Projekt Roslyn

…IN NOVOSTI C# 6.0

Miha Markič, MVP C#, miha@rthand.com

A moram vedeti kaj je Roslyn?

Večini ni potrebno

Vseeno je dobro vedeti, kaj je pod pokrovom

Naprednim razvijalcem ne škodi

Zakaj Roslyn

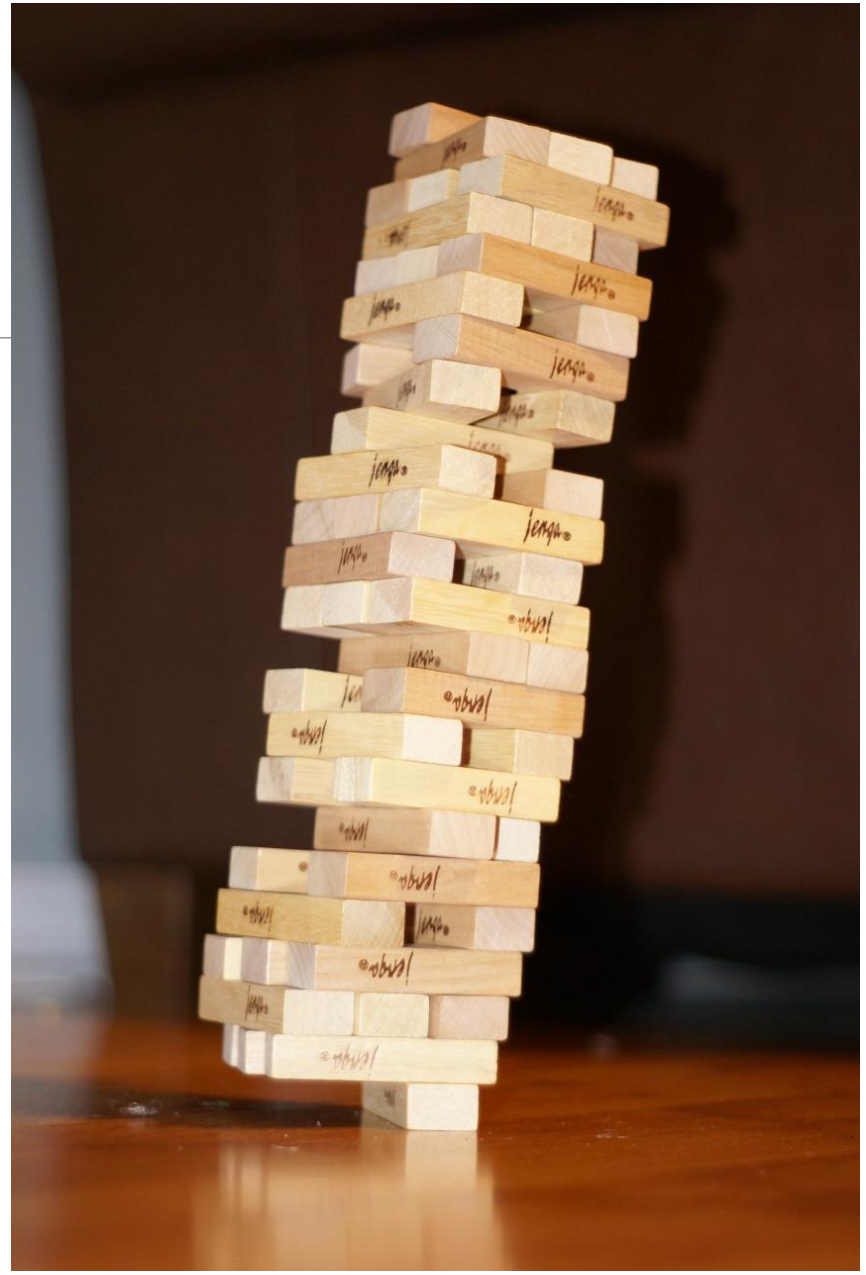
Huda stvar napisana v C++, z leti pridobivala težo

