**什么是语义**

语法规定了代码怎么写，而语义规定了代码表达了什么意思

**了解编译和符号**

Compilation 实例是编译器所看见的单个项目

符号是一个名词（如命名空间System，类Console等），其包含了描述该名词的信息（如：类型、命名空间、成员和变量的实体）

Compilation 允许你查找“符号”

通过Microsoft.CodeAnalysis.CSharp.CSharpCompilation创建Compilation实例时

**查询符号**

如下查找符号示例：

const string programText =

@"using System;

using System.Collections.Generic;

using System.Text;

namespace HelloWorld

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(""Hello, World!"");

}

}

}";

SyntaxTree tree = CSharpSyntaxTree.ParseText(programText);

// 获取根语法节点

CompilationUnitSyntax root = tree.GetCompilationUnitRoot();

// 获取编译模型

var compilation = CSharpCompilation.Create("HelloWorld")

.AddReferences(MetadataReference.CreateFromFile(

typeof(string).Assembly.Location))

.AddSyntaxTrees(tree);

// 获取语义模型

SemanticModel model = compilation.GetSemanticModel(tree);

// 获取 "using System;" 语法

UsingDirectiveSyntax usingSystem = root.Usings[0];

// 获取 name 语法，即 System

NameSyntax systemName = usingSystem.Name;

// 从语义模型获取对应的符号信息

SymbolInfo nameInfo = model.GetSymbolInfo(systemName);

// 获取符号

// 不是所有的语法都具有符号，像 "using System;" 语法就不具有符号，应为它不是一个符号

var systemSymbol = (INamespaceSymbol)nameInfo.Symbol;

foreach (INamespaceSymbol ns in systemSymbol.GetNamespaceMembers())

{

Console.WriteLine(ns);

}