

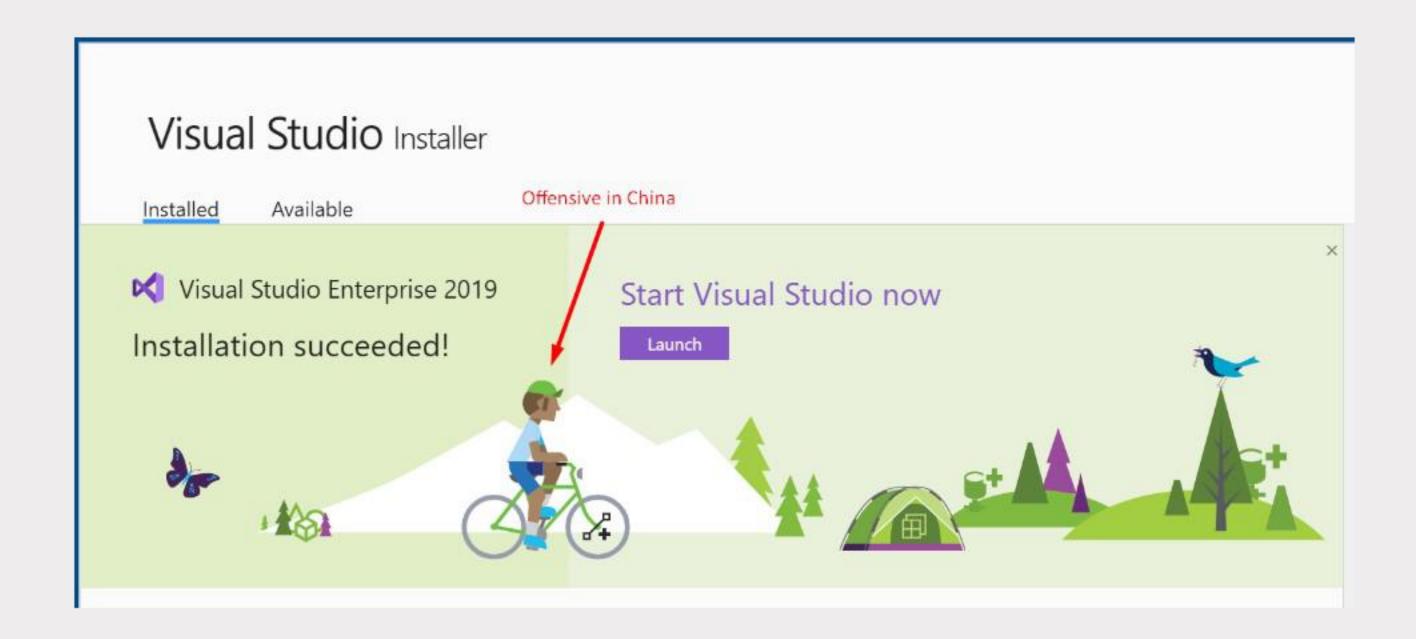
VisualStudio 2019 和 C# 8.0

林德熙







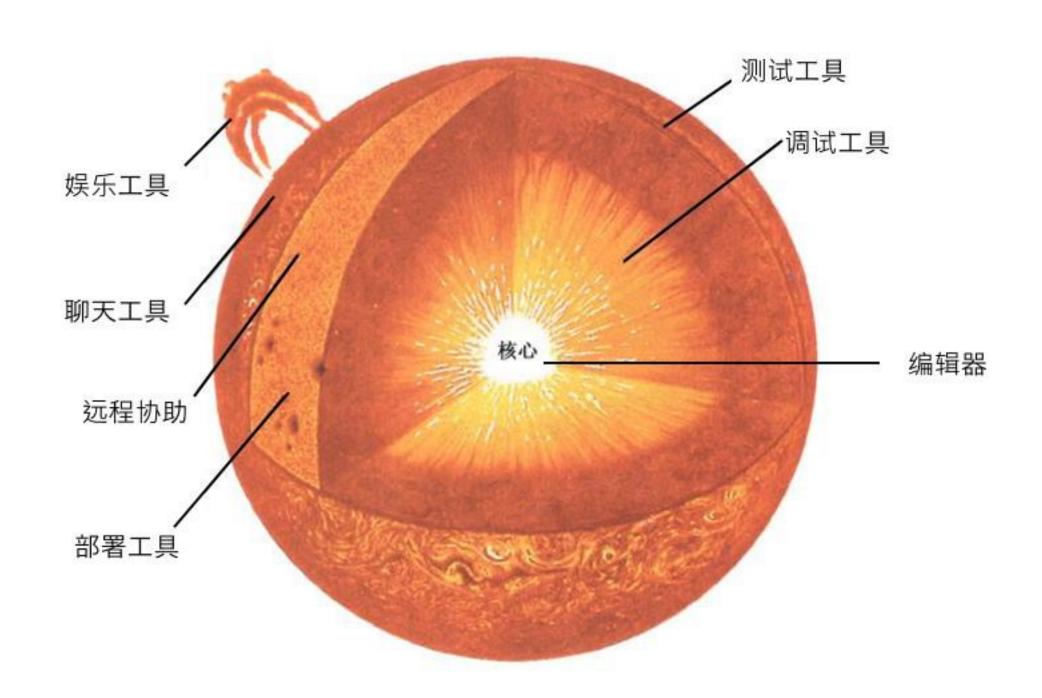


CVTE











打开 VisualStudio 玩一下 💢



全新欢迎界面





Visual Studio 2019

Open recent

ConsoleApp1

WindowsTestApp1.sln 11/16/2018 1:24 PM #

C\\U sers\Administrator\sounce\repos\WindowsTestApp1

CalculatorTutorial.sin 11/16/2018 1:24 PM P

CtyU sers\Administrator\Downloads\CalculatorTutorial

ClassLibrary2.sln 11/16/2018 1:30 PM

ClassLibrary1.sln 11/16/2018 1:26 PM

WindowsTestApp2.sln 11/16/2018 1:21 PM

C1/Users\Administrator\source\repos\WindowsTestApp2

C\/U sers\Administrator\sounce\repos

C\Users\Administrator\sounce\repos\ClassLibrary2

C\\Users\Administrator\sounce\repos\ClassLibrary1

ConsoleApp1.sin 11/16/2018 1:19 PM

C\/U sers\/Administrator\/sounce\repos\/ConsoleApp1

GitCloneSample.sln 11/16/2018 1:17 PM

C/...\Administrator\Downloads\Sample Solutions\Online Repos\GitCloneSample



Clone or checkout code

Get code from an online code repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or sln file



Open a local folder

Navigate and edit code within any folder



11/16/2018 1:20 PM

Create a new project

Choose a project template with code scaffolding to get started



Continue without code

启动慢

格式化卡

打开项目慢

对VS固有印象是什么?

