

# C#程序设计

计算机学院计算机科学系 杨鸣

计算机学院办公楼631

[ymzhongzhong@qq.com](mailto:ymzhongzhong@qq.com)

QQ群：642173754

- 学时40： 课堂28+上机12
- 教材： 《C#高级编程》， 清华大学出版社
- 参考书： 《WPF编程宝典》 《深入浅出WPF》
- 课程成绩：
  - 平时听课和作业： 40%
  - 最终课程考核： 程序验收30% + 结课报告30%
  - 有以下情况之一者， 不及格
    - 无故旷课4次及以上（迟到2次=旷课1次）
    - 不参加程序验收
    - 不交结课报告

# 什么是编程语言

- 用于编写程序的语言

- 语言：字  $\xrightarrow{\text{规则}}$  词  $\xrightarrow{\text{规则}}$  句  $\xrightarrow{\text{规则}}$  段  $\xrightarrow{\text{规则}}$  文

- 编程语言：

- 字：int

- 词：int \*

- 句：int \* a;

- 段：int a=0; for(i=0; i<10; i++) a += i; 或

- int add()

- {

- int i,a=0; for(i=0; i<10; i++) a += i;

- return a;

- }

- 文: 

```
void main()
{
    int i,a; for(i=0; i<10; i++) a += i;
    printf("%d", a);
}
```

或

```
extern int add();
void main()
{
    int a;
    a = add();
    printf("%d", a);
}
```

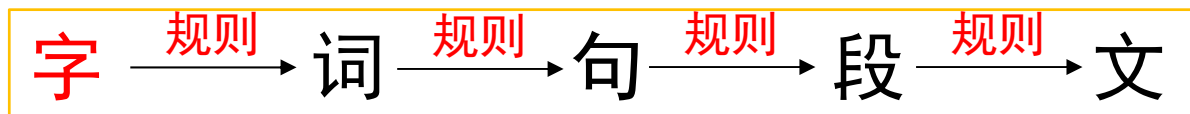
```
int add()
{
    int i,a=0;
    for(i=0; i<10; i++)
        a += i;
    return a;
}
```

- 文: 

```
void main()
{
    int i,a; for(i=0; i<10; i++) a += i;
    printf("%d", a);
}
```

或

<pre>extern int add(); void main() {     int a;     a = add();     printf("%d", a); }</pre>	<pre>int add() {     int i,a=0;     for(i=0; i&lt;10; i++)         a += i;     return a; }</pre>
---	--



# 课程定位

武林高手：手中铁剑换成树枝，亦能击败对手。手中武器只是发挥内在实力的工具和载体。

- 承上

- 《C语言程序设计》、《面向对象程序设计》等课程
- 查漏补缺、完善知识体系

- 启下

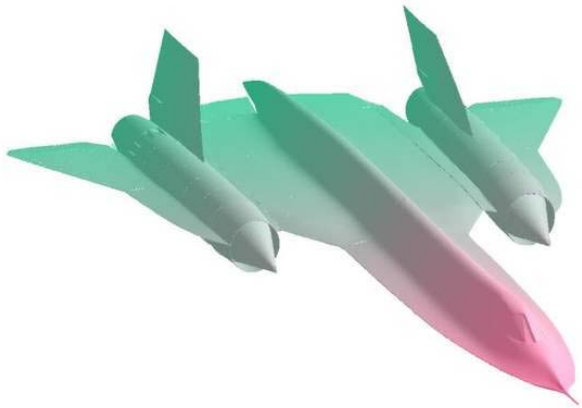
- 接口、设计模式、泛型、正则表达式、WPF可视化程序设计等新的编程技术和方法

以C#语言为载体，介绍高级编程技术和方法，提高程序开发能力。

- C#为载体？

- 面向对象的、现代的编程语言，可移植
- C#开发程序效率高，编程人员精力集中程序设计（而不是程序实现）
- C#应用广泛，比如Truffer、Unity

哈工大禁用MATLAB → 国产软件Truffer

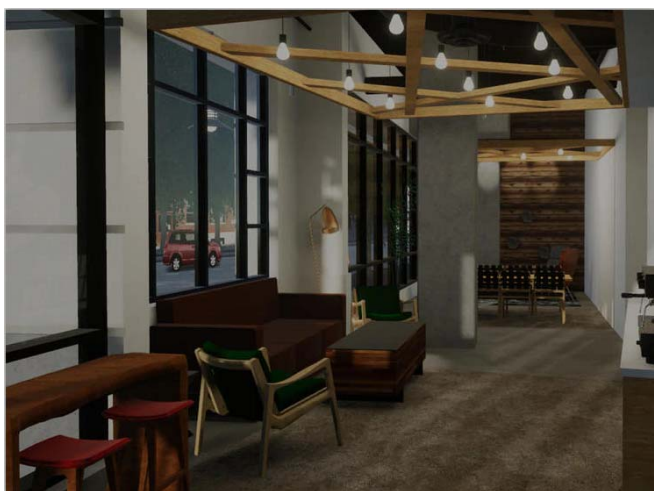


C#代码

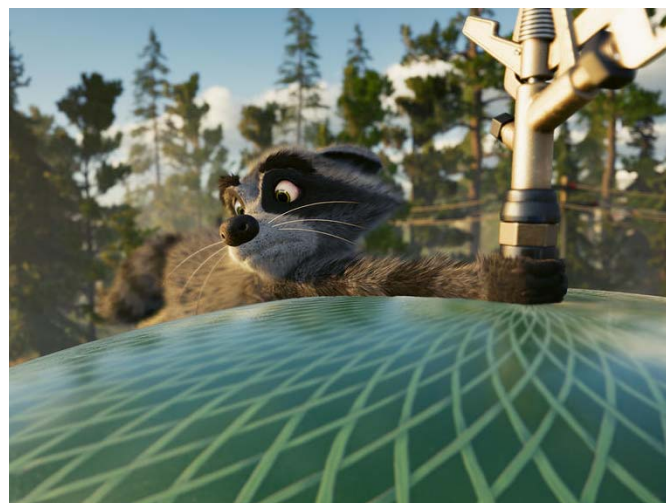
# Unity脚本：C#语言



游戏开发



建筑设计



电影、动画、影视



# 第1章 C#与NET平台介绍

- 了解.NET Framework 的结构
- 理解.NET Framework 的基本概念
  - CLR
  - JIT
  - CTS
  - MSIL
- 了解 .NET Framework 命名空间
- 了解C#程序的基本结构

