

# .NET Compiler Platform



Anton Eremin

Senior Software Developer, R&D



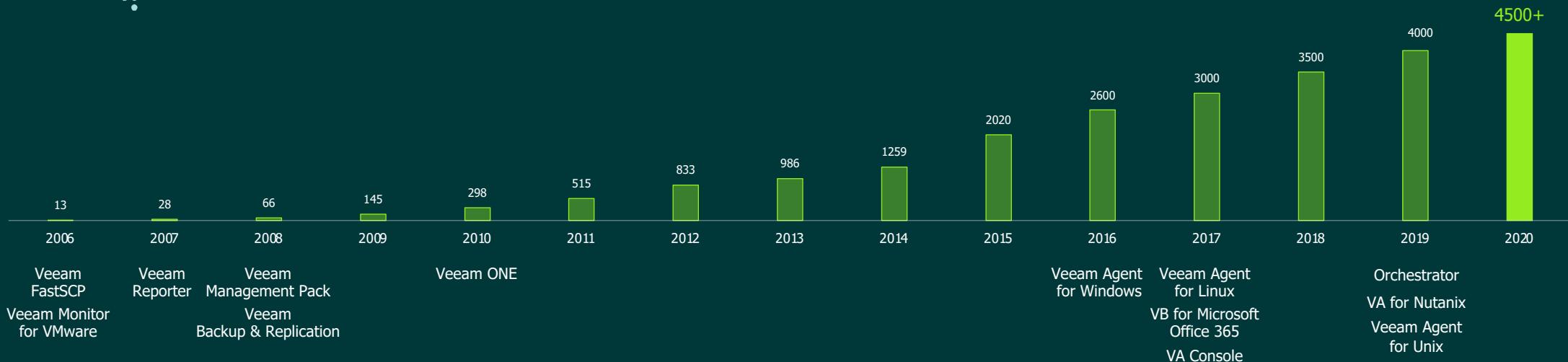
## Офисы R&D:

Прага

Санкт-Петербург



Продукты Сотрудники



400,000+  
Клиентов в мире,



35+  
Стран присутствия



В 3.5 раза  
Выше удовлетворенность  
клиентов, чем в среднем  
по индустрии

# Содержание

1



Написать более 5000 методов за месяц?



Сэкономить более 3000 ч/ч?



Заменить «копипасту»?



