([a-zA-Z0-9~]+)~> ([a-zA-Z0-9~]+)"", "$1(*$2)()", null, 0),
55             // Action<TElement> free
56             // void (*free)(TElement)
57             (new Regex(@"Action<([a-zA-Z0-9~]+)~> ([a-zA-Z0-9~]+)"", "void (*$2)($1)", null, 0),
58             // private const int MaxPath = 92;
59             // static const int MaxPath = 92;
60             (new Regex(@"private const ([a-zA-Z0-9~]+) ([_a-zA-Z0-9~]+) = ([a-zA-Z0-9~]+);",
61              ~ "static const $1 $2 = $3;", null, 0),
62             // protected virtual
63             // virtual
64             (new Regex(@"protected virtual)"", "virtual", null, 0),
65             // protected abstract TElement GetFirst();
66             // virtual TElement GetFirst() = 0;
67             (new Regex(@"protected abstract ([~;~}~}~+);)"", "virtual $1 = 0;", null, 0),
68             // public virtual
69             // virtual
70             (new Regex(@"public virtual)"", "virtual", null, 0),
71             // protected readonly
72             //
73             (new Regex(@"protected readonly ~)"", "", null, 0),
74             // protected readonly TreeElement[] _elements;

```

```

66 // TreeElement _elements[N];
67 (new Regex(@"(protected|private) readonly ([a-zA-Z<>0-9]+)([\\[\\]]+)
    ↳ ([_a-zA-Z0-9]+);"), "$2 $4[N];", null, 0),
68 // protected readonly TElement Zero;
69 // TElement Zero;
70 (new Regex(@"(protected|private) readonly ([a-zA-Z<>0-9]+) ([_a-zA-Z0-9]+);"), "$2
    ↳ $3;", null, 0),
71 // private
72 //
73 (new Regex(@"(\W)(private|protected|public|internal) "), "$1", null, 0),
74 // SizeBalancedTree(int capacity) => a = b;
75 // SizeBalancedTree(int capacity) { a = b; }
76 (new Regex(@"(^s+)(override )?(void )?([a-zA-Z0-9]+)\(((^\\([+])\\)s+=>s+([~;]+);"),
    ↳ "$1$2$3$4($5) { $6; }", null, 0),
77 // () => Integer<TElement>.Zero,
78 // () { return Integer<TElement>.Zero; },
79 (new Regex(@"\\(\\)s+=>s+([~\\r\\n;]+?);"), "() { return $1; }", null, 0),
80 // => Integer<TElement>.Zero;
81 // { return Integer<TElement>.Zero; }
82 (new Regex(@"\\)s+=>s+([~\\r\\n;]+?);"), ") { return $1; }", null, 0),
83 // () { return avlTree.Count; }
84 // []()-> auto { return avlTree.Count; }
85 (new Regex(@", \\(\\) { return ([~;]+); }"), ", []()-> auto { return $1; }", null, 0),
86 // Count => GetSizeOrZero(Root);
87 // GetCount() { return GetSizeOrZero(Root); }
88 (new Regex(@"([A-Z][a-z]+)s+=>s+([~;]+);"), "Get$1() { return $2; }", null, 0),
89 // var
90 // auto
91 (new Regex(@"(\W)var(\W)"), "$1auto$2", null, 0),
92 // unchecked
93 //
94 (new Regex(@"[\\r\\n]{2}s*?unchecked\s*?$"), "", null, 0),
95 // $"
96 // "
97 (new Regex(@"\$"""), "\"", null, 0),
98 // Console.WriteLine(...)
99 // printf("...\n")
100 (new Regex(@"Console\.WriteLine\\(\"([~\""]+)"\""), "printf(\"$1\\n\")", null, 0),
101 // throw new InvalidOperationException
102 // throw std::exception
103 (new Regex(@"throw new (InvalidOperationException|Exception)"), "throw
    ↳ std::exception", null, 0),
104 // override void PrintNode(TElement node, StringBuilder sb, int level)
105 // void PrintNode(TElement node, StringBuilder sb, int level) override
106 (new Regex(@"override ([a-zA-Z0-9 \\*+]+)(\\([~\\)]+?\\)")), "$1$2 override", null, 0),
107 // string
108 // char*
109 (new Regex(@"(\W)string(\W)"), "$1char*$2", null, 0),
110 // sbyte
111 // std::int8_t
112 (new Regex(@"(\W)sbyte(\W)"), "$1std::int8_t$2", null, 0),
113 // uint
114 // std::uint32_t
115 (new Regex(@"(\W)uint(\W)"), "$1std::uint32_t$2", null, 0),
116 // char*[] args
117 // char* args[]
118 (new Regex(@"([_a-zA-Z0-9:~*+]?)\\[\\] ([a-zA-Z0-9]+)"), "$1 $2[]", null, 0),
119 // using Platform.Numbers;
120 //
121 (new Regex(@"([\\r\\n]{2}|^~)s*?using [\\.a-zA-Z0-9]+;s*?$"), "", null, 0),
122 // struct TreeElement { }
123 // struct TreeElement { };
124 (new Regex(@"(struct|class) ([a-zA-Z0-9]+)(\\s+){([\\sa-zA-Z0-9;:_]+?)}([~;])"), "$1
    ↳ $2$3{$4};$5", null, 0),
125 // class Program { }
126 // class Program { };
127 (new Regex(@"(struct|class) ([a-zA-Z0-9]+)[~\\r\\n]*([\\r\\n]+(?<indentLevel>[\\t
    ↳ ]*)?)\\{([\\S\\s]+?[\\r\\n]+\\k<indentLevel>)\\}([~;]|$)"), "$1 $2$3{$4};$5", null, 0),
128 // class SizedBinaryTreeMethodsBase : GenericCollectionMethodsBase
129 // class SizedBinaryTreeMethodsBase : public GenericCollectionMethodsBase
130 (new Regex(@"class ([a-zA-Z0-9]+) : ([a-zA-Z0-9]+)"), "class $1 : public $2", null,
    ↳ 0),
131 }.Cast<ISubstitutionRule>().ToList();
132
133 public static readonly IList<ISubstitutionRule> LastStage = new List<SubstitutionRule>()
134 {
135     // ref sizeBalancedTree2.Root

```