# 1 C#语言的由来

- 与VB等语言相比，同等级别的C/C++应用程序往往需要更长时间来开发。
- VB以牺牲灵活性的代价来提高效率，但对编程人员的限制过多（如屏蔽一些底层代码控制的机制），无法方便地进行系统交互。
- 对于C/C++用户来说，最理想的解决方案无疑是在快速开发的同时又可以调用底层平台的所有功能和现有的类库高效地进行程序开发。
- C# (C sharp) 是对这一问题的解决方案。C#是一种新的、面向对象的、可移植的编程语言。

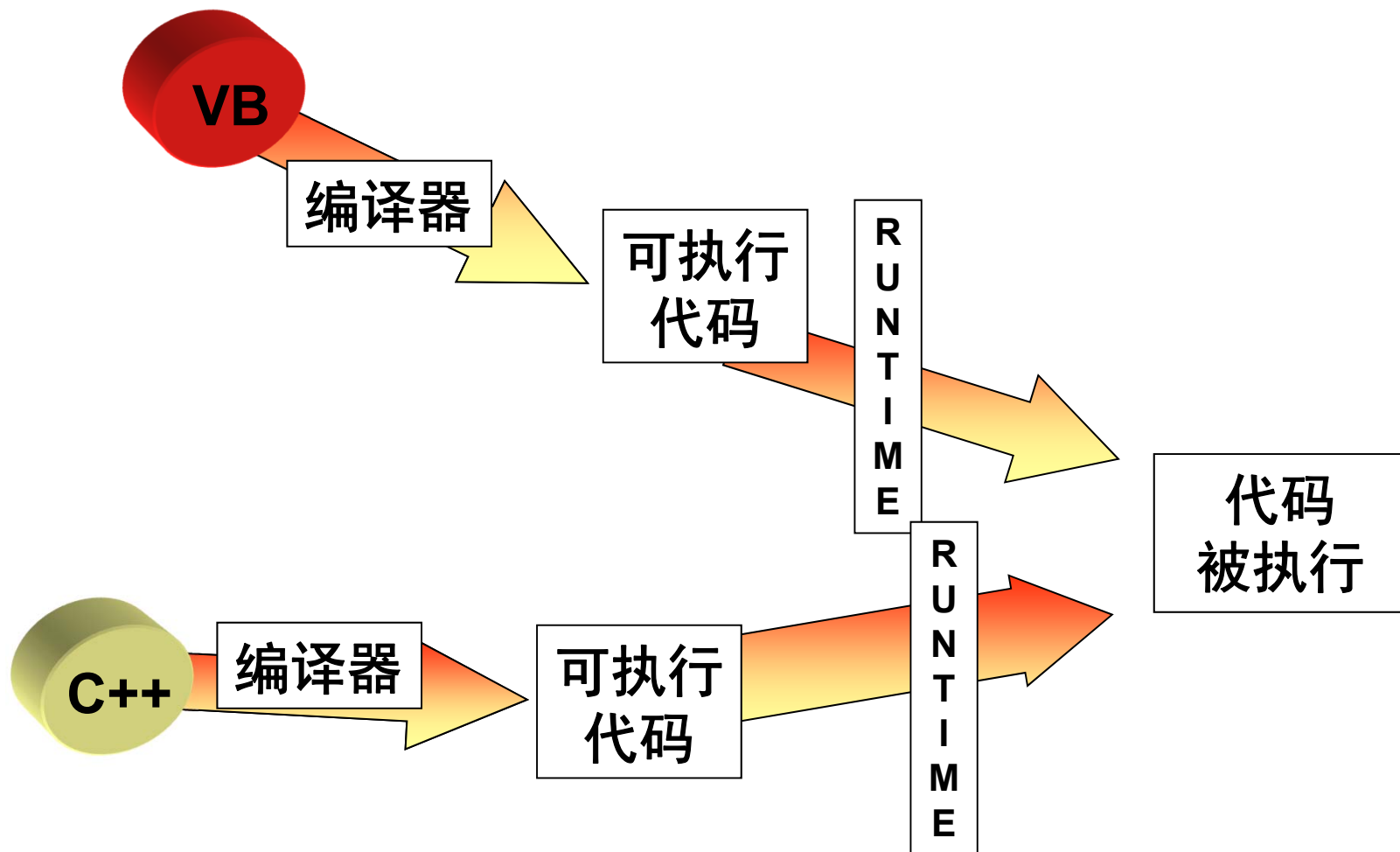
# • 移植性语言





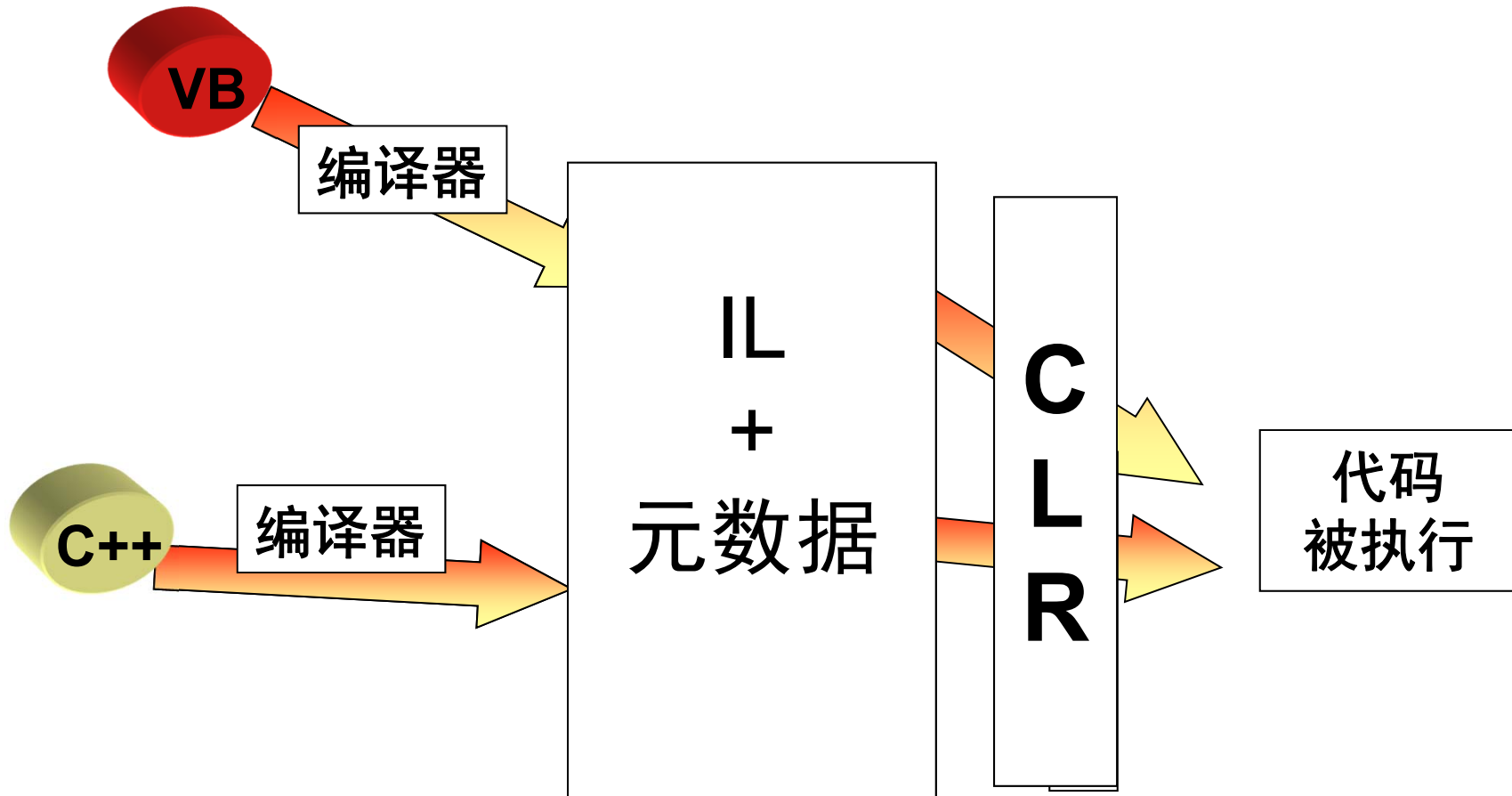