Težko dodajati/spreminjati jezik

Obstoječi prevajalnik je črna škatla

Visual Studio razširitve nimajo dostopa do prevajalnika

Noben drug si ne more pomagati s prevajalnikom



Roslyn

Prevajalnik kot storitev

5 letni razvoj

Zaenkrat **C#** in VB.NET

Odprtokoden (https://roslyn.codeplex.com/)

Napisan v C# 6.0

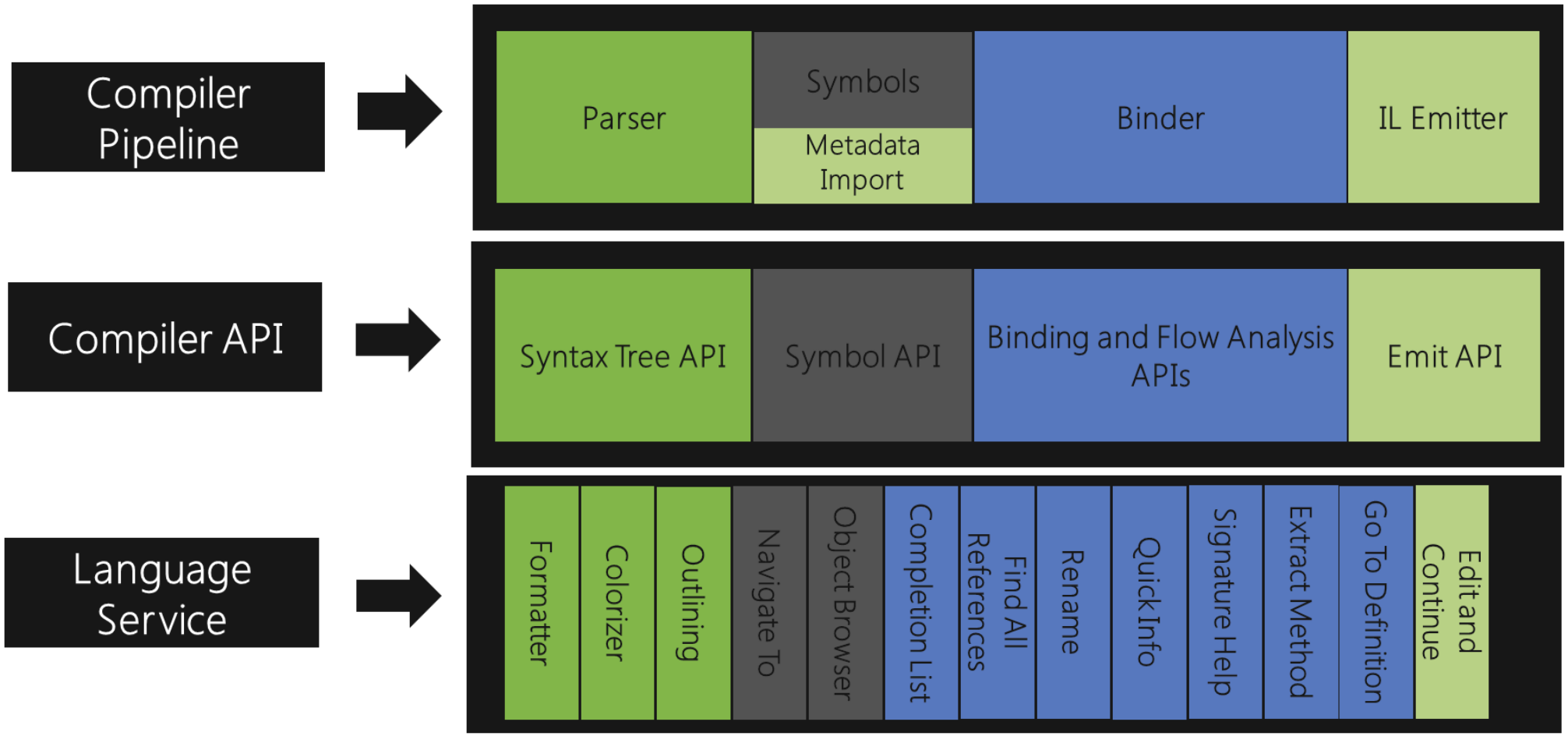
Trenutno podpira Visual Studio 14 CTP3 (za 2013 ni več posodobljen)