```

136 // &sizeBalancedTree2.Root
137 (new Regex(@"ref ([a-zA-Z0-9]+)\.([a-zA-Z0-9\*]+)", "&$1->$2", null, 0),
138 // ref GetElement(node).Right
139 // &GetElement(node).Right
140 (new Regex(@"ref ([a-zA-Z0-9]+)\((([a-zA-Z0-9\*]+))\)\.([a-zA-Z0-9]+)",
141     ↳ "&$1($2)->$3", null, 0),
142 // GetElement(node).Right
143 // GetElement(node)->Right
144 (new Regex(@"([a-zA-Z0-9]+)\((([a-zA-Z0-9\*]+))\)\.([a-zA-Z0-9]+)", "$1($2)->$3",
145     ↳ null, 0),
146 // = ref GetLeftReference(root);
147 // = GetLeftReference(root);
148 (new Regex(@" = ref ([a-zA-Z0-9]+)\((([a-zA-Z0-9\*]+))\);", " = $1($2);", null, 0),
149 // ref this->GetElement(node)
150 // this->GetElement(node)
151 (new Regex(@"ref this->([a-zA-Z0-9]+)\((([a-zA-Z0-9\*]+))\)", "this->$1($2)", null,
152     ↳ 0),
153 // ref GetElement(node)
154 // GetElement(node)
155 (new Regex(@"ref ([a-zA-Z0-9]+)\((([a-zA-Z0-9\*]+))\)", "$1($2)", null, 0),
156 // = ref left;
157 // = left;
158 (new Regex(@" = ref ([a-zA-Z0-9]+);", " = $1;", null, 0),
159 // (ref left)
160 // (left)
161 (new Regex(@"\ (ref ([a-zA-Z0-9]+)\(|\(|,)", "($1$2", null, 0),
162 // ref TElement
163 // TElement*
164 (new Regex(@"( |\()ref ([a-zA-Z0-9]+)", "$1$2* ", null, 0),
165 // return ref _elements[node];
166 // return &_elements[node];
167 (new Regex(@"return ref ([_a-zA-Z0-9]+)\([([_a-zA-Z0-9\*]+)\];", "return &$1[$2];",
168     ↳ null, 0),
169 // default
170 // 0
171 (new Regex(@"(\W)default(\W)", "${1}0$2", null, 0),
172 // //define ENABLE_TREE_AUTO_DEBUG_AND_VALIDATION
173 //
174 (new Regex(@"\\\/[ \t]*#define[ \t]+[_a-zA-Z0-9]+[ \t]*"), "", null, 0),
175 // #if USEARRAYPOOL\r\n#endif
176 //
177 (new Regex(@"#if [a-zA-Z0-9]+\s+#endif", "", null, 0),
178 // \n ... namespace
179 // namespace
180 (new Regex(@"(\\S\\r\\n){1,2}?\\r\\n+namespace)", "$1namespace", null, 0),
181 // \n ... class
182 // class
183 (new Regex(@"(\\S\\r\\n){1,2}?\\r\\n+class)", "$1class", null, 0),
184 }.Cast<ISubstitutionRule>().ToList();
185
186 public CSharpToCppTransformer(IList<ISubstitutionRule> extraRules) :
187     ↳ base(FirstStage.Concat(extraRules).Concat(LastStage).ToList()) { }
188
189 public CSharpToCppTransformer() : base(FirstStage.Concat(LastStage).ToList()) { }
190 }

```

./Platform.RegularExpressions.Transformer.CSharpToCpp/obj/Release/netstandard2.1/Platform.RegularExpressions.