## 2 .NET代码的编译

### 语言特定的编译器和运行库

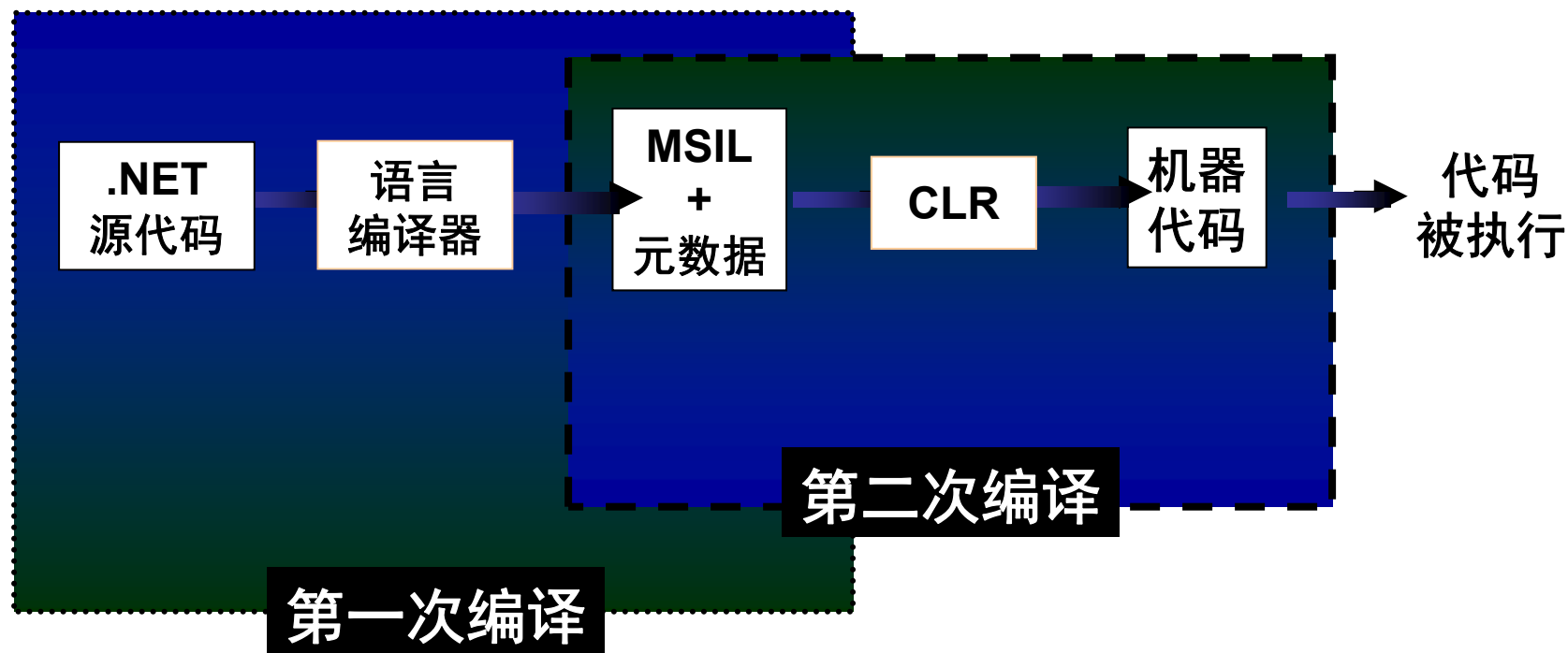


# IL 和 CLR 的使用

Intermediate Language  
Common Language Runtime

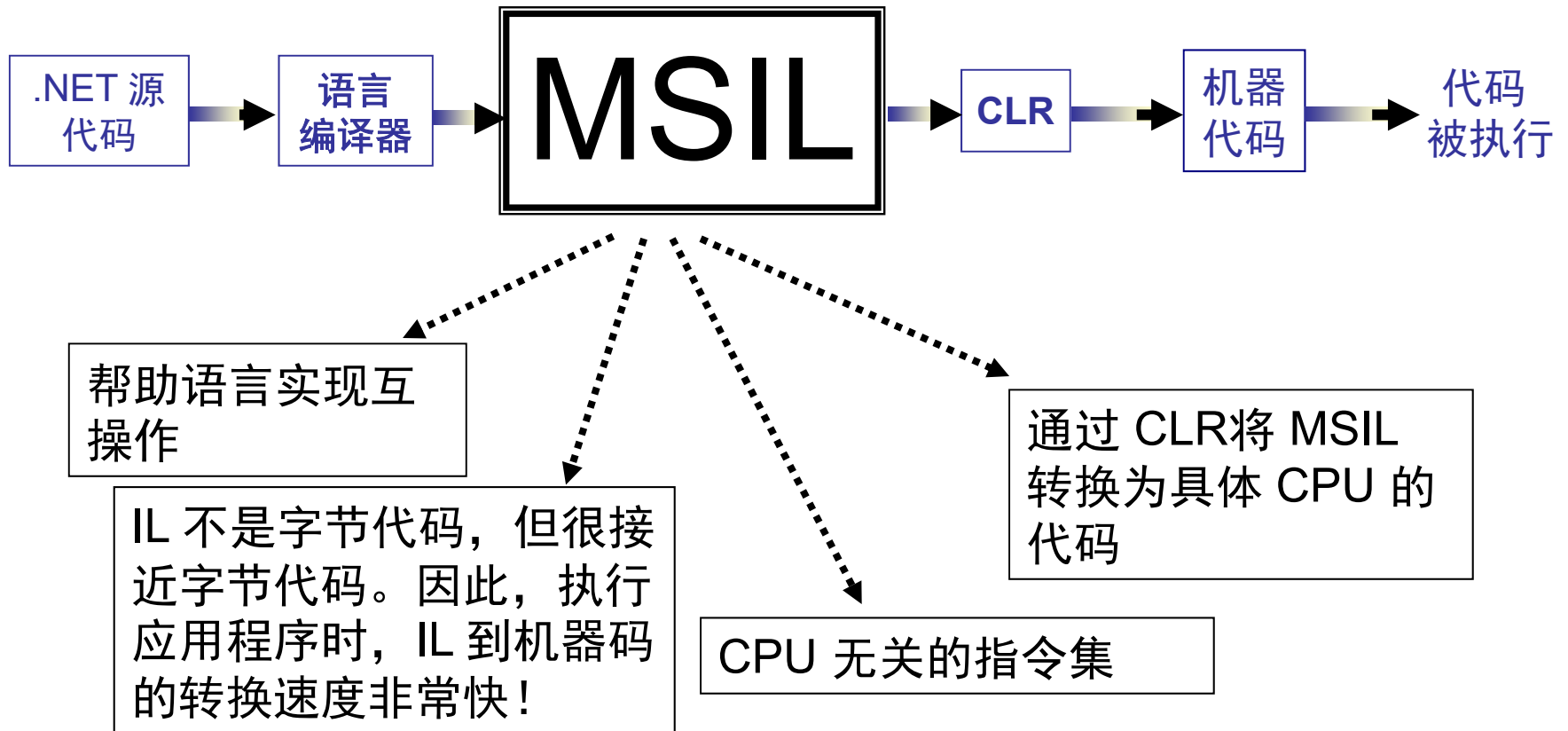


# 编译执行 .NET 程序



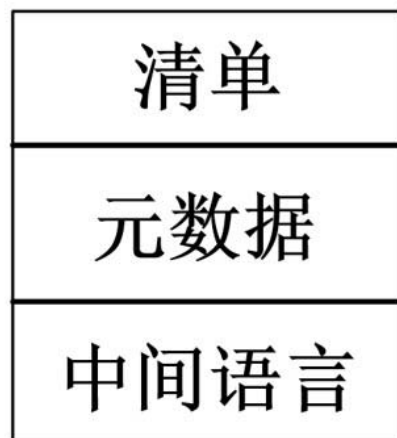
.NET 程序被编译两次，第一次编译较慢，而第二次编译较快！

