

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

# DIY Starter Kit 系列第二讲

## ➤ 电影收藏

俞晖

MSDN Online Program Manager  
dolphin@vip.163.com

# 日程

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- **Starter Kits**
- **Movie Collection Starter Kit**
- **扩展 Movie Collection Starter Kit**
- **建立自己的 Starter Kit**

# Starter Kits

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 什么是 Starter Kit?
  - Sample 样例的应用程序
  - 包含所有的源代码
  - 可以立即执行的
  - 具有内置的文档说明

# Movie Collection Starter Kit

- 特点

- 收藏您喜欢的电影
- 可以查找电影
- 可以在线查找电影

- 技术

- Windows Form – UI
- Visual C# 2.0 – 包括泛型（generics）

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# Demonstration One

## *demo*

### Movie Collection...



# 构造函数 Constructor (1)

您的潜力, 我们的动力

Microsoft®  
微软(中国)有限公司

- 类中特殊的方法, 多用于初始化实例的数据成员, 在实例化 **new** 时被自动调用
- 函数名与类名**总是相同**, 没有返回值

```
public class BankCustomer
{
    private string    FirstName;
    private string    LastName;
    private decimal   Balance;
    .
    .
    .

```

→ 

```
public BankCustomer(string fn, string ln, decimal bal)
{
    this.FirstName = fn;
    this.LastName  = ln;
    this.Balance   = bal;
}
}
```

```
BankCustomer c;
c = new BankCustomer();
c = new BankCustomer("kim", "lee", 1000.0M);
```

X

# 构造函数 Constructor (2)

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- 通过重载, 构造函数可以任意多

```
public class BankCustomer
{
    public BankCustomer(string fn, string ln, decimal bal)
    {
        this.FirstName = fn; this.LastName = ln; this.Balance = bal;
    }

    public BankCustomer ()
    {
        this.FirstName = ""; this.LastName = ""; this.Balance = 0.0M;
    }

    public BankCustomer(string fn, string ln) // start with 0.0 balance
    {
        this.FirstName = fn; this.LastName = ln; this.Balance = 0.0M;
    }

    public BankCustomer(System.IO.StreamReader reader) // initialize via stream (e.g. file)
    {
        this.FirstName = reader.ReadLine();
        this.LastName = reader.ReadLine();
        this.Balance = decimal.Parse( reader.ReadLine() );
    }
}
```

# 构造函数 Constructor (3)

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- 构造函数间的调用

```
public class BankCustomer
{
    public BankCustomer(string fn, string ln, decimal bal)
    {
        this.FirstName = fn;
        this.LastName = ln;
        this.Balance = bal;
    }

    public BankCustomer(string fn, string ln) // start with 0.0 balance
    : this(fn, ln, 0.0M)
    { }

    public BankCustomer(System.IO.StreamReader reader) // initialize via stream (e.g. file)
    : this(reader.ReadLine(), reader.ReadLine(), decimal.Parse(reader.ReadLine()))
    { }
```



# 构造函数 Constructor (4)

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- 类中特殊的方法, 多用于初始化实例的数据成员, 在实例化 **new** 时被自动调用
  - 函数名与类名**总是相同**
  - 没有返回值
  - 任意数量 (通过重载**Overload**)
  - 构造函数间的调用
  - 访问修饰符的作用 (**public protected private**)
- 如果没有显式定义, 那么系统提供一个不带任何参数的**public**的构造函数
- 静态构造函数

# 属性 Property (1)

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 内部看像函数，外部看像字段
- 读/写由get/set存取器来进行
  - 没有参数列表
- **Adv:** 可以像给公有字段赋值一样轻松，同时允许通过get/set来控制对属性的访问

# 属性 Property (2)

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

```
public class BankCustomer
{
    :
    :
    private decimal m_Balance; //field
    public decimal Balance //property
    {
        get { return this.m_Balance; }
        set
        {
            if (value < 0.0M) //new balance cannot be negative! So ignore...
            ;
            else //update balance to new value
                this.m_Balance = value;
        }
    }
}
```

```
bc.Balance = bc.Balance - amt;
```