复制粘贴卡

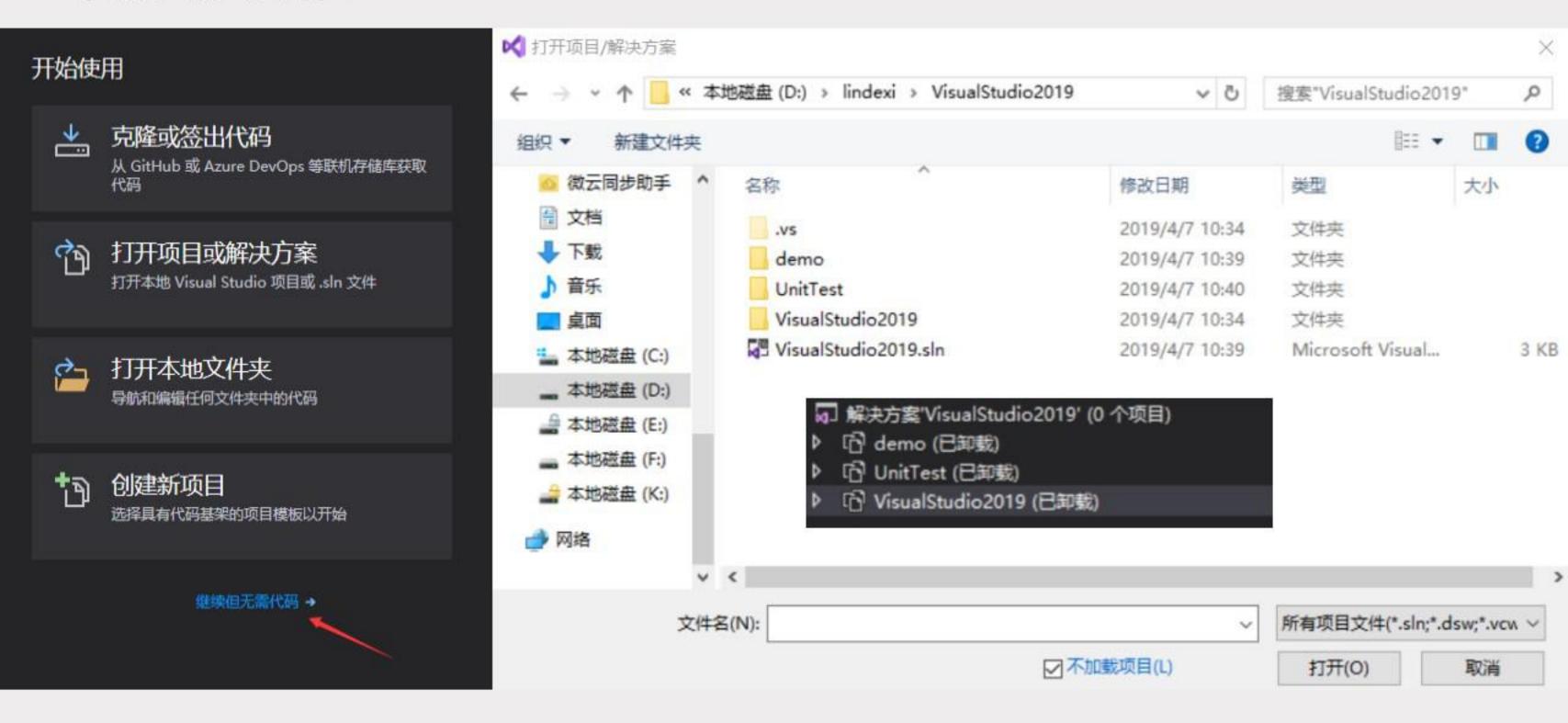
分析项目慢

输入卡

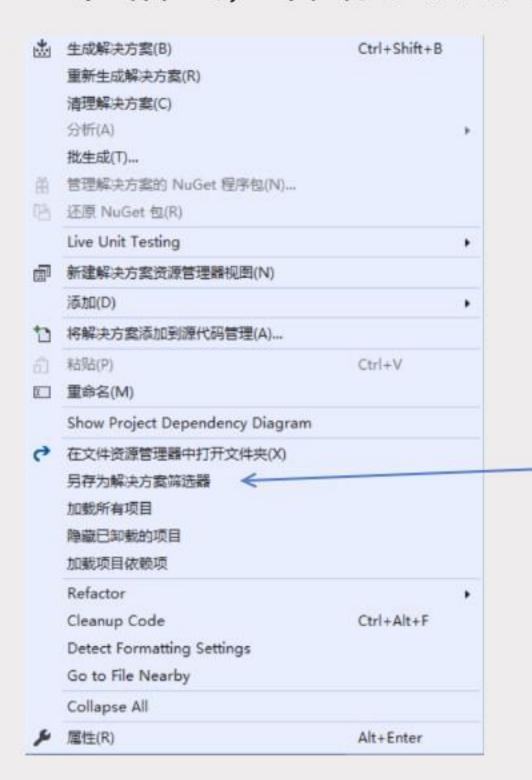


又卡又慢?💢

不加载大项目



sInf 格式,保存大项目里面默认不加载项目





文件名(N):		~
保存类型(T):	解决方案筛选器文件(*.slnf)	Ÿ



- UnitTest

 VisualStudio2019

 gitattributes

 gitignore

 vsconfig

 VisualStudio2019 Core.sln

 VisualStudio2019 with demo.sln

 VisualStudio2019 with test.sln

 VisualStudio2019.sln
 - 只带核心项目的 只带demo的 只带单元测试的

UnitTest
VisualStudio2019
gitattributes
gitignore
vsconfig
VisualStudio2019 Core.sInf
VisualStudio2019 with demo.sInf
VisualStudio2019 with test.sInf
VisualStudio2019.sIn