# .Net Compiler Platform

2

Code Analysis

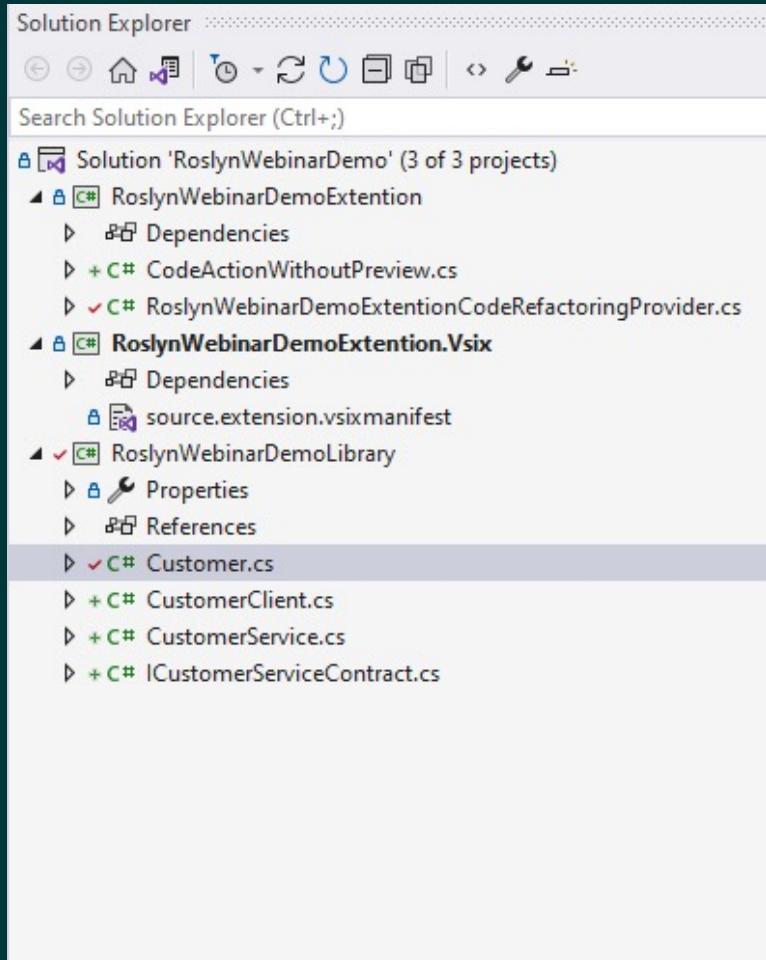
Workspaces

Features

VS Integrations

# Workspaces

3



# Workspaces

```
using (var workspace = MSBuildWorkspace.Create())
{
```

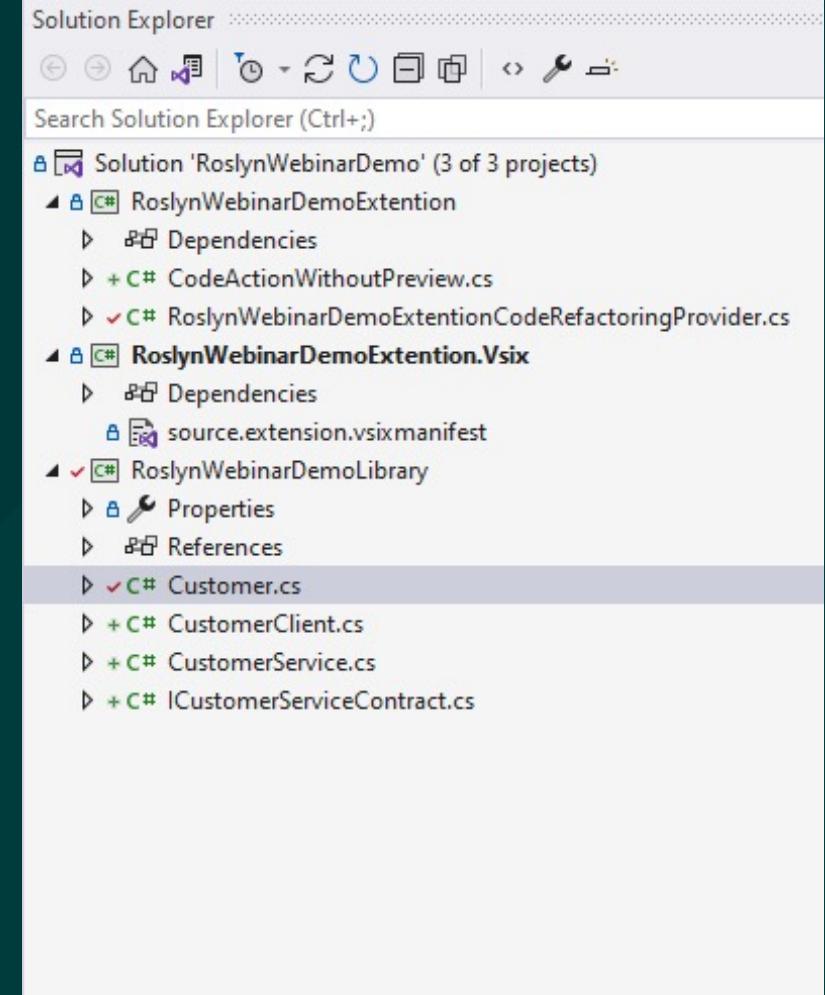
```
    var solutionPath = "C:\\Webinars\\RoslynWebinarDemo.sln";
    var solution = await workspace.OpenSolutionAsync(solutionPath);
```

```
    var compilations = await Task.WhenAll(solution.Projects.Select(x => x.GetCompilationAsync()));
```

```
    var project = solution.Projects.FirstOrDefault(p => p.Name == "RoslynWebinarDemoLibrary");
    if (project == null || !project.HasDocuments)
        return;
```

```
    var document = project.Documents.FirstOrDefault(d => d.Name == "Customer.cs");