# Microsoft 中间语言

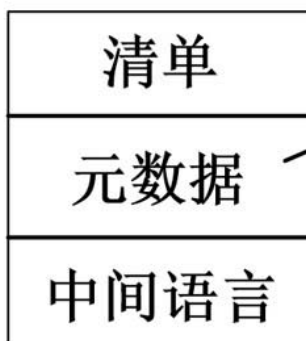




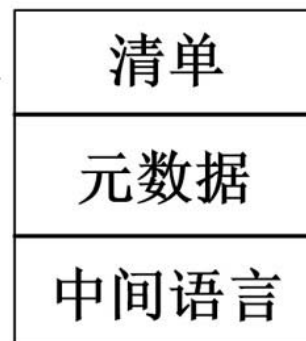
# 程序集



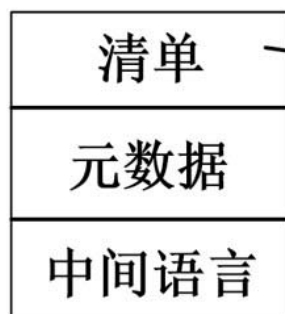
client.exe



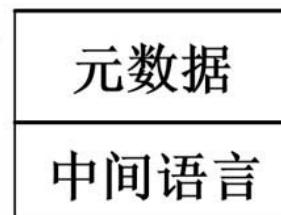
client.exe



clientlib.dll



client.exe

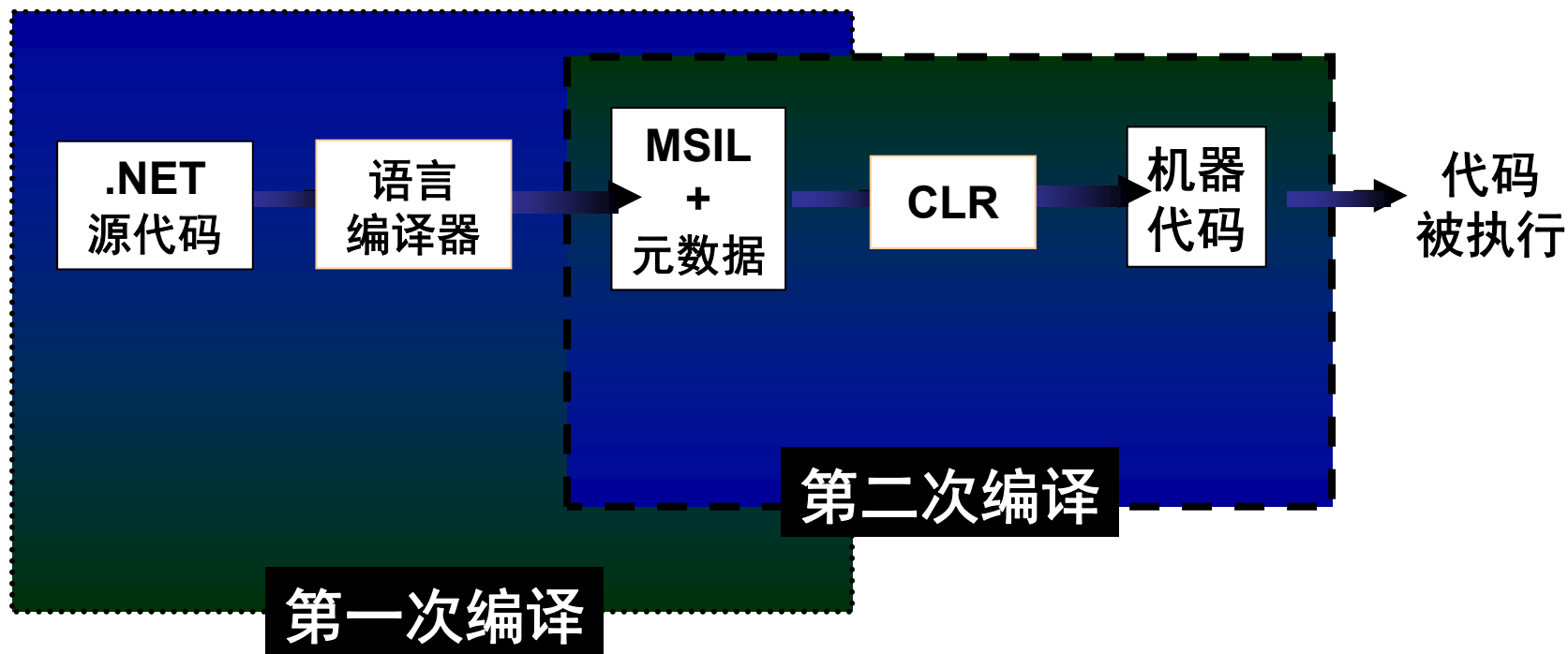


clientlib.netmodule