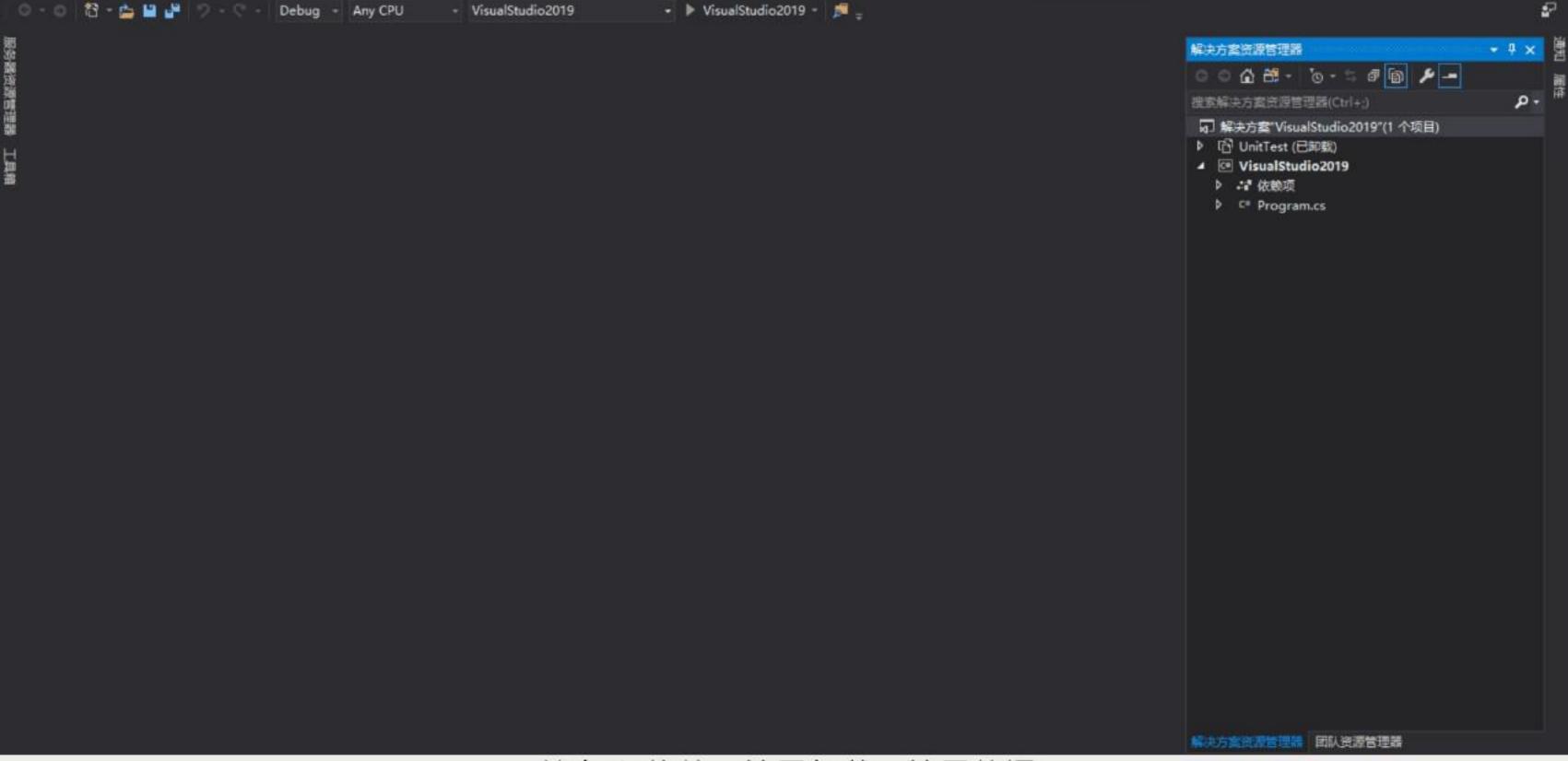
sInf=sIn+filter 一次更改,所有项目生效

添加一个项目需要修改所有工程

```
{
    "solution": {
        "path": "VisualStudio2019.sln",
        "projects": [
            "VisualStudio2019\\VisualStudio2019.csproj"
        ]
    }
}
```

依赖于sln文件,只指定需要加载的项目





单个sln依赖,按需加载,按需编译



解决卡的问题



编辑性能优化



差异显示模式(D):	整行
□ 显示差异概述边	距
✔ 在复制/剪切时	夏制格式文本
最大长度(L): 10	240
☑ 键入时自动取消	长时间运行的辅助操作
	か是+は) なだり
自动调整允许	10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0



编辑工具 💢

```
3 Enamespace BirinugeGurwainaynanu
    {
4
        0 个引用
        class Program
5 E
 6
           0 个引用
           static void Main(string[] args)
8
9
               var foo = new Foo();
               int n = (int) 2
10
               Console.WriteLine(value: "Hello World!");
12
13
14
        1 个引用
15 E
        class Foo
16
           0 个引用
           public string F { get; set; }
17
18
19
20
                            € 1
                                             > | d =
                                     1 1
                                                 Find Results
                      导航到下一个错误或警告(Alt+PgDn)
 € Ø1 11 → d+
```



```
0 个引用
                     static void Main(string[] args)
                          var foo = new Foo();
                          int n = (int) 2;
        10
                          Console.WriteLine(value: "Hello World!");
        11
        12
        13
        14
                 1 个引用
        15
                 class Foo
        16
                     0 个引用
                     public string F { get; set; }
        17
        18
        19
        20
         ← 30 A1 → d+
输出 错误列表 程序包管理器控制台 断点 Find Results
Local variable 'n' is only assigned but its value is never used
```



CVTE

Dream-Future

取消

确定



社区化

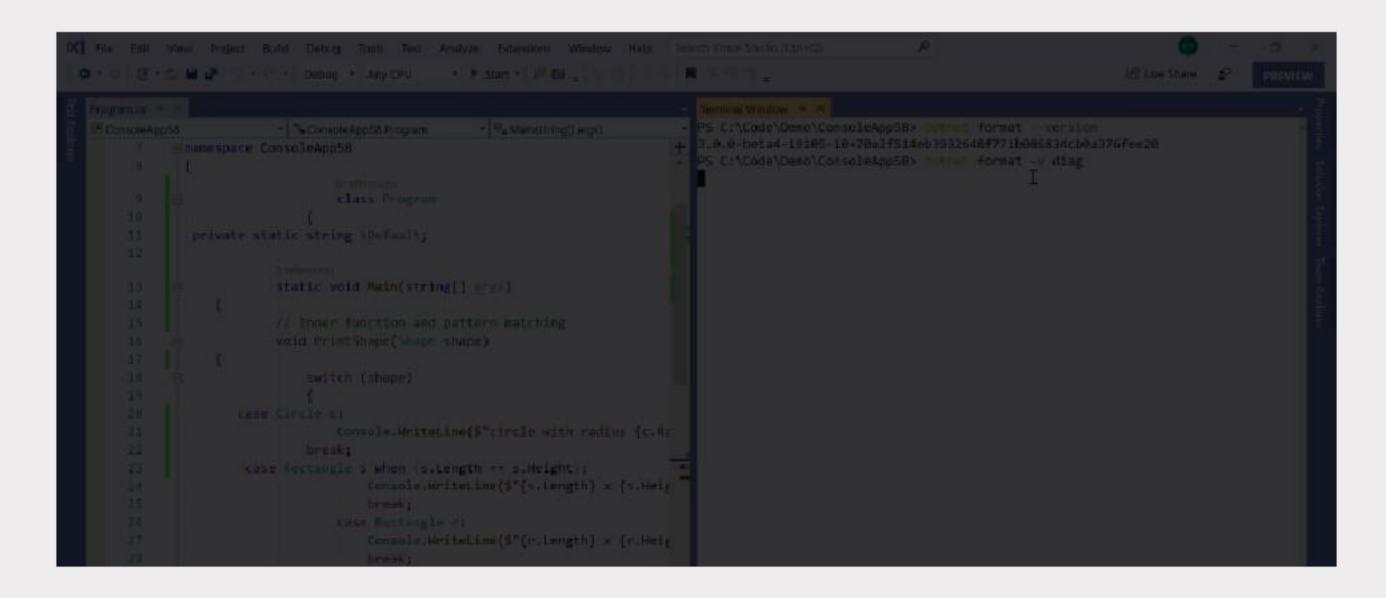
统一的格式化工具

Dotnet Format

 You can now apply code style preferences from the command-line with the dotnet format global tool. To install, you will need .NET Core 2.1 SDK or later. Run the following command in your favorite terminal: dotnet tool install -g dotnet-format --version 3.0.0-beta4-19105-10