```

1 //-----
2 // <auto-generated>
3 //     Generated by the MSBuild WriteCodeFragment class.
4 // </auto-generated>
5 //-----
6
7 using System;
8 using System.Reflection;
9
10 [assembly: System.Reflection.AssemblyConfigurationAttribute("Release")]
11 [assembly: System.Reflection.AssemblyCopyrightAttribute("Konstantin Diachenko")]
12 [assembly: System.Reflection.AssemblyDescriptionAttribute("LinksPlatform\'s
13     ↳ Platform.RegularExpressions.Transformer.CSharpToCpp Class Library" +
14     "")]
15 [assembly: System.Reflection.AssemblyFileVersionAttribute("0.0.1.0")]
16 [assembly: System.Reflection.AssemblyInformationalVersionAttribute("0.0.1")]
17 [assembly: System.Reflection.AssemblyTitleAttribute("Platform.RegularExpressions.Transformer.CSh
18     ↳ arpToCpp")]
19 [assembly: System.Reflection.AssemblyVersionAttribute("0.0.1.0")]

```

./Platform.RegularExpressions.Transformer.CSharpToCpp.Tests/CSharpToCppTransformerTests.cs

```
1 using Xunit;
2
3 namespace Platform.RegularExpressions.Transformer.CSharpToCpp.Tests
4 {
5     public class CSharpToCppTransformerTests
6     {
7         [Fact]
8         public void HelloWorldTest()
9         {
10             const string helloWorldCode = @"using System;
11 class Program
12 {
13     public static void Main(string[] args)
14     {
15         Console.WriteLine(""Hello, world!"");
16     }
17 }";
18             const string expectedResult = @"class Program
19 {
20     public:
21     static void Main(char* args[])
22     {
23         printf(""Hello, world!\n"");
24     }
25 };";
26             var transformer = new CSharpToCppTransformer();
27             var actualResult = transformer.Transform(helloWorldCode, new Context(null));
28             Assert.Equal(expectedResult, actualResult);
29         }
30     }
31 }
```

./Platform.RegularExpressions.Transformer.CSharpToCpp.Tests/obj/Release/netcoreapp3.0/Platform.RegularExpressions

```
1 //-----
2 // <auto-generated>
3 //     Generated by the MSBuild WriteCodeFragment class.
4 // </auto-generated>
5 //-----
6
7 using System;
8 using System.Reflection;
9
10 [assembly: System.Reflection.AssemblyCompanyAttribute("Platform.RegularExpressions.Transformer.CSharpToCpp.Tests")]
11 [assembly: System.Reflection.AssemblyConfigurationAttribute("Release")]
12 [assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
13 [assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0")]
14 [assembly: System.Reflection.AssemblyProductAttribute("Platform.RegularExpressions.Transformer.CSharpToCpp.Tests")]
15 [assembly: System.Reflection.AssemblyTitleAttribute("Platform.RegularExpressions.Transformer.CSharpToCpp.Tests")]
16 [assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]
```

Index

- ./Platform.RegularExpressions.Transformer.CSharpToCpp.Tests/CSharpToCppTransformerTests.cs, 3
- ./Platform.RegularExpressions.Transformer.CSharpToCpp.Tests/obj/Release/netcoreapp3.0/Platform.RegularExpressions.Transform4
- ./Platform.RegularExpressions.Transformer.CSharpToCpp/CSharpToCppTransformer.cs, 1
- ./Platform.RegularExpressions.Transformer.CSharpToCpp/obj/Release/netstandard2.1/Platform.RegularExpressions.Transform3