

# LinksPlatform's Platform.RegularExpressions.Transformer.CSharpToCpp Class Library

## ./Platform.RegularExpressions.Transformer.CSharpToCpp/CSharpToCppTransformer.cs

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

```

66 //
67 (new Regex(@"protected readonly "), "", null, 0),
68 // protected readonly TreeElement[] _elements;
69 // TreeElement _elements[N];
70 (new Regex(@"(protected|private) readonly ([a-zA-Z<>0-9]+)([\\[\\]]+)
71   ↳ ([_a-zA-Z0-9]+);"), "$2 $4[N];", null, 0),
72 // protected readonly TElement Zero;
73 // TElement Zero;
74 (new Regex(@"(protected|private) readonly ([a-zA-Z<>0-9]+) ([_a-zA-Z0-9]+);"), "$2
75   ↳ $3;", null, 0),
76 // private
77 //
78 (new Regex(@"(\W)(private|protected|public|internal) "), "$1", null, 0),
79 // SizeBalancedTree(int capacity) => a = b;
80 // SizeBalancedTree(int capacity) { a = b; }
81 (new Regex(@"(^\\s+)(override )?(void )?([a-zA-Z0-9]+)\\(((\\[\\])\\)\\s+=>\\s+([~;]+);"),
82   ↳ "$1$2$3$4($5) { $6; }", null, 0),
83 // () => Integer<TElement>.Zero,
84 // () { return Integer<TElement>.Zero; },
85 (new Regex(@"\\(\\)\\s+=>\\s+([~\\r\\n;]+?);"), "()", { return $1; }, null, 0),
86 // => Integer<TElement>.Zero;
87 // { return Integer<TElement>.Zero; }
88 (new Regex(@"\\(\\)\\s+=>\\s+([~\\r\\n;]+?);"), "()" { return $1; }, null, 0),
89 // []()-> auto { return avlTree.Count; }
90 (new Regex(@"", "\\(\\) { return ([~;]+); }"), "()", []()-> auto { return $1; }, null, 0),
91 // Count => GetSizeOrZero(Root);
92 // GetCount() { return GetSizeOrZero(Root); }
93 (new Regex(@"([A-Z][a-z]+)\\s+=>\\s+([~;]+);"), "Get$1() { return $2; }", null, 0),
94 // var
95 // auto
96 (new Regex(@"(\\W)var(\\W)"), "$1auto$2", null, 0),
97 // unchecked
98 //
99 (new Regex(@"[\\r\\n]{2}\\s*?unchecked\\s*?$"), "", null, 0),
100 // $"
101 // "
102 (new Regex(@"\\$"""), "\\\"", null, 0),
103 // Console.WriteLine("...")
104 // printf("...\\n")
105 (new Regex(@"Console\\.WriteLine\\(\"([~\"']+)\")\\n\""), "printf(\"$1\\n\")", null, 0),
106 // throw new InvalidOperationException
107 // throw std::exception
108 (new Regex(@"throw new (InvalidOperationException|Exception)"), "throw
109   ↳ std::exception", null, 0),
110 // override void PrintNode(TElement node, StringBuilder sb, int level)
111 // void PrintNode(TElement node, StringBuilder sb, int level) override
112 (new Regex(@"override ([a-zA-Z0-9 \\*+]+)\\(\\([~\\(\\)]+?\\)\\)", "$1$2 override", null, 0),
113 // string
114 // char*
115 (new Regex(@"(\\W)string(\\W)"), "$1char*$2", null, 0),
116 // sbyte
117 // std::int8_t
118 (new Regex(@"(\\W)sbyte(\\W)"), "$1std::int8_t$2", null, 0),
119 // uint
120 // std::uint32_t
121 (new Regex(@"(\\W)uint(\\W)"), "$1std::uint32_t$2", null, 0),
122 // char*[] args
123 // char* args[]
124 (new Regex(@"([_a-zA-Z0-9:~*+]?)[\\(\\)] ([a-zA-Z0-9]+)", "$1 $2[]", null, 0),
125 // using Platform.Numbers;
126 //
127 (new Regex(@"([\\r\\n]{2}|^~)\\s*?using [\\.a-zA-Z0-9+;\\s*?$)", "", null, 0),
128 // struct TreeElement { }
129 // struct TreeElement { };
130 (new Regex(@"(struct|class) ([a-zA-Z0-9]+)\\(\\s+\\{([\\sa-zA-Z0-9;:_]+?}\\([~;])\\)", "$1
131   ↳ $2$3{$4};$5", null, 0),
132 // class Program { }
133 // class Program { };
134 (new Regex(@"(struct|class) ([a-zA-Z0-9]+)[~\\r\\n]*([\\r\\n]+(?<indentLevel>[\\t
135   ↳ ]*)?)\\{([\\S\\s]+?[\\r\\n]+\\k<indentLevel>)\\}([~;]|$)", "$1 $2$3{$4};$5", null, 0),
136 // class SizedBinaryTreeMethodsBase : GenericCollectionMethodsBase
137 // class SizedBinaryTreeMethodsBase : public GenericCollectionMethodsBase
138 (new Regex(@"class ([a-zA-Z0-9]+) : ([a-zA-Z0-9]+)", "class $1 : public $2", null,
139   ↳ 0),
140 }.Cast<ISubstitutionRule>().ToList();

```



```
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 }
```

## Index

./Platform.RegularExpressions.Transformer.CSharpToCpp.Tests/CSharpToCppTransformerTests.cs, 3  
./Platform.RegularExpressions.Transformer.CSharpToCpp/CSharpToCppTransformer.cs, 1