    if (document == null)
        return;
```

```
    var updateNode = await EditDocument(document);
```



# Document

5

Document

SyntaxTree

Objects

# Code Analysis

Синтаксическая  
модель

Компиляторы

Семантическая  
модель

# Синтаксическая модель

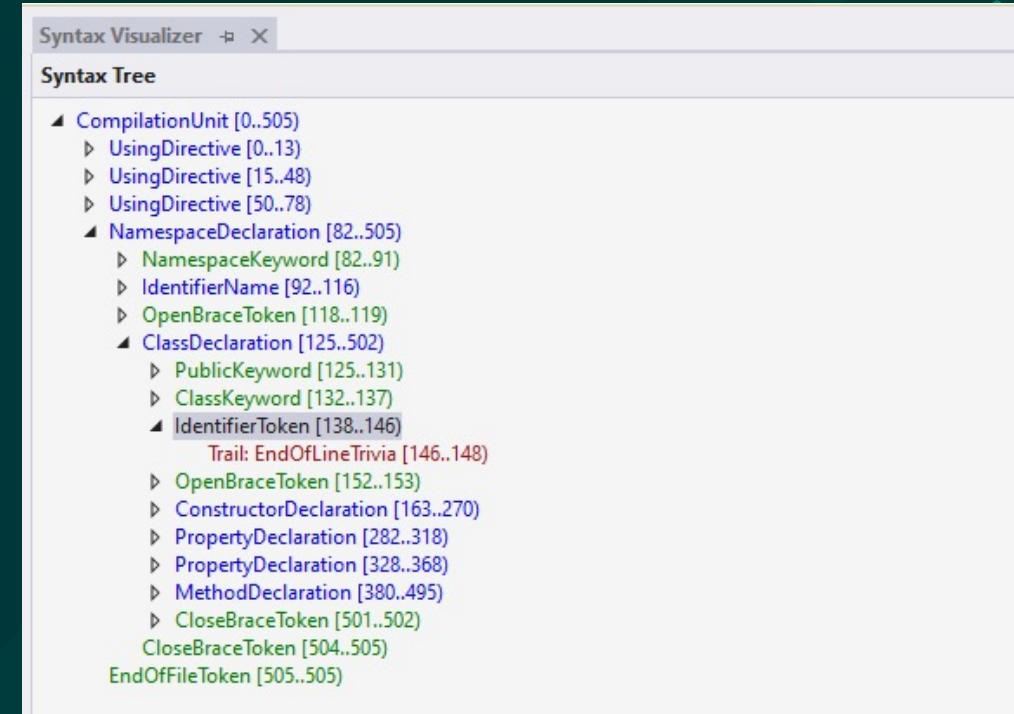
7

```
using System.Data.SqlClient;

namespace RoslynWebinarDemoLibrary
{
    public class Customer
    {
        public Customer(long id, string name)
        {
            Id = id;
            Name = name;
        }

        public long Id { get; private set; }
        public string Name { get; private set; }

        internal List<SqlParameter> GetParameters()
        {
            throw new NotImplementedException();
        }
    }
}
```



