



Provozně
ekonomická
fakulta

20.4. 2016, Brno

Autor práce:

Bc. Lenka Chalupová

Vedoucí práce:

Ing. Jiří Lýsek, Ph.D.

Automatizace kontroly kódu během vývoje



- Analýza dostupných nástrojů pro kontrolu kódu
- Na základě požadavků zadavatele navrhnout a implementovat nástroj pro automatickou kontrolu kódu během vývoje
- Kontrola kódu na základě definovaných pravidel
 - Syntaktická pravidla pro psaní kódu
 - Pravidla pro statickou analýzu kódu

- .Net Framework
- Programovací jazyk C#
- Platforma Roslyn
- Vývojové prostředí Microsoft Visual Studio 2015

Příklady syntaktických pravidel pro psaní kódu

Identifikátor	Popis pravidla
Class, Structure	Psáno pomocí Pascalovy notace, nepoužívat stejný název u třídy a namespace.
Namespace	Psáno pomocí Pascalovy notace.
Method	Psáno pomocí Pascalovy notace.
Public property	Psáno pomocí Pascalovy notace, pokud je datového typu boolean použít prefix „Is“.
Non public field	Psáno pomocí Pascalovy notace, použít prefix „m“.
Interface	Psáno pomocí Pascalovy notace, použít prefix „I“.
Constant	Psáno velkými písmeny s využitím symbolu '_' jako rozdělovače slov.

Příklady pravidel pro statickou analýzu kódu

Název pravidla	Popis pravidla
UserHostAddress	Kontroluje, zda je v kódu použito <code>Request.UserHostAddress</code> , použít místo toho <code>RequestContext.UserHostAddress</code> .
Redirect	Kontroluje, zda je použito v kódu <code>Request.Redirect</code> , upozorní vývojáře, že se nemá používat.
PostBack	Kontroluje, zda je v kódu použito <code>IsPostBack</code> , místo toho použít <code>RequestHelper.IsPostBack()</code> .
Callback	Kontroluje, zda je v kódu použito <code>IsCallback</code> , místo toho použít <code>RequestHelper.IsCallback()</code> .
Browser	Kontroluje, zda je v kódu použito <code>Request.Browser.Browser</code> , použít místo toho <code>BrowserHelper.GetBrowser()</code> .

Kroky vytvoření analyzátoru pro kontrolu kódu

1. Analýza pravidla
2. Vytvoření diagnostiky pro analýzu pravidla
3. Vytvoření návrhu pro opravu kódu
4. Testování analyzátoru pro dané pravidlo

Ukázka analyzátoru syntaktického pravidla pro psaní konstant

```
public override void Initialize(AnalysisContext context)
{
    context.RegisterSymbolAction(AnalyzeField, SymbolKind.Field);
}

private void AnalyzeField(SymbolAnalysisContext context)
{
    var field = (IFieldSymbol)context.Symbol;
    if (field.IsConst && !field.Type.TypeKind.Equals(TypeKind.Enum))
    {
        if (field.Name.ToCharArray().Any(char.IsLower))
        {
            //vytvoření diagnostiky pravidla
            var diagnostic = Diagnostic.Create(Rule, field.Locations[0], field.Name);
            context.ReportDiagnostic(diagnostic);
        }
    }
}
```

Ukázka opravy kódu syntaktického pravidla pro psaní konstant

```
private const string title = "Make uppercase for constant";

public sealed override ImmutableArray<string> FixableDiagnosticIds
{
    get { return ImmutableArray.Create(Analyzerers.ConstAnalyzer.DiagnosticId); }
}

public sealed override FixAllProvider GetFixAllProvider()
{
    return WellKnownFixAllProviders.BatchFixer;
}

public sealed override async Task RegisterCodeFixesAsync(CodeFixContext context)
{
    var root = await context.Document.GetSyntaxRootAsync(context.CancellationToken).ConfigureAwait(false);
    var diagnostic = context.Diagnostics.First();
    var token = root.FindToken(diagnostic.Location.SourceSpan.Start);
    context.RegisterCodeFix(
        CodeAction.Create(title: title,
            createChangedSolution: c => MakeUpper(context.Document, token, c),
            equivalenceKey: title),
        diagnostic);
}
```