Razpečevanje preko NuGet

- `Install-Package Microsoft.CodeAnalysis -Pre`

Končna prva verzija predvidoma z Visual Studio 14

Prevajalnik



Prevajalnik – sintaktična drevesa

Slika izvorne kode v izbranem trenutku

Vsebuje vse podatke

- Celotna izvorna koda
- Napake

Povratna

- Da se povrniti v izvorno kodo, kot je bila napisana

Nespremenljiva struktura

- Varna za večnitne operacije

Prevajalnik – sintaktična drevesa

Vozlišča (Syntax Nodes)

- Predstavljena s podrazredom SyntaxNode
- Predstavlja sintaktične konstrukte (deklaracije, stavki, izrazi(expressions)…)
- Niso zaključni
- Ima otroke
- Ima starša, razen korenski

Žetoni (Syntax Tokens)

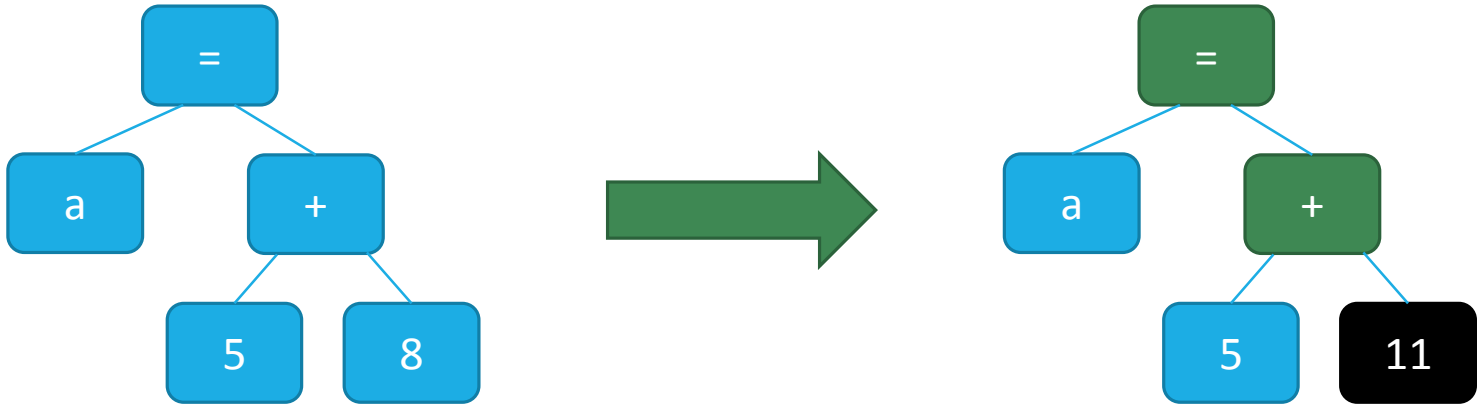
- Zaključki
- Ključne besede, identifikatorji, črke, ločila, vrednosti

Nepomembneži (Syntax Trivia)

- Okraski (presledki, komentarji, …)
- Del žetonov, ne pa drevesa

Prikaz sintaktičnega drevesa

Spremembe dreves



Spremembe dreves

```
SyntaxTree drevo = CSharpSyntaxTree.ParseText(„…“);
SyntaxNode koren = (CompilationUnitSyntax)drevo.GetRoot();
var staraVrednost5Vozel = koren.ChildNode()…
var novaVrednost11Vozel = SyntaxFactory.LiteralExpression(
  SyntaxKind.NumericLiteralExpression,
  SyntaxFactory.Literal(11));
koren.ReplaceNode(staraVrednost5Vozel, novaVrednost11Vozel);
```