https://github.com/dotnet/format



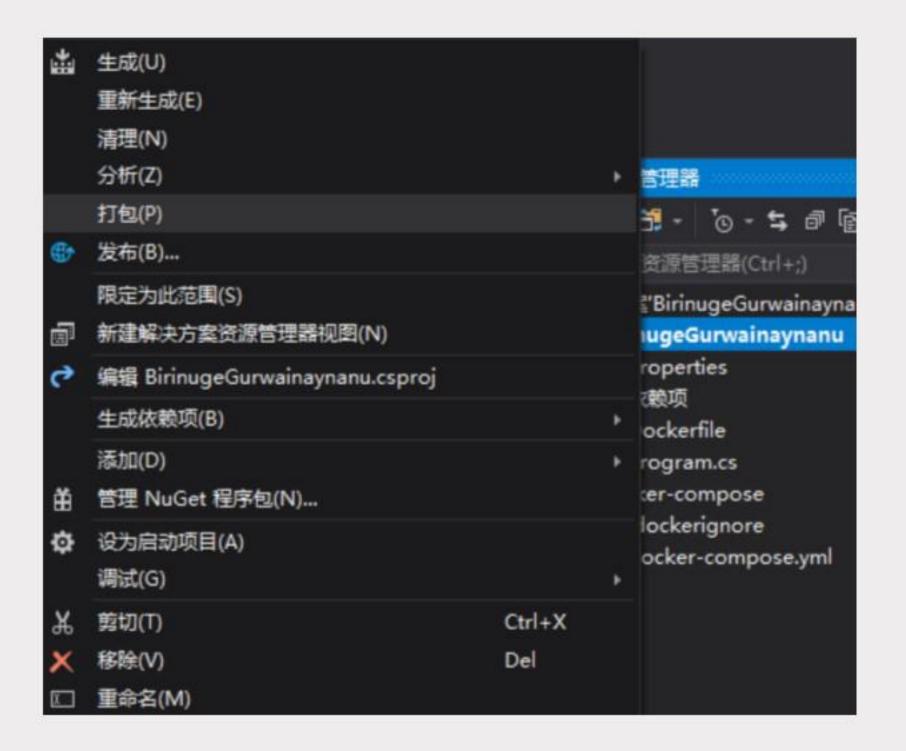
命令行可集成CI

支持私有 editor config 格式

支持 VSC 格式化



更靠近社区开发的功能





右击一键打包 NuGet

许可证支持表达式或自定义的文件



□ 在构建时生成 NuGet 包(G)	
□ 要求接受许可证(L)	
包 ID (I):	BirinugeGurwainaynanu
包版本(V):	1.0.0
作者(A):	BirinugeGurwainaynanu
公司(Y):	BirinugeGurwainaynanu
产品(P):	BirinugeGurwainaynanu
	^
说明(D):	
版权(C):	
许可(L):	
表达式(E):	● MIT
文件(F):	O (2.5



开发效率

优化自动格式化和重构



自动格式化和重构

- 让命名空间和文件所在文件夹保持一致
- 对类里面成员抽出接口
- 换行对齐参数
- 将匿名对象转换为 Value Tuple
- 转换代码为 Lambda 代码
- 反转if判断
- 自动使用 C# 8.0 的 range 功能
- 自动替换@\$"为\$@"
- 虽然还有很多功能,但是 Resharper 都有

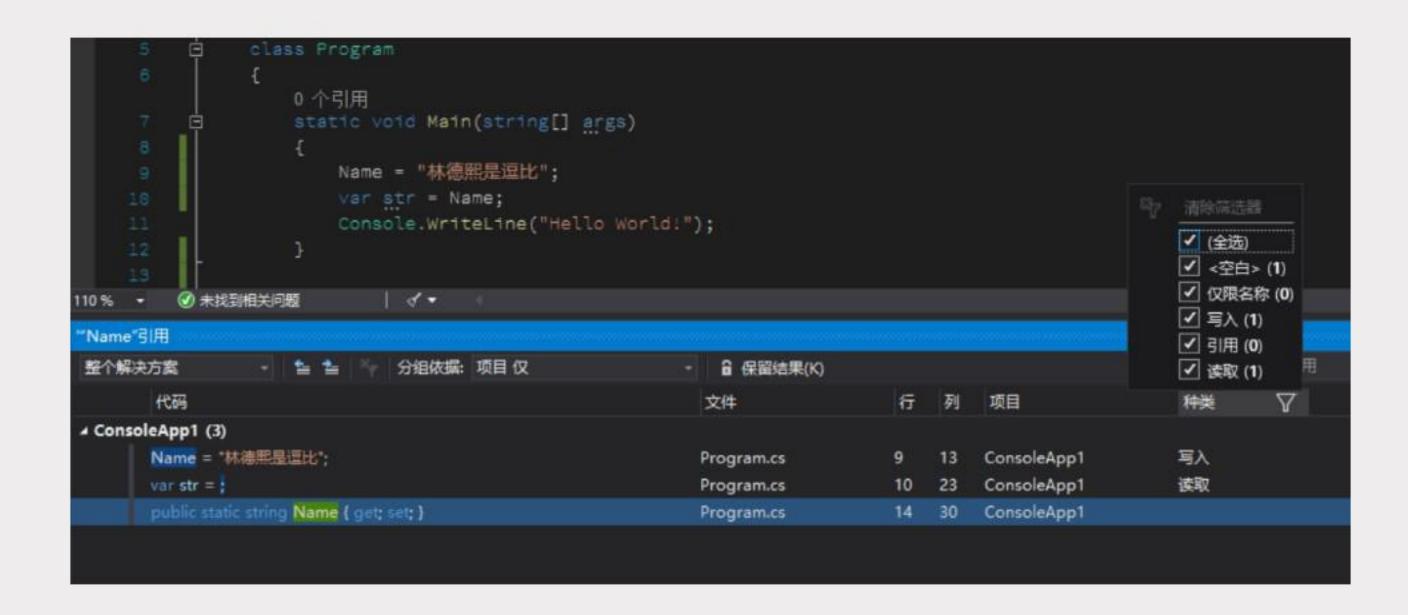
```
if (food.IsHealthy && scheduledTime >= DateTime.UtcNow)
    return true;
return false;
```



```
0 references
public void EatBreakfast()
    var banana = new Banana();
0 references
public bool Eat(IFood food, DateTime scheduledTime)
```

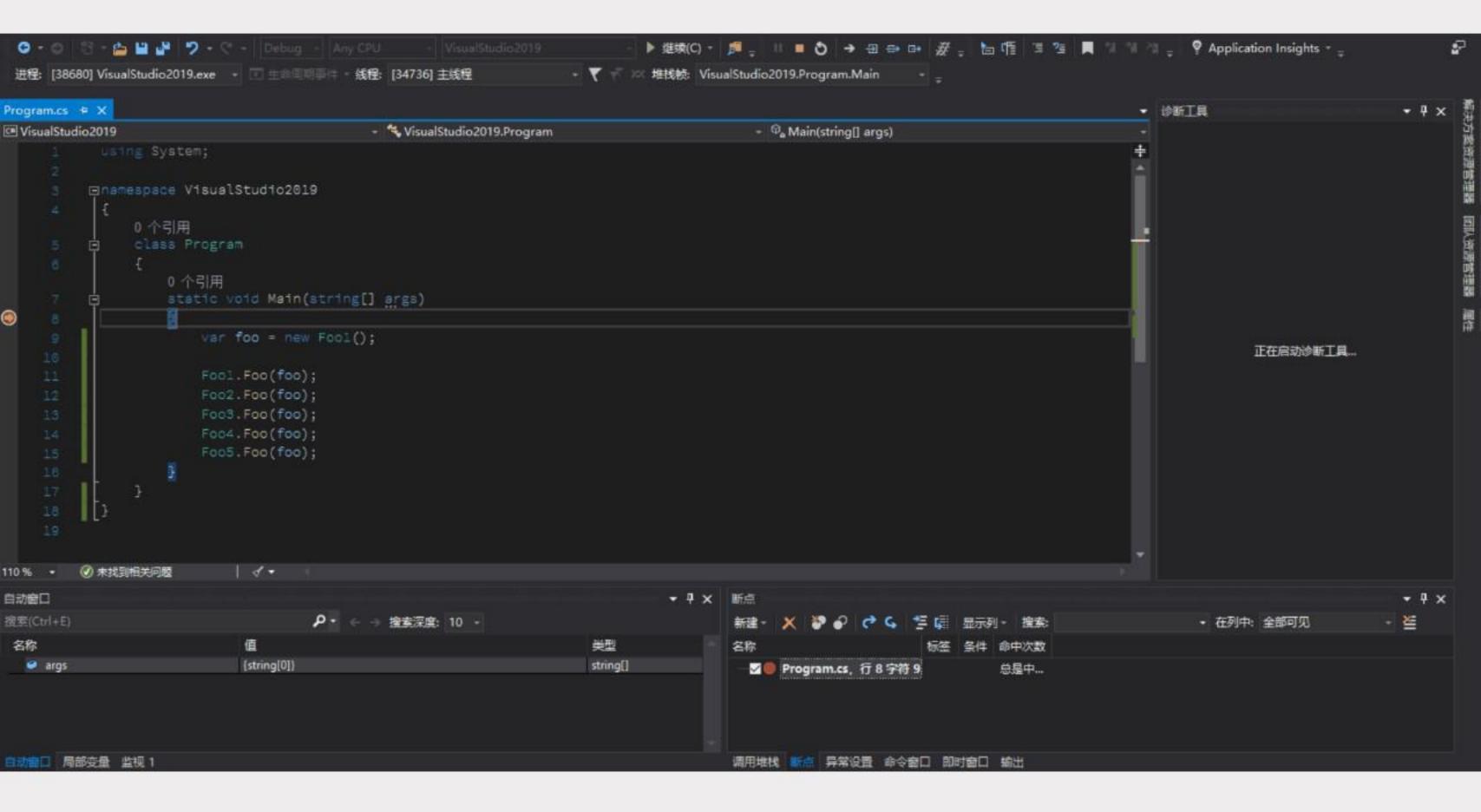
```
0 个引用
   0 个引用
   public IEnumerable<string> Foo()
       foreach (var temp in new[] { "广州", ".NET", "俱乐部" })
           if (temp.Length > 3)
               yield return temp;
       yield break;
```

