

DIANCHENG JICHUSHIYAN

软件开发与应用专业系列教材

DIANCHENG JICHUSHIYAN

# NET 编程基础实验

邱钦伦 刘雍豪 编

中央广播电视大学出版社

软件开发与应用专业系列教材

# **.NET 编程基础实验**

邱钦伦 刘雍豪 编

中央广播电视大学出版社

## 图书在版编目 (CIP) 数据

.NET 编程基础实验 / 邱钦伦, 刘雍豪编. —北京: 中央广播电视大学出版社, 2005.1

(软件开发与应用专业系列教材)

ISBN 7-304-02927-7

I. .N… II. ①邱…②刘… III. 主页制作—程序设计—实验—电视大学—教材 IV. TP393.092-33

中国版本图书馆 CIP 数据核字 (2005) 第 003651 号

版权所有, 翻印必究。

软件开发与应用专业系列教材

### .NET 编程基础实验

邱钦伦 刘雍豪 编

---

出版·发行: 中央广播电视大学出版社

电话: 发行部: 010-68519502 总编室: 010-68182524

网址: <http://www.crtvup.com.cn>

地址: 北京市海淀区西四环中路 45 号

邮编: 100039

经销: 新华书店北京发行所

---

策划编辑: 何勇军

责任编辑: 王立群

印刷: 北京集惠印刷有限公司

印数: 0001~5000

版本: 2004 年 11 月第 1 版

2005 年 1 月第 1 次印刷

开本: 787×1092 1/16

印张: 5.75 字数: 140 千字

---

书号: ISBN 7-304-02927-7/TP·238

定价: 11.00 元

---

(如有缺页或倒装, 本社负责退换)

## 内 容 提 要

《.NET 编程基础》是中央广播电视大学软件开发与应用专业信息系统开发方向和网站应用编程方向的专业核心课程。该课程以 .NET 框架的基础知识和 ADO.NET 为核心, 具有非常实用和先进的特点。

本书是与《.NET 编程基础》相配套的实验教材, 全书分为九个实验, 基本上每章都有一个实验。其中第 1 章是 .NET 知识的概述, 没有实验, 第 10 章作为了解内容, 没有实际操作要求。

由于这些实验简单实用, 并配备了大量插图, 所以读者也可以将本书独立使用, 作为快速了解 .NET 框架知识的参考资料。

# 前 言

《.NET 编程基础》是中央广播电视大学软件开发与应用专业信息系统开发方向和网站应用编程方向的专业核心课程。该课程以 .NET 框架的基础知识和 ADO.NET 为核心, 具有非常实用和先进的特点。

为了提高学生的学习质量, 这门课程配备了相应的实验教材。《.NET 编程基础实验》包含九个实验:

1. 通用类型操作实验
2. 字符串数组集合
3. 使用类
4. 数据流和文件操作
5. 使用 ADO.NET 获取数据
6. 使用 ADO.NET 修改数据
7. 实现表的主从关系
8. 创建和使用 Web 服务
9. 部署 Windows 应用程序

考虑到电大学生的基础, 本课程所设置的实验都比较简单实用。学生通过实验, 能够进一步加深对《.NET 编程基础》的理解, 提高实际动手能力。

本书编写过程中得到中央电大各级领导和中央电大出版社领导及编辑人员的关心、指导和大力支持, 也得到了很多同仁的帮助, 在此表示衷心感谢。

由于时间紧迫, 能力有限, 不足之处请批评指正。

《.NET 编程基础》编写组

2004 年 8 月

# 目 录

|                                         |      |
|-----------------------------------------|------|
| 实验一 通用类型操作实验 .....                      | (1)  |
| 实验目的 .....                              | (1)  |
| 练习1 使用枚举类型 .....                        | (2)  |
| 练习2 创建自定义值类型 .....                      | (4)  |
| 练习3 创建一个对象 .....                        | (6)  |
| 练习4 值类型和引用类型的赋值 .....                   | (8)  |
| 实验二 字符串数组集合 .....                       | (12) |
| 实验目的 .....                              | (12) |
| 练习 字符串的拆分和排序 .....                      | (13) |
| 实验三 使用类 .....                           | (17) |
| 实验目的 .....                              | (17) |
| 练习 创建和使用类 .....                         | (18) |
| 实验四 数据流和文件操作 .....                      | (22) |
| 实验目的 .....                              | (22) |
| 练习1 搜索文件 .....                          | (23) |
| 练习2 搜索目录 .....                          | (25) |
| 练习3 带子文件夹的搜索 .....                      | (28) |
| 实验五 使用 ADO.NET 获取数据 .....               | (33) |
| 实验目的 .....                              | (33) |
| 练习1 连接的打开与关闭 .....                      | (34) |
| 练习2 使用 DataAdapter 和 DataSet 获取数据 ..... | (38) |
| 练习3 使用 Command 和 DataReader 获取数据 .....  | (42) |