# Синтаксическая модель

```
1 reference | 0 changes | 0 authors, 0 changes
private static async Task<SyntaxNode> EditDocument(Document document)
{
    SyntaxTree syntaxTree = await document.GetSyntaxTreeAsync();
    SyntaxNode rootNode = await syntaxTree.GetRootAsync();
```



# Узлы

9

```
using System.Data.SqlClient;

namespace RoslynWebinarDemoLibrary
{
    4 references | Anton, 1 day ago | 1 author, 1 change
    public class Customer
    {
        0 references | Anton, 1 day ago | 1 author, 1 change
        public Customer(long id, string name)
        {
            Id = id;
            Name = name;
        }

        1 reference | Anton, 1 day ago | 1 author, 1 change
        public long Id { get; private set; }
        1 reference | Anton, 1 day ago | 1 author, 1 change
        public string Name { get; private set; }

        1 reference | 0 changes | 0 authors, 0 changes
        internal List<SqlParameter> GetParameters()
        {
            throw new NotImplementedException();
        }
    }
}
```

→ NamespaceDeclarationSyntax

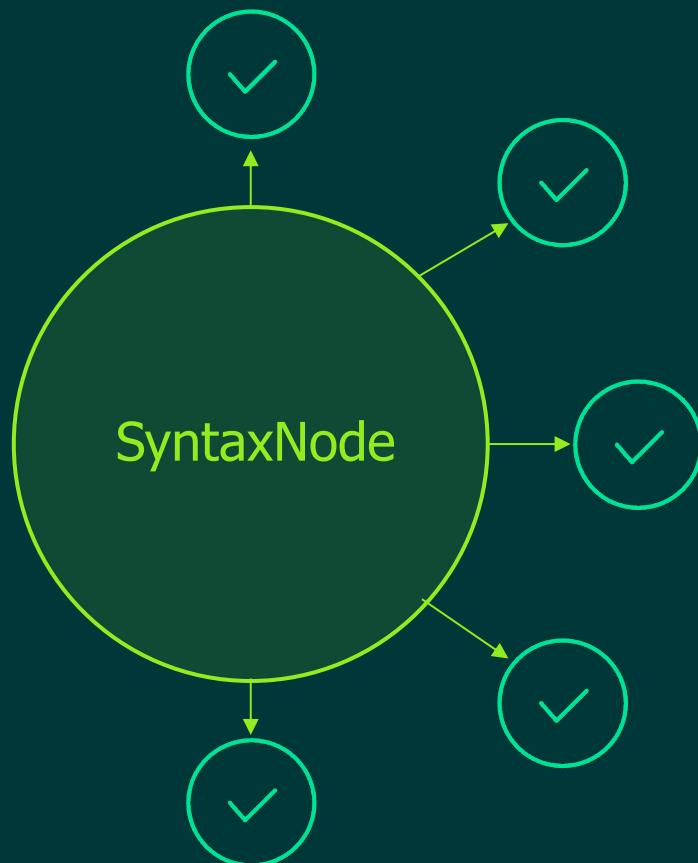
→ ClassDeclarationSyntax

→ ParameterListSyntax

→ PropertyDeclarationSyntax

→ PropertyBlockSyntax

Будут преобразованы в IL



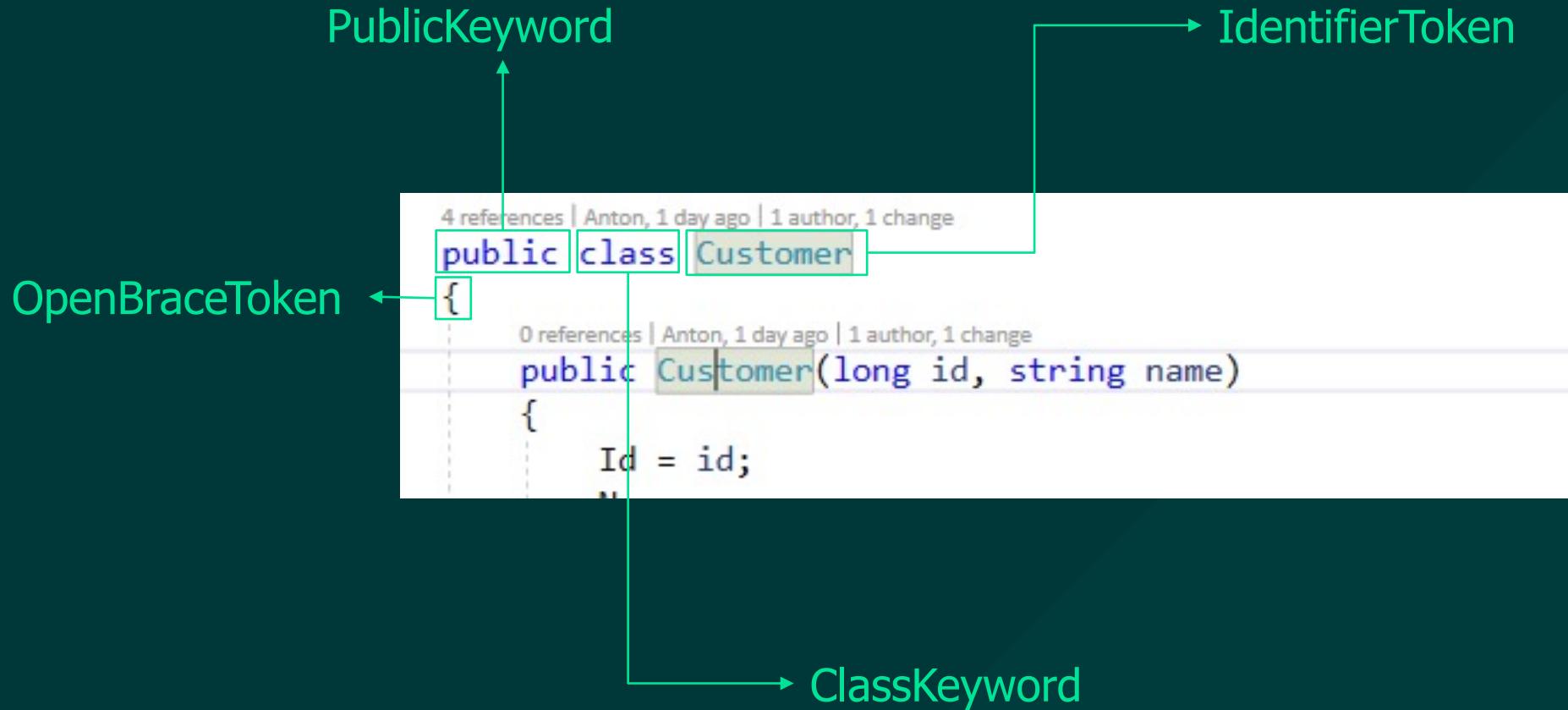
Syntax Visualizer ×

Syntax Tree

- ▲ CompilationUnit [0..505]
  - ▷ UsingDirective [0..13]
  - ▷ UsingDirective [15..48]
  - ▷ UsingDirective [50..78]
- ▲ NamespaceDeclaration [82..505]
  - ▷ NamespaceKeyword [82..91]
  - ▷ IdentifierName [92..116]
  - ▷ OpenBraceToken [118..119]
- ▲ ClassDeclaration [125..502]
  - ▷ PublicKeyword [125..131]
  - ▷ ClassKeyword [132..137]
  - ▲ IdentifierToken [138..146]
    - Trail: EndOfLineTrivia [146..148]
    - ▷ OpenBraceToken [152..153]
    - ▷ ConstructorDeclaration [163..270]
    - ▷ PropertyDeclaration [282..318]
    - ▷ PropertyDeclaration [328..368]
    - ▷ MethodDeclaration [380..495]
    - ▷ CloseBraceToken [501..502]
    - CloseBraceToken [504..505]
    - EndOfFileToken [505..505]