过滤引用



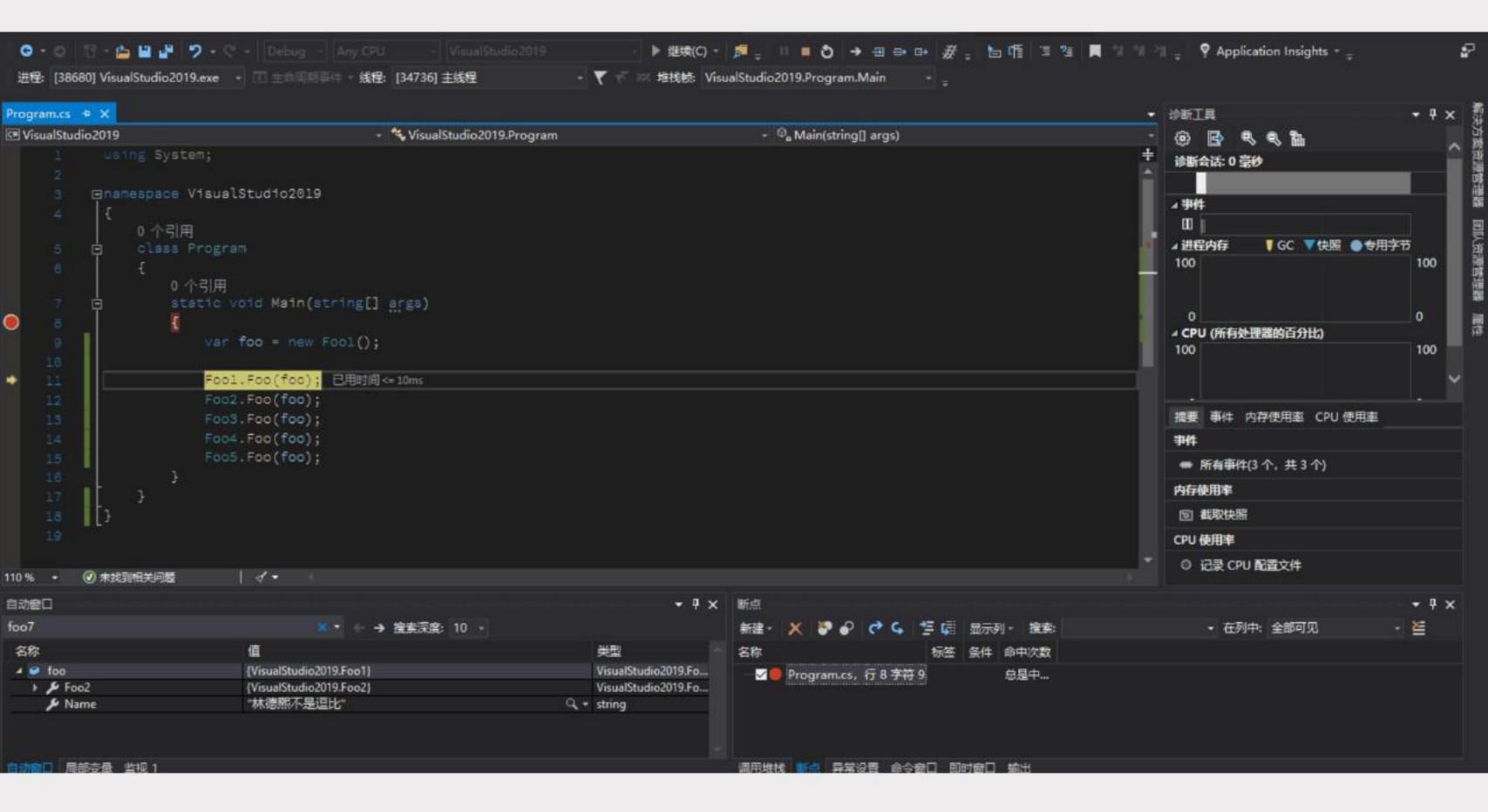


调试工具 💢

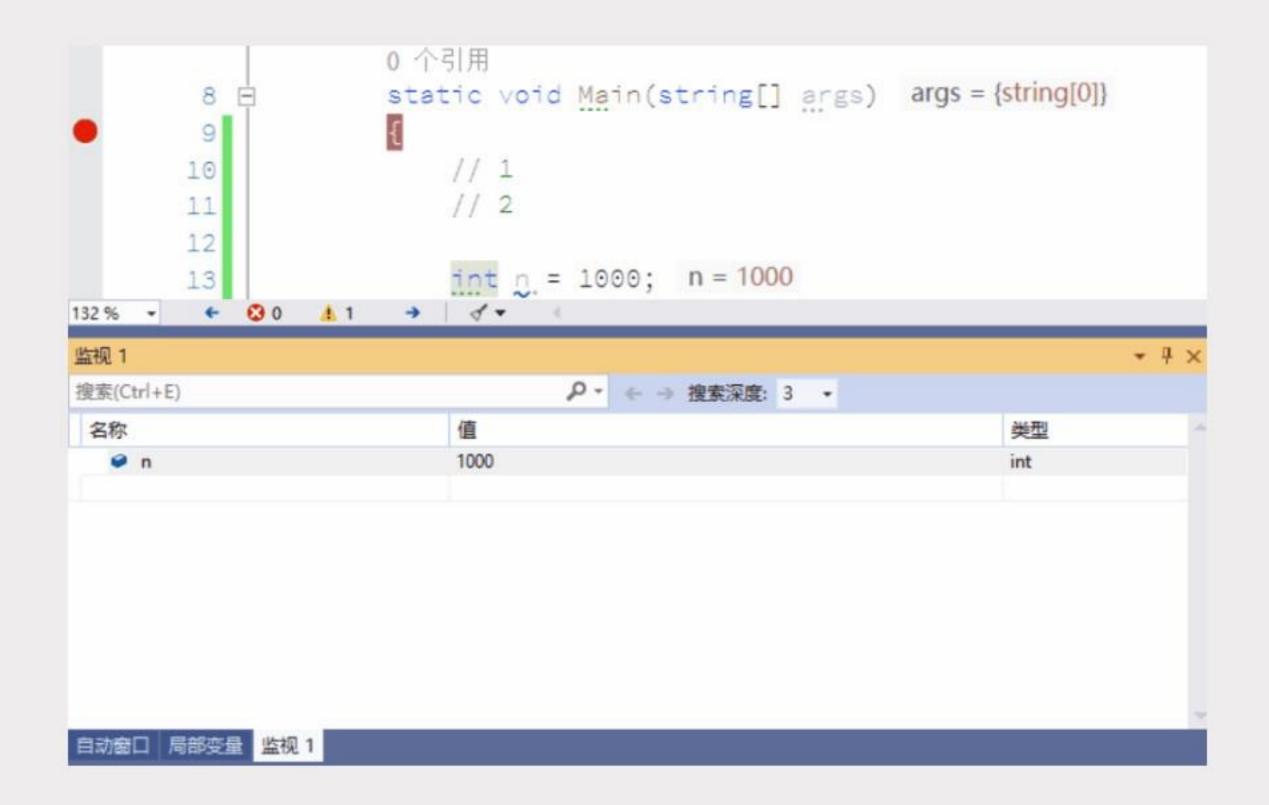




值更改断点 💢



格式化显示内容提示





性能调试工具

细分的功能







启动项目

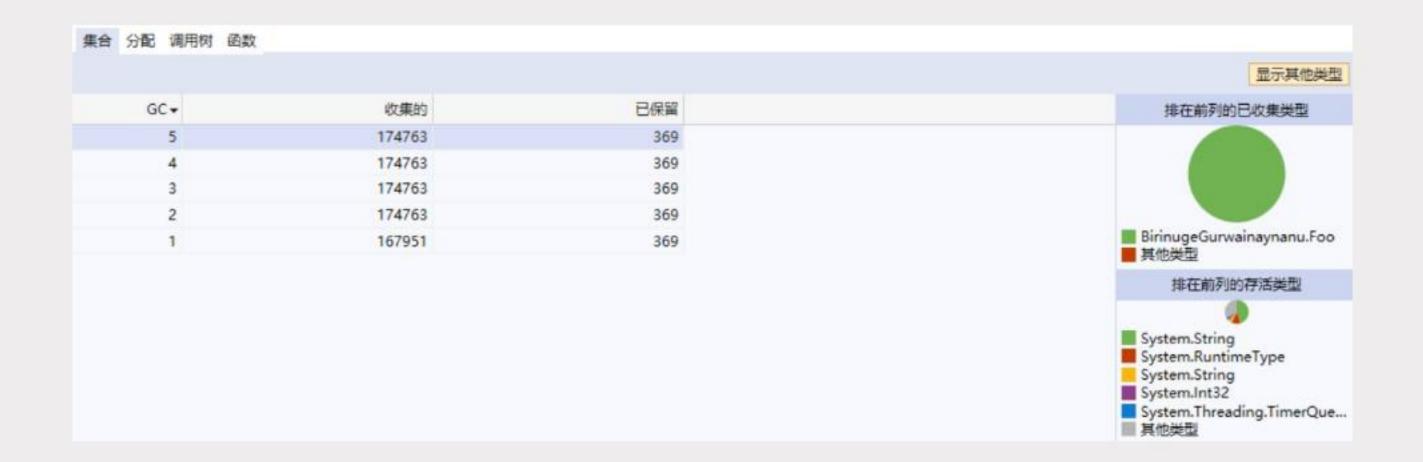
BirinugeGurwainaynanu

