

▲ C# Kod Denetim

▷ Properties

▷ References

▲ Checks

▷ C# AssignmentInIfTester.cs

▷ C# DeadCodeTester.cs

▷ C# EmptyCatchTester.cs

▷ C# GotoTester.cs

▷ C# ITester.cs

▷ C# LongFunctionTester.cs

▷ C# NestedIfTester.cs

▷ C# NoCommentTester.cs

▷ C# ParamCountTester.cs

▷ C# UnreachableCodeTester.cs

Her bir kusurlu kod için bir sınıf yazılıyor.

Tüm bu sınıflar ITester arayüzünden türüyor.

▲ Parser

CPP14.tokens

▷ C# CPP14BaseListener.cs

▷ C# CPP14Lexer.cs

CPP14Lexer.tokens

▷ C# CPP14Listener.cs

▷ C# CPP14Parser.cs

Antlr tarafından oluşturulan C++ lexer ve parser dosyaları.

▲ Resources

reload.png

Start-icon.png

▷ About.cs

App.config

▷ BaseForm.cs

▷ C# Error.cs

▷ MainForm.cs

▷ C# Program.cs

▷ C# Settings.cs

App.config kusurlu kodlar tespit edilmesine ilişkin ayarları içeriyor. Örneğin :

```
<add key="MaxParamCount" value="3"/>
```

MainForm.cs programın ana penceresini oluşturuyor. Parse etme ve analiz işlemleri bu sınıfta yapılıyor.

Error sınıfı bulunan kusurlu kodların satır numaralarını ve açıklamalarını tutar.

MainForm sınıfı

C++ kaynak kodu dosyasının nasıl analiz edildiğine ait kod ve açıklamaları

```
private void Parse()
{
    using (FileStream filestream = new FileStream(txtFile.Text, FileMode.Open))
    {
        AntlrInputStream inputstream = new AntlrInputStream(filestream);
        CPP14Lexer lexer = new CPP14Lexer(inputstream);
        CommonTokenStream tokens = new CommonTokenStream(lexer);
        CPP14Parser parser = new CPP14Parser(tokens);
        CPP14Parser.TranslationunitContext translationunit = parser.translationunit();

        ...

        Walk(translationunit);

        ...
    }
}

private void Walk(IParseTree t)
{
    if (t is IErrorNode)
    {
        return;
    }
    Testers.ForEach(tester => tester.Test(t));
    for (int i = 0; i < t.ChildCount; i++)
    {
        Walk(t.GetChild(i));
    }
}
```

1. Dosyayı oku.
2. Kaynak kodu tokenlarına ayırıştır.
3. Parse Tree'yi oluştur.
4. Ağacı gez. Her düğümü tüm testerlar ile test et. Bazı testerlar alt ağacı da (sub tree) gezebiliyor. Dosya tüm ağacı temsil ederken, for döngüsü alt ağacı meydana getir.
5. Eğer varsa hataları kullanıcıya göster.

ITester

```
void Test(IParseTree t);  
LinkedList<Error> GetErrors();
```