# Лексемы

11

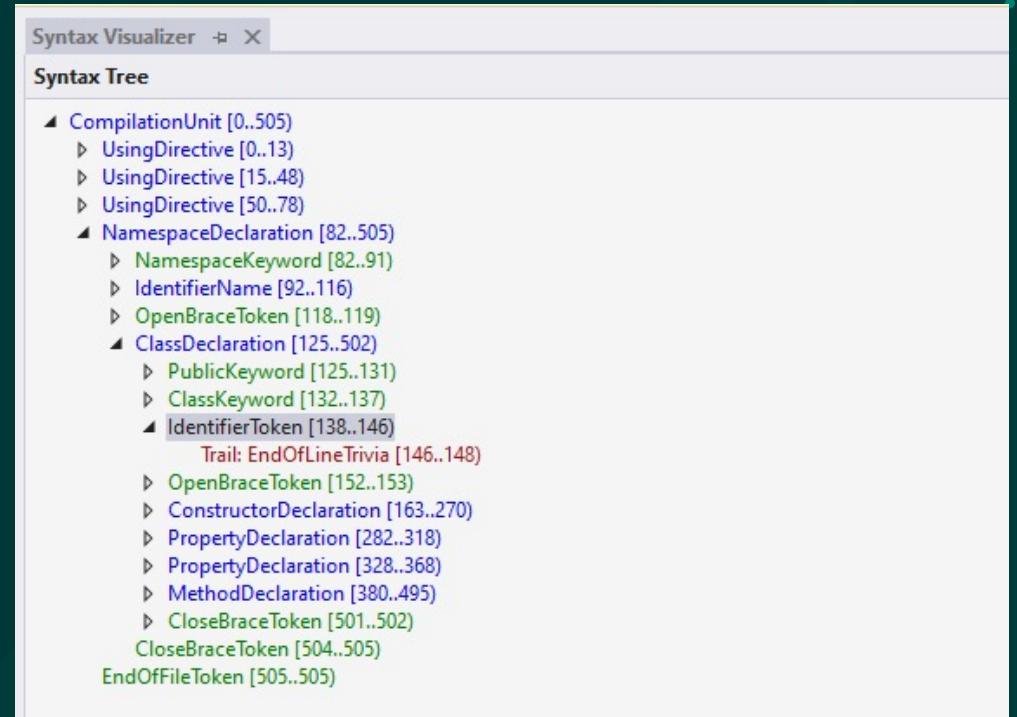


# Лексемы

12

SyntaxToken

IsKind(SyntaxToken) Kind()



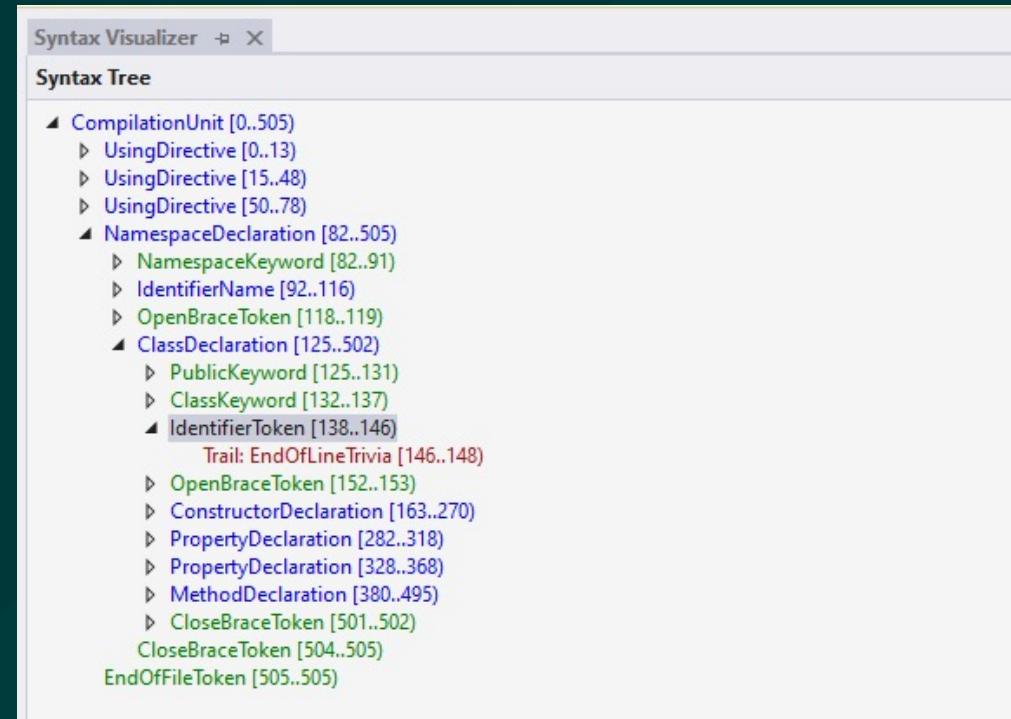
# Дополнительные элементы

13

Форматирование

Комментарии

IsKind(SyntaxToken) Kind()



# Фабрика

14

```
0 references | 0 changes | 0 authors, 0 changes
public async void Save()
{
    var values = new Dictionary<string, string>{{"Id", $"{Id.ToString()}"}, {"Name", $"{Name}"}};
    var content = new FormUrlEncodedContent(values);
    var response = await client.PostAsync("http://www.example.com/receiptile.aspx", content);
    var responseString = await response.Content.ReadAsStringAsync();
}
```



```
using System.Data.SqlClient;
namespace RoslynWebinarDemoLibrary
{
    4 references | Anton, 1 day ago | 1 author, 1 change
    public class Customer
    {
        0 references | Anton, 1 day ago | 1 author, 1 change
        public Customer(long id, string name)
        {
            Id = id;
            Name = name;
        }

        1 reference | Anton, 1 day ago | 1 author, 1 change
        public long Id { get; private set; }
        1 reference | Anton, 1 day ago | 1 author, 1 change
        public string Name { get; private set; }

        1 reference | 0 changes | 0 authors, 0 changes
        internal List<SqlParameter> GetParameters()
        {
            throw new NotImplementedException();
        }
    }
}
```

# Фабрика

15

```
1 reference | 0 changes | 0 authors, 0 changes
private static async Task<SyntaxNode> EditDocument(Document document)
{
    SyntaxTree syntaxTree = await document.GetSyntaxTreeAsync();
    SyntaxNode rootNode = await syntaxTree.GetRootAsync();
}
```

```
SyntaxNode nameSpaceNode = rootNode.ChildNodes().FirstOrDefault(n => n is NamespaceDeclarationSyntax);
if (nameSpaceNode == null)
    return rootNode;

ClassDeclarationSyntax classDeclarationSyntax = nameSpaceNode
    .ChildNodes()
    .FirstOrDefault(n => n is ClassDeclarationSyntax) as ClassDeclarationSyntax;

if (classDeclarationSyntax == null)
    return rootNode;
```