可用工具

行故障排除时很有用

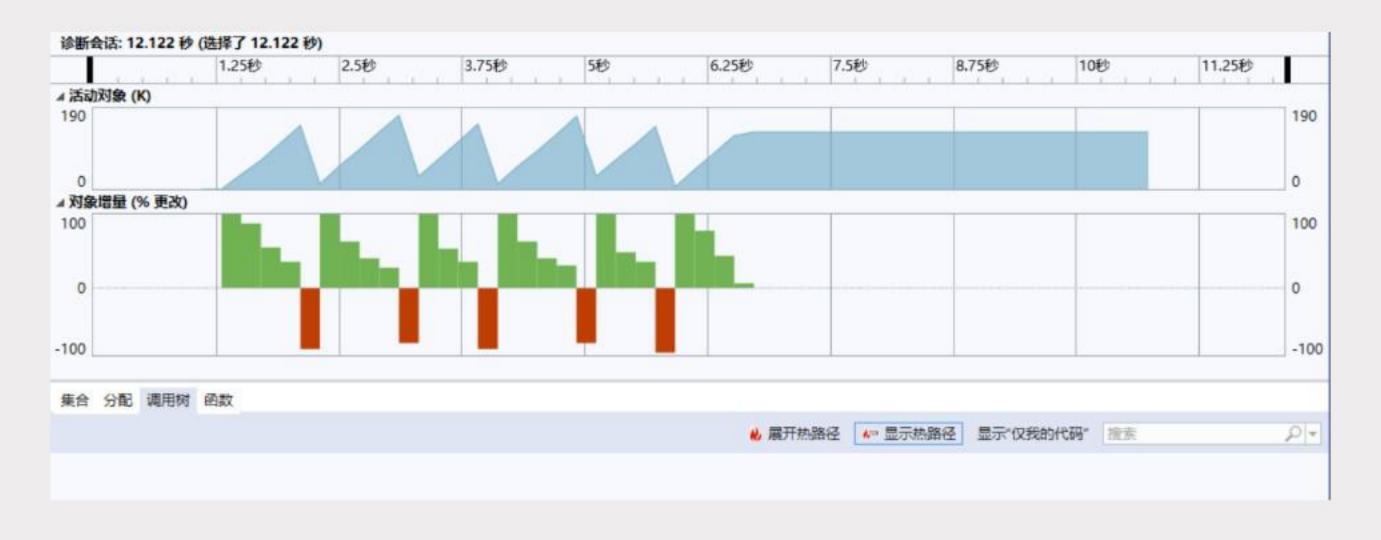
☑ .NET 对象分配跟踪 Φ
 查看 .NET 对象分配的位置以及 GC 收回它们的时间。
 查看 CPU 执行代码时的时间耗费情况。当 CPU 遇到性能瓶颈时很有用
 检测
 检测应用程序以调查确切的调用数和调用时间
 应用程序时间线
 检查应用程序中所用时间的分布情况。这对像低帧率之类的问题进





