不同点：

1. 类class
2. 访问修饰符：

VB：Public、Protected、Friend、Private

C#：Public、Protected、Internal、Private

1. 静态成员

VB：Shared Member

C#：Static Member

1. 属性成员

VB：使用Property关键字，set（Byval value as 类型），只传值：Public Property X() As 类型,具有用default使用的默认属性，可通过实例而不写属性名直接访问。

C#：只需花括号内写get和set函数，只传值：Public 类型 X {get; set;}

1. 构造函数

VB：Public Sub New（）

C#：Public 类名（）

1. 类中代指自己的实例关键字

VB：me

C#：this

注意，代指关键字是类的实例，只能找到实例属性和方法，在所有静态方法和属性中都不能使用代指关键字。

1. 继承

VB: 在子类中使用Inherent 父类

C#：在子类名后加：父类

1. 重写

VB：在父类中使用Overridable关键字，在子类中使用Overrides关键字

C#：在父类中使用virtual关键字，在子类中使用override关键字

1. 在子类中调用父类关键字、

VB：使用Mybase.

C#: 使用base.

1. 抽象类

VB：使用MustInherit在类名前，虚方法前使用MustOverride修饰

C#：使用abstruct在类名前，虚方法前使用abstruct修饰

1. 无法继承类

VB：使用NotIheritable关键字

C#：使用sealed关键字

1. 接口

VB: 在子类中使用Implements接口名

C#：在子类名后加：接口名

1. 委托

VB：Dim 委托名 As 委托类型=new（AddressOf 方法名）

或Dim 委托名 As 委托类型= AddressOf 方法名

C#：new 委托类型（方法名）

或 委托类型 委托名=方法名

1. 多播委托

VB：Dim 委托名 As 委托类型=System.Delegate.Combine(委托名1，委托名2)

Dim 委托名 As 委托类型=System.Delegate.Remove(委托名，委托名1)

C#：委托类型 委托名+=委托名1

委托类型 委托名-=委托名1

1. 事件

VB：事件的绑定 AddHandler Button.Click,AddressOf 处理方法

C#：事件的绑定 Button.Click+=处理方法

1. 泛型

VB：类名（Of T）

C#：类名<T>

1. 语句
2. 条件跳转语句

VB：Select Case 筛选器

Case 1

Case 2 To 5

Case else

C#：Switch （筛选器）

Case 1:

break;

Case 2:

Case 3:

Case 4:

Case 5:

break;

default:

break;