# Фабрика

16

```
string statement = $"var values = new Dictionary<string, string>{{{\"Id\", ${Id.ToString()}}, {"Name", ${Name}}}};" +  
    $"{Environment.NewLine}var content = new FormUrlEncodedContent(values);" +  
    $"{Environment.NewLine}var response = await client.PostAsync(\"http://www.example.com/receiptle.aspx\", content);"  
    $"{Environment.NewLine}var responseString = await response.Content.ReadAsStringAsync();"  
StatementSyntax statementSyntax = SyntaxFactory.ParseStatement(statement);  
BlockSyntax blockSyntax = SyntaxFactory.Block(new[] { statementSyntax });
```

0 references | 0 changes | 0 authors, 0 changes

```
public async void Save()  
{  
    var values = new Dictionary<string, string>{{"Id", ${Id.ToString()}}, {"Name", ${Name}}};  
    var content = new FormUrlEncodedContent(values);  
    var response = await client.PostAsync("http://www.example.com/receiptle.aspx", content);  
    var responseString = await response.Content.ReadAsStringAsync();  
}
```

# Фабрика

17

```
TypeSyntax voidTypeSyntax = SyntaxFactory.PredefinedType(SyntaxFactory.Token(SyntaxKind.VoidKeyword));

MethodDeclarationSyntax methodDeclarationSyntax = SyntaxFactory.MethodDeclaration(voidTypeSyntax, "Save");
methodDeclarationSyntax = methodDeclarationSyntax.AddModifiers(
    new[]
    {
        SyntaxFactory.Token(SyntaxKind.PublicKeyword),
        SyntaxFactory.Token(SyntaxKind.AsyncKeyword)
    });
methodDeclarationSyntax = methodDeclarationSyntax.WithBody(blockSyntax);
```

```
0 references | 0 changes | 0 authors, 0 changes
public async void Save()
{
    var values = new Dictionary<string, string>{{"Id", $"{Id.ToString()}"}, {"Name", $"{Name}"}};
    var content = new FormUrlEncodedContent(values);
    var response = await client.PostAsync("http://www.example.com/receipt.aspx", content);
    var responseString = await response.Content.ReadAsStringAsync();
}
```

# Иммутабельность

18

```
ClassDeclarationSyntax updatedClassDeclarationSyntax =
    classDeclarationSyntax.AddMembers(methodDeclarationSyntax);
SyntaxNode updatedRoot =
    rootNode.ReplaceNode(classDeclarationSyntax, updatedClassDeclarationSyntax);
return updatedRoot.NormalizeWhitespace();
```

```
var updateNode = await EditDocument(document);

var updatedSolution = solution.WithDocumentSyntaxRoot(document.Id, updateNode);

if (!workspace.TryApplyChanges(updatedSolution))
    Console.WriteLine("Something went wrong");
}
```

# Семантическая модель

Компиляция

Объекты

Типы

# Семантическая модель

20

## ISymbol

IPropertySymbol

IMethodSymbol

ITypeSymbol

INamedTypeSymbol

SymbolKind  
{get;}

String[]

IntPtr

void Do()

Type

# Семантическая модель

21

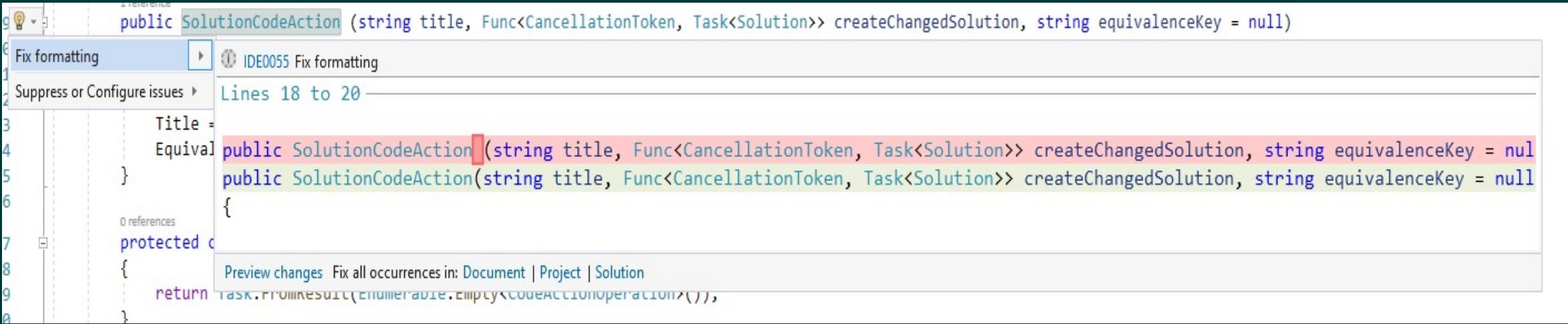
```
private async Task InspectNodes(Document document, SyntaxNode node)
{
    SemanticModel semanticModel = await document.GetSemanticModelAsync();

    ISymbol symbol = semanticModel.GetDeclaredSymbol(node);

    if(symbol is IPropertySymbol propertySymbol)
    {
        ITypeSymbol typeSymbol = propertySymbol.Type;
        ImmutableArray<INamedTypeSymbol> interfaces = typeSymbol.AllInterfaces;
    }
}
```