一键显示热路径







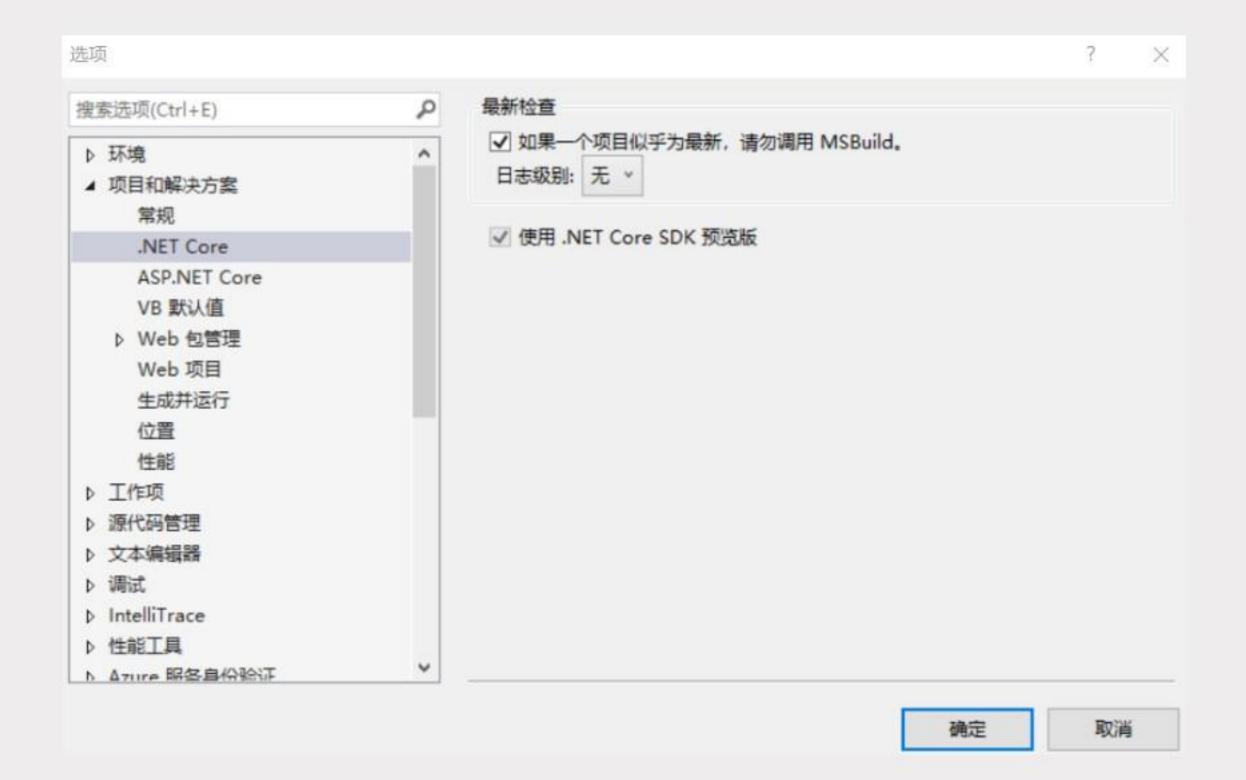
		₩ 展开热路径	₩ 显示热路径 显示"仅我的代码"	搜索	- ۵
函数名称	总计(分配)▼	自身(分配)	模块名		
a [40] corecinal	155	U	corecir.aii		
■ System.Private.CoreLib.dlll0x007ffc181	132	0	System.Private.CoreLib.dll		
■ System.Private.CoreLib.dll!0x007ffc1	129	0	System.Private.CoreLib.dll		
System.Private.CoreLib.dllI0x007ffc	114	0	System.Private.CoreLib.dll		
■ System.Private.CoreLib.dll!0x007	113	0	System.Private.CoreLib.dll		
▲ [本机] coreclr.dll	113	0	corectr.dll		
	113	0	System.Private.CoreLib.dll		
▲ [本机] coreclr.dll	113	0	corectr.dll		
System.Private.CoreLi	112	0	System.Private.CoreLib.dll		
▲ System.Private.Core	86	0	System.Private,CoreLib.dll		
▲ System.Private.Co	55	0	System.Private.CoreLib.dll		
▲ System.Private	55	0	System.Private.CoreLib.dll		
D System.Privat	30	0	System.Private.CoreLib.dll		
D System.Privat	25	0	System.Private.CoreLib.dll		

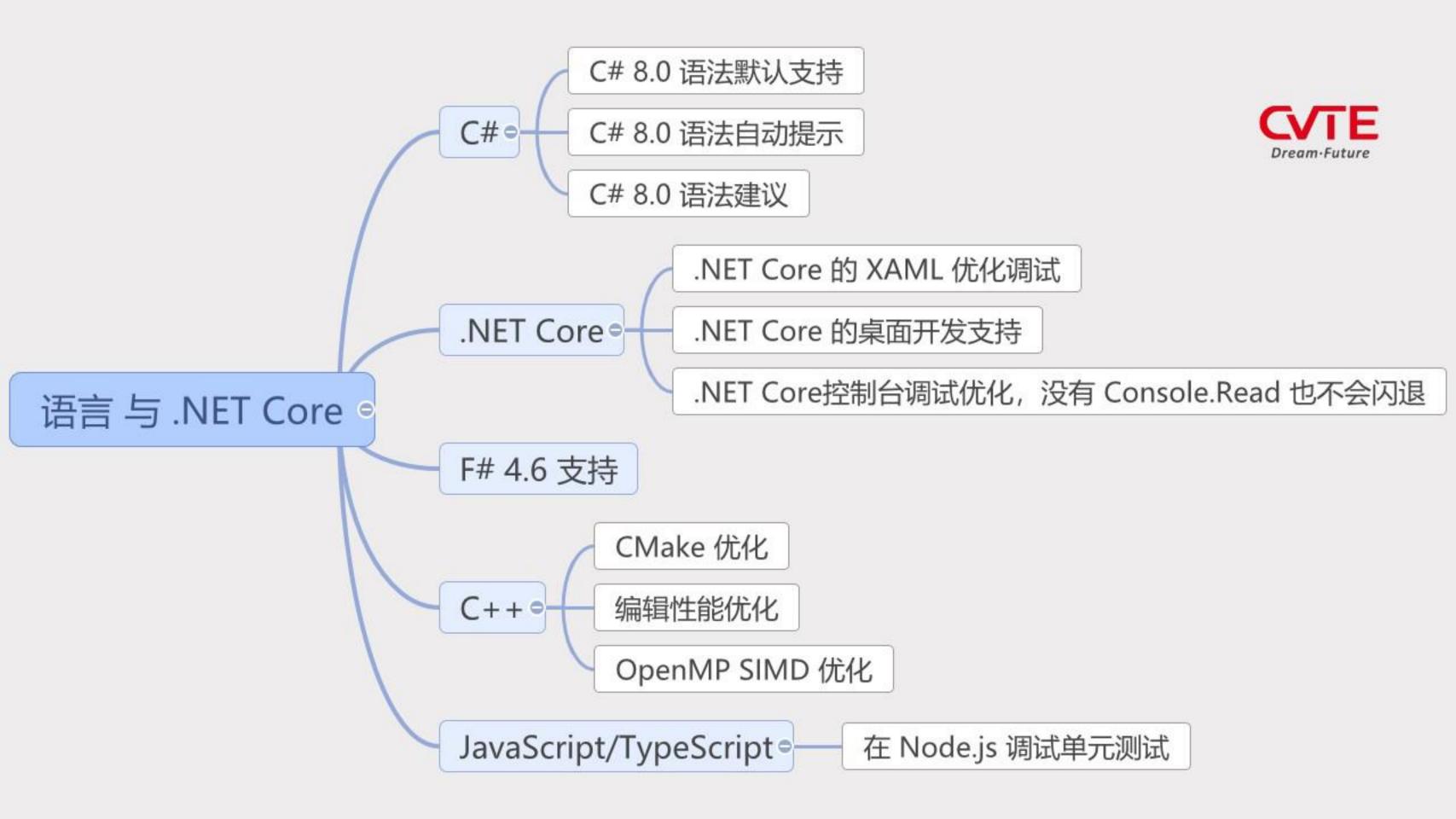


更多平台支持优化

默认使用预览版(如果是预览版的VS)









C# 8.0





switch

switch expressions



```
public enum Rainbow
{
    Red,
    Orange,
    Yellow,
    Green,
    Blue,
}
```

```
public static string FromRainbow(Rainbow colorBand) =>
            colorBand switch
            Rainbow. Red => "红",
           Rainbow. Orange => "橙",
           Rainbow. Yellow => "黄",
            Rainbow. Green => "绿",
            Rainbow. Blue => "蓝",
                           => throw new
ArgumentException (message: "invalid enum value", paramName:
nameof(colorBand)),
```





property pattern



```
internal class Foo
    public string F { get; set; }
private static string LeafalljurearjehuNerawaljeeyinaryem(Foo foo)
   var str = foo switch
       { F: "林德熙" } => "林德熙是逗比",
       { F: "逗比" } => "逗比",
  return str;
```

Tuple patterns

```
private static string Foo(string first, string second)
   var str = (first, second) switch
        ("lin", "dexi") => "林德熙是逗比",
       (, "dexi") => "没错, 这就是逗比",
       (,) => "不认识",
   return str;
   Console. WriteLine (Foo ("lin", "dexi"));
   Console. WriteLine(Foo("逗比", "dexi"));
   Console. WriteLine (Foo (null, null));
    没错,这就是逗比
    不认识
```