Ukázka opravy kódu syntaktického pravidla pro psaní konstant

```
private async Task<Solution> MakeUpper(Document document, SyntaxToken declaration, CancellationToken
    cancellationToken)
{
    var name = declaration.ValueText;
    var semantiModel = await document.GetSemanticModelAsync(cancellationToken);
    var symbol = semantiModel.GetDeclaredSymbol(declaration.Parent, cancellationToken);
    var solution = document.Project.Solution;
    var upperChar = name.ToCharArray().Any(char.IsUpper);
    if (upperChar)
    {
        var firstLow = char.ToLower(name[0]) + name.Substring(1);
        var upper = MatchUpper(firstLow);
        var newName = $"{upper}";
        return await Renamer.RenameSymbolAsync(solution, symbol, newName, solution.Workspace.Options,
            cancellationToken);
    }
    else
    {
        var newName = $"{name.ToUpper()}";
        return await Renamer.RenameSymbolAsync(solution, symbol, newName, solution.Workspace.Options,
            cancellationToken);
    }
}
```

Ukázka zobrazení opravy syntaktického pravidla pro psaní konstant

```
public const string popis = "DESCRIPTION";
```



Make uppercase for constant

Constant Field popis does not uppercase.

...

```
public const string popis = "DESCRIPTION";  
public const string POPIS = "DESCRIPTION";
```

...

Preview changes

Fix all occurrences in: [Document](#) | [Project](#) | [Solution](#)

```
public const string JmenoNejakeKonstatnty = "nazev";
```



Make uppercase for constant

Constant Field JmenoNejakeKonstatnty does not uppercase.

...

```
public const string JmenoNejakeKonstatnty = "nazev";  
public const string JMENO_NEJAKE_KONSTATNTY = "nazev";
```

...

Preview changes

Fix all occurrences in: [Document](#) | [Project](#) | [Solution](#)

Testování nástroje

Error List - Current Project (CMSApp)

Current Project | 162 of 167 Errors | 15976 of 16273 Warnings | 0 Messages | Search Error List

	Code	Description	Project	File	Line	S..
▶	✘ Delegate	Field OnInitializeLabels does not start with capital letter or does not suffix 'EventHandler'.	CMSApp	BaselmageEditor.a scx.cs	507	Active
▶	✘ Delegate	Field OnSavelmage does not start with capital letter or does not suffix 'EventHandler'.	CMSApp	BaselmageEditor.a scx.cs	523	Active
▶	✘ Delegate	Field OnGetMetaData does not start with capital letter or does not suffix 'EventHandler'.	CMSApp	BaselmageEditor.a scx.cs	532	Active
▶	✘ Constant	Const controlPath are not uppercase.	CMSApp	SelectDocumentG roup.ascx.cs	19	Active
▶	✘ Parameter	Field CurrentStagingTaskGroup does not start with small letter or all letters are uppercase.	CMSApp	StagingTaskGroup Menu.ascx.cs	220	Active
▶	✘ Constant	Const SelectedSuffix are not uppercase.	CMSApp	UniMenuButtons. ascx.cs	52	Active
▶	✘ Constant	Const Section508 are not uppercase.	CMSApp	AccessibilityValida tor.ascx.cs	96	Active
▶	✘ Constant	Const StancaAct are not uppercase.	CMSApp	AccessibilityValida tor.ascx.cs	101	Active
▶	✘ Constant	Const mSkipUrlsStartingWith are not uppercase.	CMSApp	LinkChecker.ascx. cs	39	Active
▶	✘ Enumeration	Field Return Type does not start with capital letter or does not suffix 'Enum'.	CMSApp	CountrySelector.a scx.cs	27	Active
▶	✘ Enumeration	Field ValueSourceType does not start with capital letter or does not suffix 'Enum'.	CMSApp	TextBoxDefaultSet ting.ascx.cs	19	Active
▶	✘ Parameter	Field Value does not start with small letter or all letters are uppercase.	CMSApp	GridColumnDesig ner.aspx.cs	365	Active
▶	✘ Parameter	Field Name does not start with small letter or all letters are uppercase.	CMSApp	GridColumnDesig ner.aspx.cs	365	Active

Error List | Output

- Testování ve společnosti
- Splněny požadavky a cíle práce
- Návrh budoucího vývoje

DĚKUJI ZA POZORNOST