# Расширения IDE

22



The screenshot shows a code editor window with the following code snippet:

```
public SolutionCodeAction (string title, Func< CancellationToken, Task< Solution >> createChangedSolution, string equivalenceKey = null)
{
    Title = title;
    EquivalenceKey = equivalenceKey;
}

public SolutionCodeAction(string title, Func< CancellationToken, Task< Solution >> createChangedSolution, string equivalenceKey = null)
{
}
```

A context menu is open at the end of the first constructor's opening brace, with the "Fix formatting" option selected. A tooltip for "IDE0055 Fix formatting" is displayed, along with the range "Lines 18 to 20". Below the code, a preview message says "Preview changes Fix all occurrences in: Document | Project | Solution".



Code Refactoring (.NET Standard)

Create a C# refactoring, deployed as a VSIX extension

C#

Windows

Linux

macOS

VSSDK

Roslyn

Extensions

# Расширения IDE

23

VSIX

CodeRefactoringProvider

CodeAction

# CodeRefactoringProvider

24

```
[ExportCodeRefactoringProvider(LanguageNames.CSharp, Name = nameof(VeeamBackupCodeRemotingReadyInterfaceCodeRefactoringProvider)), Shared]
1 reference
internal class VeeamBackupCodeRemotingReadyInterfaceCodeRefactoringProvider : CodeRefactoringProvider
{
    0 references
    public sealed override async Task ComputeRefactoringsAsync(CodeRefactoringContext context)...
}
```



# CodeAction

25

Task

Preview

Post actions

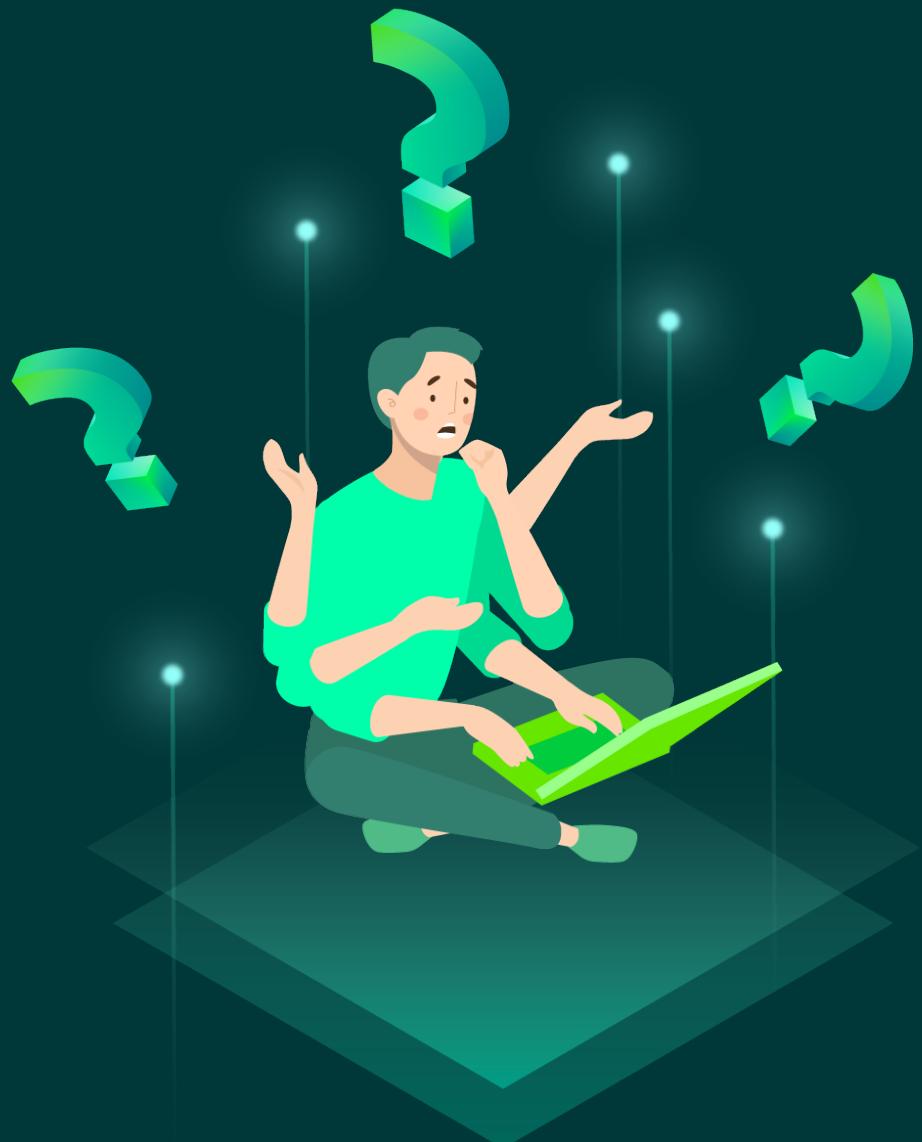
# SimpleCodeAction

26

DocumentChangeAction

SolutionChangeAction

DIY



Опыт использования  
400 интерфейсов  
на новый протокол?