Semantični model

```
CSharpCompilation prevedeno = CSharpCompilation.Create(„Tubo")
                .AddReferences(
                    new MetadataFileReference(
                        typeof(object).Assembly.Location))
                .AddSyntaxTrees(drevo);
var model = prevedeno.GetSemanticModel(tree);
var info = model.GetSymbolInfo([SyntaxNode]);
```

Prevajalnik

Diagnostics API

- Centralizirana diagnostika
- Napake, opozorila
- ki jih zazna prevajalnik
- In ki jih posredujejo vtičniki
- prevedeno.GetDiagnostics()

Scripting API (REPL, skripte)

- Izvajanje skript
- Izvajanje REPL (prečitaj, ovrednoti, natisni, zankaj)

Workspace API

Analiza kode in preurejanje kode (refactoring) na nivoju projekta ali rešitve

- Ni potrebno „ročno“ analiziranje

Neodvisno od razvojnega okolja (npr. Visual Studio)

Zbirka pogosto uporabljenih orodij (i.e. Find All References, Formatting, …)

Razvojno okolje uporablja te storitve

Demonstracija

Vtičnik za dodajanje beseda Async metodam, ki izvajajo asinhrone operacije

Novosti C# 6.0

Dodaj

```
<LangVersion>Experimental</LangVersion>
```

v .csproj datoteko.

http://tinyurl.com/kg62qnt

Rdeča nit je zmanjšanje nepotrebne kode

Novosti C# 6.0
Začetna vrednost avto-lastnosti

```
public class Tubo {
    public string Naziv { get; set; } = "Miha";
}
```

```
public class Tubo {
    public string Naziv { get; set; }
    public Tubo() {
        Naziv = "Miha";
    }
}
```

Novosti C# 6.0
Osnovni konstruktor

```
public class Tubo(string naziv) {
    public string Naziv { get; } = naziv;
}

public class Tubo {
    private string naziv;
    public Tubo(string naziv) {
        this.naziv = naziv;
    }
    public string Naziv { get { return naziv; } }
}
```

Novosti C# 6.0
Lambda kot telo

Metode

```
public Point Premakni(int dx, int dy) => new Point(x + dx, y + dy);
```

Lastnosti

```
public string Naziv => Ime + " " + Priimek;
```

Novosti C# 6.0
Deklaracija *out* argumenta

```
if (int.TryParse("11", out int i))
{
    Console.WriteLine(i);
}


int i;
if (int.TryParse("11", out i))
{
    Console.WriteLine(i);
}
```

Novosti C# 6.0

Pogojni ničelni operater (Null-conditional operaters)

```
int? številoStrank = stranke?.Length;
int? številoStrank = stranke != null ? stranke.Length: (int?)null;


string telefonska = oseba?.Mati?.Telefonska;
string telefonska = null;
if (oseba != null && oseba.Mati != null)
  telefonska = oseba.Mati.Telefonska;
```

Novosti C# 6.0

Inicializatorji v strukturah

Using za statične tipe

Filtri izjem

nameof ključna beseda

Inicializatorji indeksov

await v catch/finally sklopih

Vrednosti v binarnem zapisu in presledki med ciframi

In še kaj…nič še ni dokončnega.

http://tinyurl.com/me5bmlt

Righthand Miha Markič s.p.

http://blog.rthand.com

miha@rthand.com

Microsoft MVP za C#, SLODUG vodja, DevExpress MVP, LLBLGenPro Partner, …

www.slodug.si

Vprašanja

?

Koristne povezave

CodeRush
http://tinyurl.com/d7d9xn

Roslyn
https://roslyn.codeplex.com/

Visual Studio 14 CTP 3 (blog vnos z vsemi informacijami)
http://tinyurl.com/pwhepbe

PostSharp
http://www.sharpcrafters.com/

Slike

Jenga by Ed Garcia (source http://tinyurl.com/mazlm8l)