# 索引器 Indexer (1)

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- 索引器就是一类特殊的属性, 通过它们你就可以像处理数组一样处理对象。
- 索引器通常在代表元素集合的类中定义

```
public class BankCustomer
{
    private double balance;
    private Person[ ] relatives;
    public BankCustomer()
    {
        relatives = new Person[100];
    }
    ...
}
```

# 索引器 Indexer (2)

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 索引器类似于属性
- 属性名是 *this*，意思是回引类的当前实例，参数列表包含在方括号而非括号之内。

```
public Person this [int index]
{
    get
    {
        return relatives [index];
    }
    set
    {
        if (value != null)
        {
            relatives [index] = value;
        }
    }
}
```

```
BankCustomer bc = new
    BankCustomer();
Bc[10] = new Person("Name");
```



# 事件 Event (1)

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- C#中使用委托模型来实现事件
- 事件处理方法不必在将生成事件的类中定义
- 需要做的事情就是把事件源和事件处理程序结合起来。
- 使用事件处理委托, 简称事件委托可以定义为生成事件的类的一个成员
- 事件委托是多播的

# 事件 Event (2)

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- 事件委托的形式:

```
public delegate void MouseEventHandler (object source, EventArgs e);
```

**object source** : 事件源

**EventArgs** : **System.EventArgs** 类的实例或派生类实例, 它包含事件的另外的信息

- .NET Framework 定义了大量的事件处理委托

- `public delegate void KeyEventHandler(object source, KeyEventArgs args)`
- `public delegate void MouseEventHandler(object source, MouseEventArgs args)`
- .....

# 事件 Event (3)

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 用户自定义事件委托

```
public delegate void NameEventHandler (object source, EventArgs e);
```

- 必须自己来定义NameEventArgs类

- 创建事件委托的实例

➢ 不使用new关键字，使用event关键字

```
public event NameEventHandler handler;
```

# 事件 Event (4)

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 触发事件
  - 把事件委托的一个实例定义为类的成员
  - 确定何时生成事件代码
  - 定义生成提供事件的**EventArgs**对象的代码



# 事件Demo

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

## *demo*

- ▶ 自定义事件
- ▶ 定义NameList类，当调用add()方法的时候，查看事件处理机（EventHandler）是否存在，如果存在，产生事件委托，由处理机处理。



# Movie Collection 结构

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

**MainForm.cs**

核心窗体，是其他控件的容器

**ListDetails.cs**

自定义用户控件：查看和修改你的电影收藏

**SearchOnline.cs**

自定义用户控件：在线 web services 的查找

**DVDCollectionDatabase.mdf**

项目数据库，用于存储电影信息

**DVDCollectionDataSet.xsd**

Dataset 和数据层对象。用于执行数据库操作、

**Resources folder**

资源文件夹

**Documentation folder**

帮助文档

**Controls folder**

用户使用的控件

**Web References folder**

Web Service 引用

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

# Demonstration Two

## *demo*

### Movie Collection 分析...

# Movie Collection 技术

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- Windows Forms – 用来创建UI
- Settings and Resources – Designers
- C# 2.0 - generics

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# Demonstration Three

## *demo*

Movie Collection 更改...

# 建立自己的 Starter Kits

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- **Starter Kits** 是一个项目的模版
- 项目模版 ( **Project Templates** ) 是.zip文件
  - 包括了整个项目
  - 包括了 .vstemplate 文件
- 将你的项目输出成模版
  - **File|Export Template**
  - 向导将自动建立 .zip 和 .vstemplate 文件



# 总结

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- Starter Kits
- Movie Collection Starter Kits
- 建立自己的 Starter Kit

# 其它参考资源

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- **Download Starter Kit**

- <http://msdn.microsoft.com/vstudio/downloads/starterkits>

- **Webcast:**

- <http://www.microsoft.com/china/msdn/webcasts>

- **Resources:**

- MSDN Library  
<http://www.microsoft.com/china/MSDN/library/default.mspix>

- **MSDN flash:**

- <http://www.microsoft.com/msdn/china/newsletter/>

# 感谢大家的参与!

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司


- 非常感谢大家的参与
- 您的**反馈**对我们来说是非常重要的



Email: [msdnonln@microsoft.com](mailto:msdnonln@microsoft.com)


# Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问



提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

**Microsoft®**  
微软(中国)有限公司

**Microsoft®**

msdn  


**MSDN Webcasts**