# IL语言范例

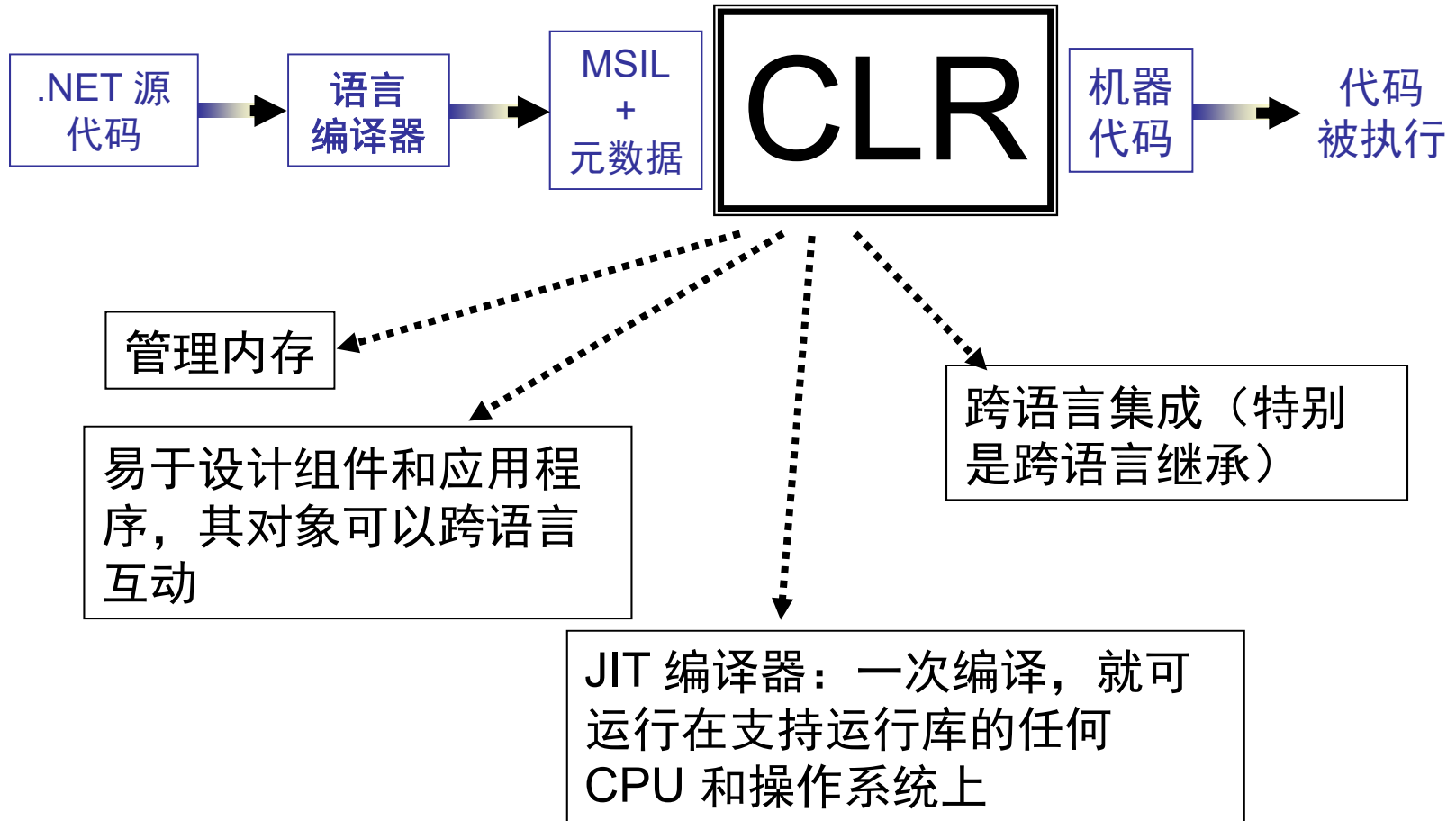
```
IL_0000: nop
IL_0001: call
IL_0006: nop
IL_0007: ldc.i4.0
IL_0008: call
IL_000d: nop
IL_000e: ldnull
IL_000f: ldftn
IL_0015: newobj
IL_001a: call
IL_001f: nop
IL_0020: newobj
IL_0025: call
IL_002a: nop
IL_002b: newobj
IL_0030: throw
```