# Посчитаем



=

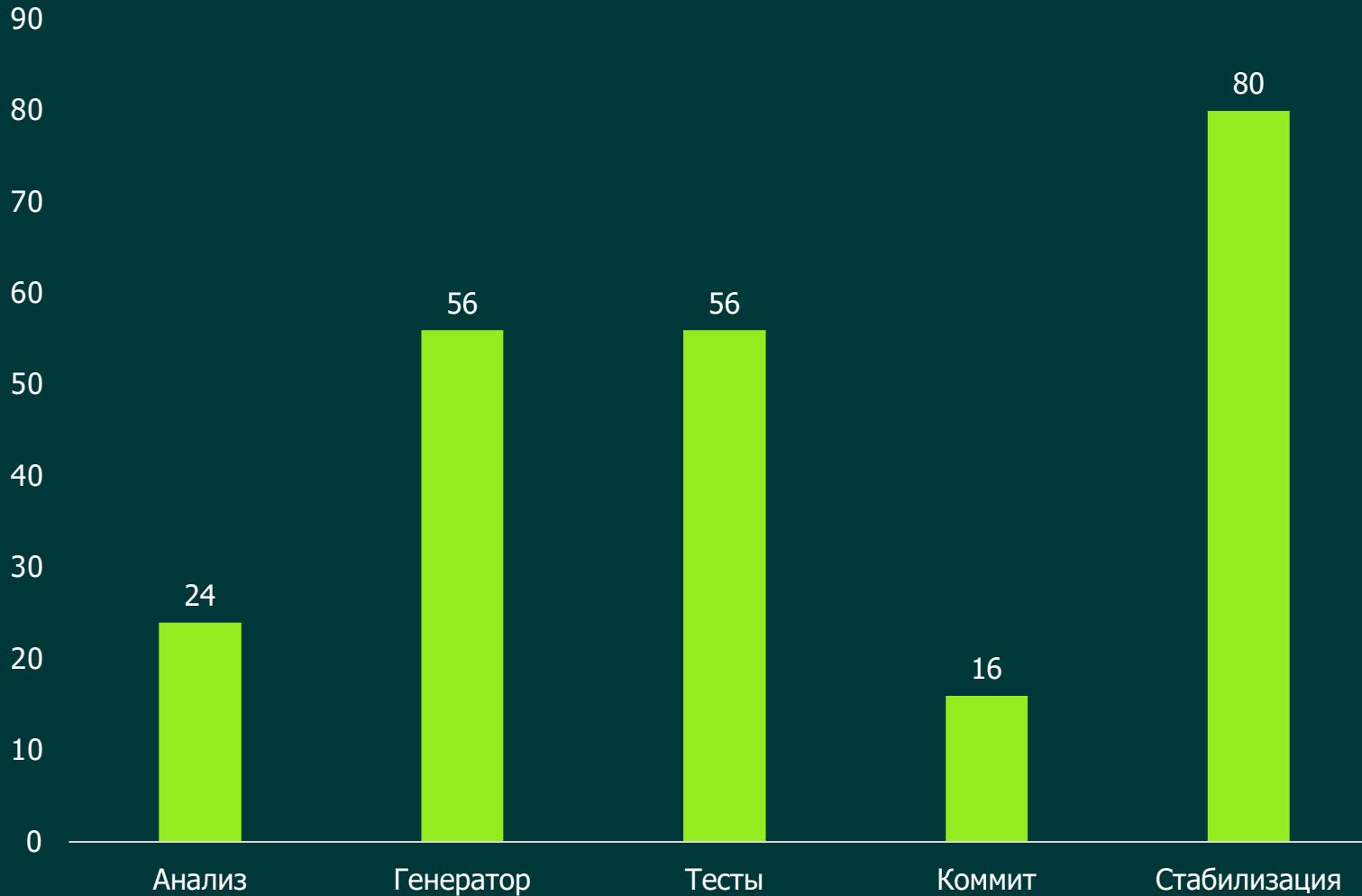
3200



=

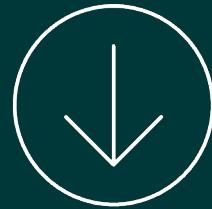
320

# Решение

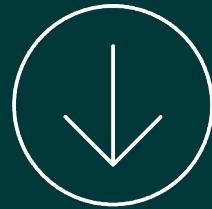


# Плюсы

Стоимость  
разработки



Рутинна



Качество  
кода



# Минусы

Code Style



Не панацея



Ресурсы



# Где использовать?

Анализ кода

Код стайл

MVC

Обертки для вызовов

Проверки на билдерах

# Что почитать?

[Roslyn Cookbook by Manish Vasani](#)

[Microsoft. Understand the .NET Compiler Platform SDK model](#)

[Official GitHub repository](#)



Senior C# Developer  
(с возможностью переезд в Прагу)



Backend C# Developer



C# Developer  
(с возможностью релокации в Прагу)

# Thank you



[www.aeremin.net](http://www.aeremin.net)  
Twitter: @eremintony  
Telegram <https://t.me/tonnyeremin>