Deconstruct

逐渐 C# 和 .NET 分离



```
public class Point
    public int X { get; }
    public int Y { get; }
   public Point(int x, int y) => (X, Y) = (x, y);
   public void Deconstruct(out int x, out int y) =>
        (x, y) = (X, Y);
private static string Foo (Point point)
   var(x, y) = point;
   return $" {x}, {y}";
```



Positional patterns

```
private static string Foo(Point point) =>
   point switch
    var (x, y) when x > 0 && y > 0 => "两个值都大于零",
    var (x, y) when x < 0 && y < 0 => "两个值都小于零",
   var (, ) => "不合法的值",
     => throw new ArgumentException("point"),
    Console. WriteLine (Foo (new Point (1, 1)));
    Console. WriteLine (Foo (new Point (1, 0)));
    Console. WriteLine (Foo (null));
```

两个值都大于零

System.ArgumentException:"point"

不合法的值





using declarations



```
static void WriteLinesToFile(string str)
{
   using var file = new System. IO. StreamWriter("lindexi. txt");
   file. WriteLine(str);
   // file 会在这里释放
}
```



Static local functions



```
private static int Foo()
   int y = 5;
   int x = 7;
   return Add(x, y);
   static int Add(int left, int right) => left + right;
private static int Foo()
   int y = 5;
   int x = 7;
                                                  静态本地函数不能包含对
   return Add(x, y);
                                                       "x"的引用
   static int Add(int left, int right) => x + right;
```



Disposable ref structs

```
public ref struct Point
  public int X { get; }
  public int Y { get; }
  public Point(int x, int y) => (X, Y) = (x, y);
  public void Dispose()
      private static void Foo()
          Point point = new Point();
          using (point)
```



C# 和 .NET 关系不紧密



Asynchronous streams



```
private static async void Foo()
            await foreach (var number in GenerateSequence())
                Console. WriteLine (number);
       private static async System. Collections. Generic. IAsyncEnumerable (int)
GenerateSequence()
            for (int i = 0; i < 20; i++)
                await Task. Delay (100);
                yield return i;
```



Indices and ranges



Span<int>

int[]

```
var n = new[]
   0, 1, 2, 3, 4, 5, 6, 7, 8, 9
// 倒数第1个
var foo = n[^1]; // 9
// 倒数第5个到倒数第1个
var f1 = n[\hat{5}..\hat{1}];
// 从第3个开始到最后一个
var f2 = n[3..];
// 从第3个到第6个
var f3 = n[3..6];
// 获取全部
```

var all = n[..];



Range range = 3..;

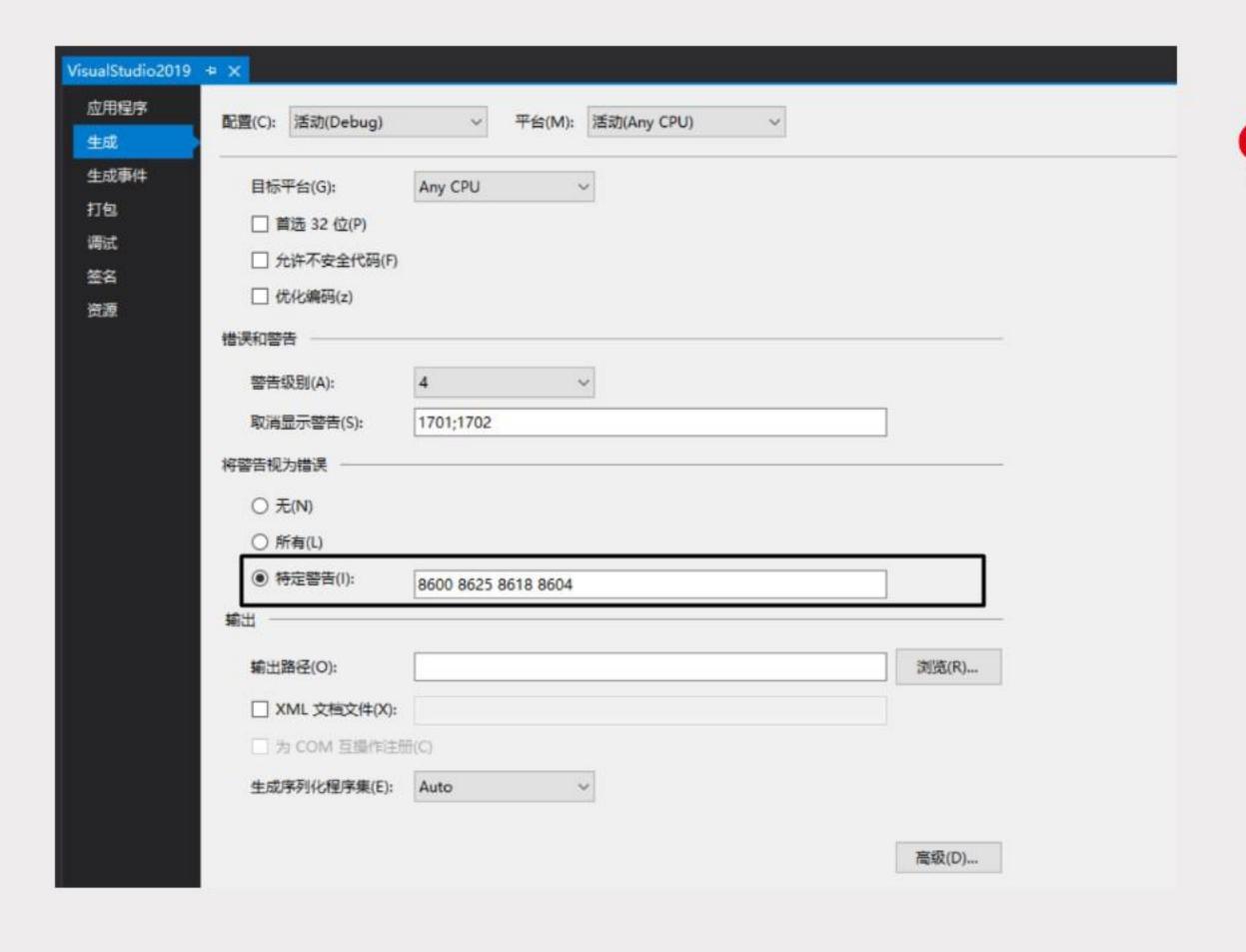
```
range = ^1..^3;
        var intCollection = new IntCollection();
        foreach(var temp in intCollection[range])
2 个引用
public class IntCollection
    1 个引用
    public IntCollection()...
    1 个引用
    public System.Collections.Generic.IEnumerable<int> this[System.Range range]...
    private int[] _intCollection;
```



Nullable reference type



所有值不可空,除非显式指定







言版本(L):	C# 8.0 (beta)	· -
部编译器错误报告(1):	提示	~
检查算术溢出/下溢(K	9	
H		
试信息(E):	可移植	~
件对齐(F):	512	~
基址(B):	0x00400000	



很多不需要关心, 但很有用的功能



版本: 16。 发行说明

比较版本 如何离线安装

Visual Studio 2019

适用于 Android、iOS、Windows、Web 和云的功能完备型集成开发环境 (IDE)

社区

功能强大的 IDE, 免费 供学生、开放源代码参 与者和个人使用

免费下载 🕹

下载预览版 >

Professional

最适合小型团队的专业IDE

企业版

适用于任何规模团队的可缩放端到端解决方案

免费试用 🕹

下载预览版 >

免费试用 🕹

下载预览版 >





VisualStudio 2019 新特性