# 编译执行 .NET 程序



.NET 程序被编译两次，第一次编译较慢，而第二次编译较快！

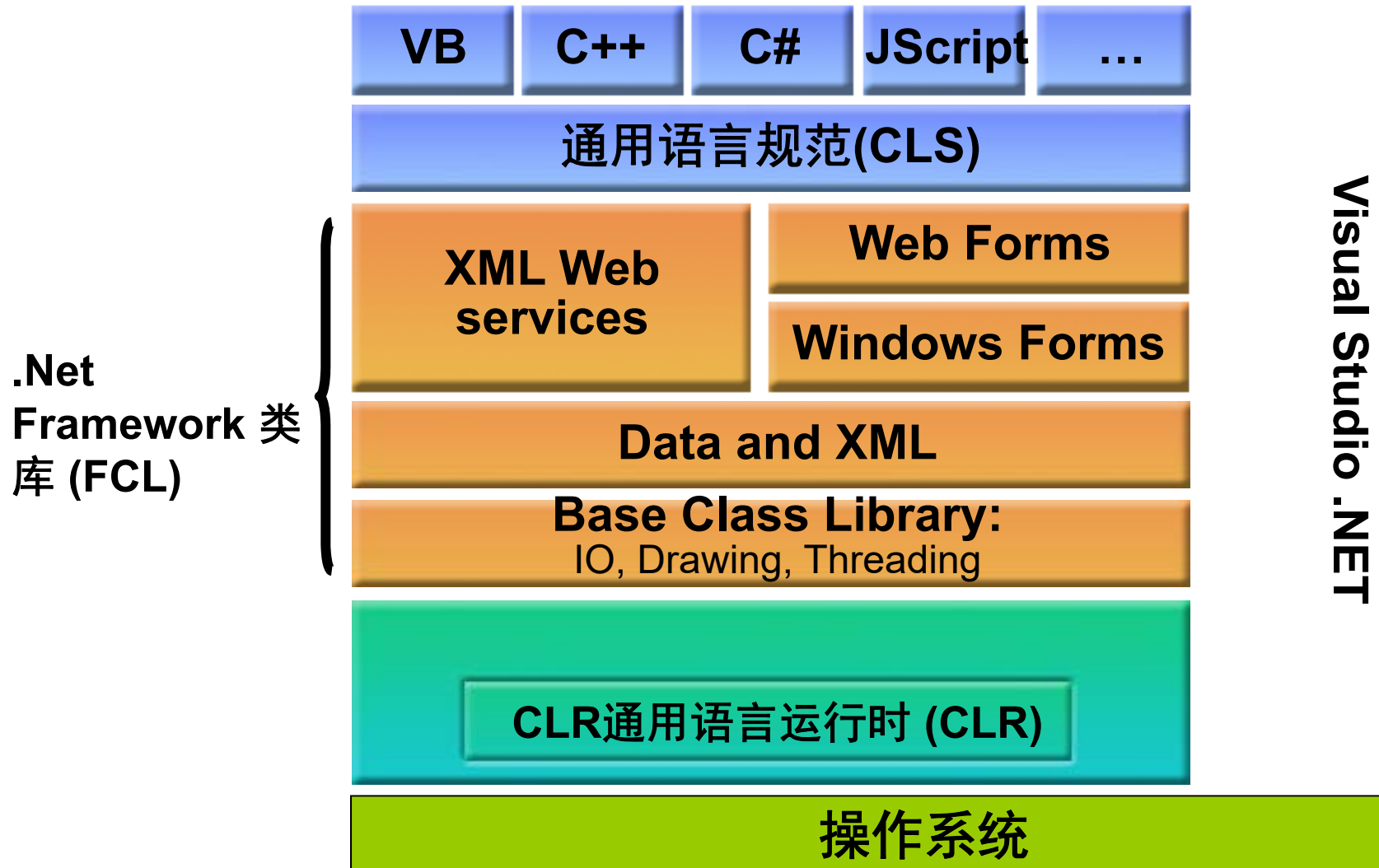
# 通用语言运行时CLR



# 3 .NET Framework 简介

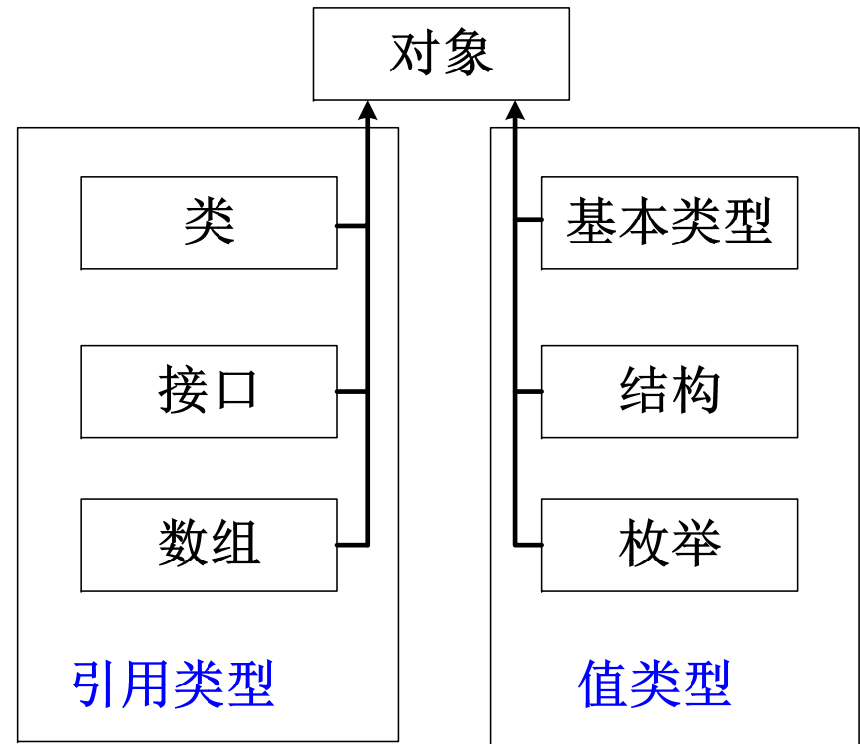
- .NET Framework类似于JVM
- .NET Framework 两个主要组件：
  - 通用语言运行时 (CLR)
  - 统一的类库集
- .NET的类库：
  - 线程
  - 文件输入/输出 (I/O)
  - 数据库支持
  - XML 解析
  - 数据结构.....

# .NET Framework 的体系结构



# CLS和CTS

- 通用语言规范 CLS  
Common Language Specification
  - 规定所有 .NET 语言都应遵循的规则
  - 生成可与其他语言互操作的应用程序
- 通用类型系统 (Common Type System, CTS)
  - 包含标准数据类型
  - 包含准则集



CLS、CTS 和 MSIL 紧密配合以实现语言互操作性

## 4 命名空间

- 命名空间就像在文件系统中一个文件夹容纳多个文件一样，可以看作某些类的一个容器。
- 定义命名空间：`namespace`，使用命名空间：`using`
- 类似`#include`



1.H

```
class A
{
    ...
}
```

3.cpp

```
#include "1.h"
#include "2.h"
....
```

2.H

```
class A
{
    ...
}
```

1.H

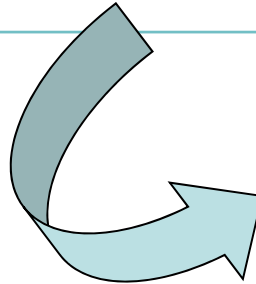
```
class A
{
    ...
}
```

2.H

```
class A
{
    ...
}
```

3.cpp

```
#include "1.h"
#include "2.h"
....
```



3.cpp

```
class A
{
    ...
}
class A
{
    ...
}
...
```

```
namespace a1
{
    class A
    {
        ...
    }
}
```

```
namespace a2
{
    class A
    {
        ...
    }
}
```

3.cs

using a1;

using a2;

a1.A

a2.A

# 常用命名空间

命名空间	说明
System.Drawing	处理图形和绘图，包括打印
System.Data	处理数据存取和管理，在定义 ADO.NET 技术中扮演重要角色
System.IO	管理对文件和流的同步和异步访问
System.Windows	处理基于窗体的窗口的创建
System.Reflection	包含从程序集读取元数据的类
System.Threading	包含用于多线程编程的类
System.Collections	包含定义各种对象集的接口和类

```
using System.Drawing;
public class myRect
{
    private Point p0;
    private int a, b;
    public myRect(int aa,int bb, Point p)
    {
        a = aa; b = bb; p0 = p;
    }
    public int Area() //求面积
    {Return (a * b);}
    public int round() //求周长
    {Return (2 * (a + b));}
    public void move(int dx, int dy) //移动顶点
    {
        p0.X += dx;
        p0.Y += dy;
    }
}
```

```
public class myRect
{
    private System.Drawing.Point p0;
    private int a, b;
    public myRect(int aa,int bb, System.Drawing.Point p)
    {
        a = aa; b = bb; p0 = p;
    }
    public int Area() //求面积
    {Return (a * b);}
    public int round() //求周长
    {Return (2 * (a + b));}
    public void move(int dx, int dy) //移动顶点
    {
        p0.X += dx;
        p0.Y += dy;
    }
}
```

# 5 创建 “Hello World”应用程序

- 单击 “开始” → “程序” → “Microsoft Visual Studio”，打开 VS.NET 的起始页。
- 要创建 C# 控制台应用程序，请从列表中选择 “创建建” → “项目”。此时将打开 “新建项目” 窗口。
- 选择 “Visual C# 项目” 作为项目类型，选择 “控制台应用程序” 作为模板。

# C# 模板

```
using System;
using System.Collections.Generic;
using System.Text;

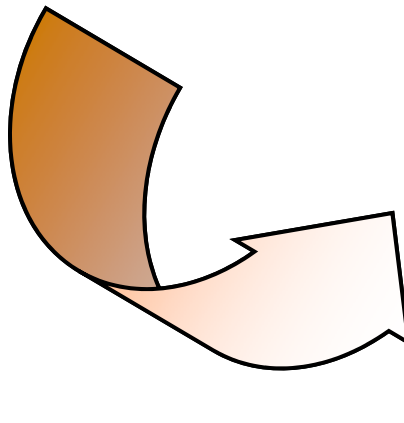
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

输出



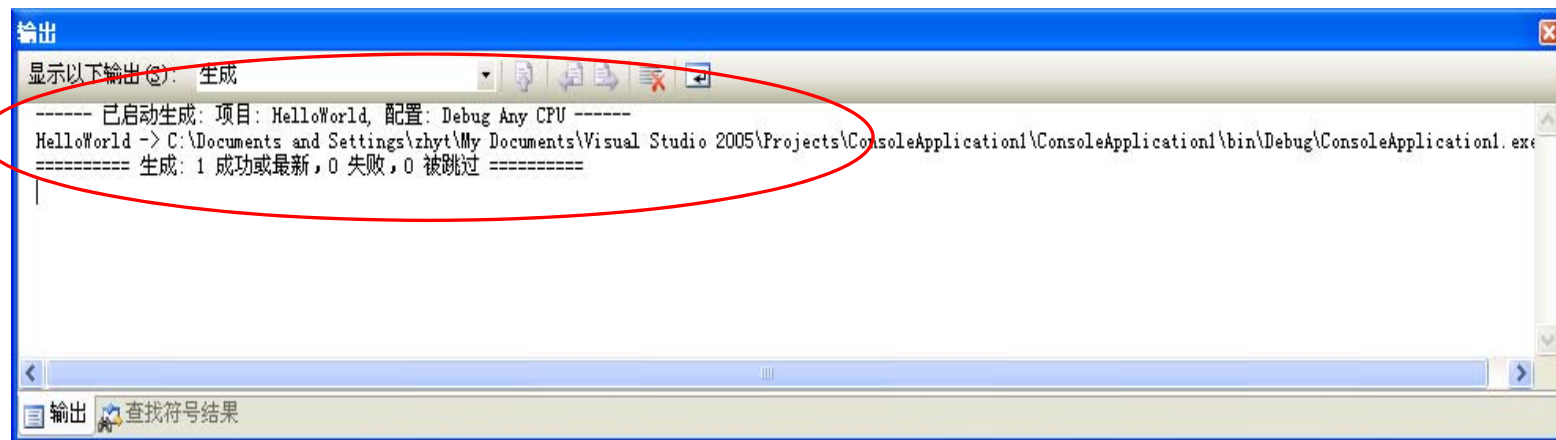
## “Hello World”应用程序文件

- HelloWorld.csproj
- AssemblyInfo.cs
- Program.cs

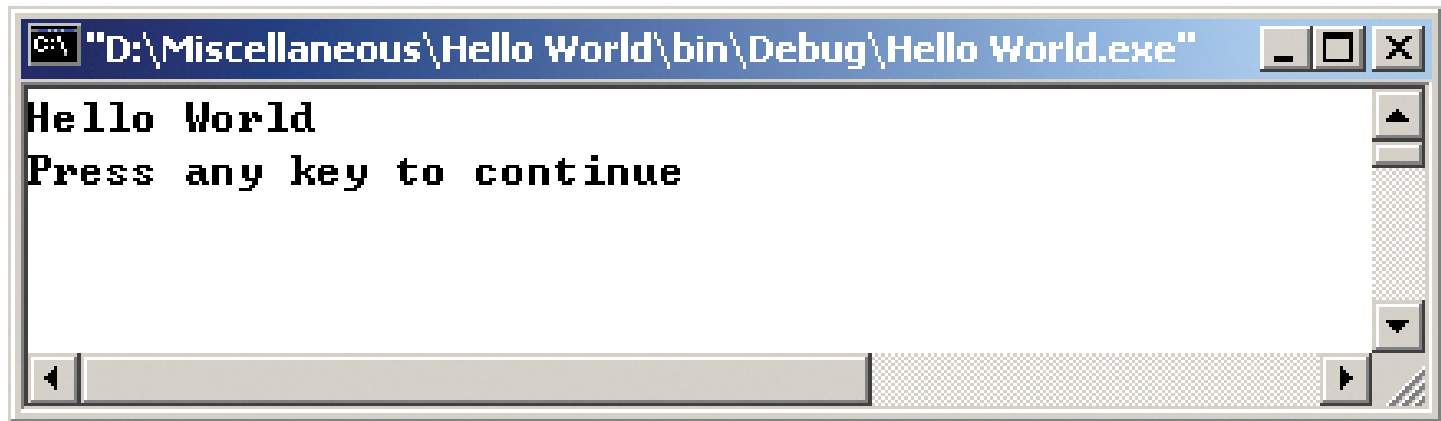


# 编译和执行 C# 应用程序

- 要生成 C# 项目，请从菜单中选择“生成”→“生成解决方案”。
- 该过程将编译项目中包括的所有文件，编译结果显示在“输出”窗口中。
- 如果结果显示“生成: 1 成功或最新, 0 失败, 0 被跳过”，则说明已经成功生成，然后即可交付该应用程序。



- 要运行刚才生成的应用程序，请从菜单中选择“调试” → “开始执行(不调试)”
- 输出结果显示在 VS.NET 控制台窗口中



# Hello World 程序

//这是用 C# 编写的一个简单的 HelloWorld 程序 描述代码的注释

using System; 导入 System 命名空间

namespace Notepad 声明命名空间 Notepad

{  
    class HelloWorld 声明 HelloWorld 类

    {  
        static void Main() 程序入口点， Main 的返回类型为 void

        { 控制台类的 WriteLine() 方法用于显示输出结果

            Console.WriteLine("Hello World");

        }

    }

将文件保存为 \*.cs

}

## 说明:

1. 程序一般包括以下几部分：
  - 名称空间的引用：使用**using**关键字指明引用的名称空间。
  - 名称空间的声明：使用**namespace**关键字声明名称空间。
  - 类：使用**class**关键字声明类。
  - Main方法：**Main**方法是C# 程序的入口。
2. 类中包含了程序所要用到的数据和所要执行的方法的定义。  
每个类中可以有：
  - (1)变量声明
  - (2)构造函数
  - (3)方法
  - (4)属性
  - (5)事件
3. 一个文件中可以有1个或者多个类。
4. 所有语句都以分号";"结束。