|                                          |             |
|------------------------------------------|-------------|
| <b>实验六 使用 ADO.NET 修改数据 .....</b>         | <b>(45)</b> |
| 实验目的 .....                               | (45)        |
| 练习 1 使用 Command 对象修改数据库 .....            | (46)        |
| 练习 2 使用 DataAdapter 和 DataSet 修改数据 ..... | (56)        |
| <b>实验七 实现表的主从关系 .....</b>                | <b>(67)</b> |
| 实验目的 .....                               | (67)        |
| 练习 创建基于 Windows 的主从关系 .....              | (68)        |
| <b>实验八 创建和使用 Web 服务 .....</b>            | <b>(73)</b> |
| 实验目的 .....                               | (73)        |
| 练习 创建和使用 Web 服务 .....                    | (74)        |
| <b>实验九 部署 Windows 应用程序 .....</b>         | <b>(79)</b> |
| 实验目的 .....                               | (79)        |
| 预备知识 .....                               | (80)        |
| 练习 部署 Windows 应用程序 .....                 | (80)        |

# 实验一

## 通用类型操作实验

### 实验目的

完成本实验后，你将能够：

- 了解值类型与引用类型的区别
- 了解如何创建自定义类型
- 能够创建枚举类型

## 练习 1 使用枚举类型

### 【摘 要】

本练习通过创建一个名为 Quarter 的枚举类型来描述季节。Quarter 包括四个成员 Spring, Summer, Autumn, Winter, 分别代表整数 0, 1, 2, 3。

### 【操作步骤指导】

#### 步骤 1: 创建一个“Windows 应用程序”新项目

打开 Visual Studio.NET, 然后新建一个项目, 项目名称为 Lab1 - 1, 项目类型选择“Visual Basic 项目”, 模板选择“Windows 应用程序”。

#### 步骤 2: 设计用户界面

用户界面如图 1-1 所示。

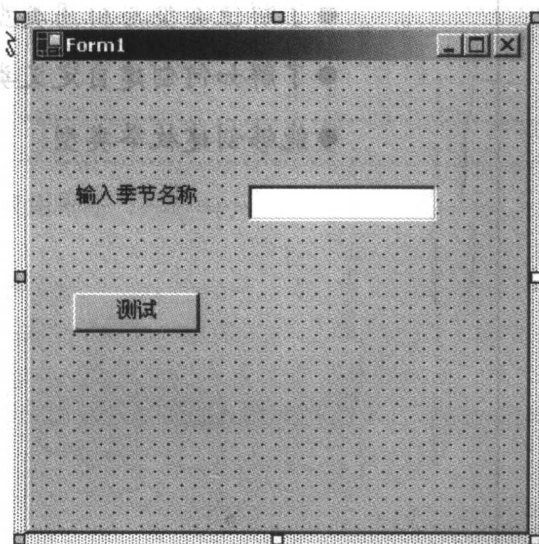


图 1-1 使用枚举类型的设计界面

#### 步骤3: 创建一个名为 Quarter 的 int 型枚举

```
Enum Quarter  
    Spring  
    Summer  
    Autumn  
    Winter  
End Enum
```

步骤 4: 在 Button1\_Click 事件中执行以下任务:

- (1) 定义一个 Quarter 变量 q;
- (2) 根据 TextBox1 中输入的文本, 利用 Select ... Case 语句在 q 变量中保存正确的 Quarter 文本字段;
- (3) 定义一个整型变量 q\_int, 存储变量 q 的数字值;
- (4) 使用 messagebox 输出 q\_int 的值。

步骤 5: 测试

运行程序。在 TextBox1 中输入一个季节名称 (例如 Autumn), 程序返回相应结果。如图 1-2。

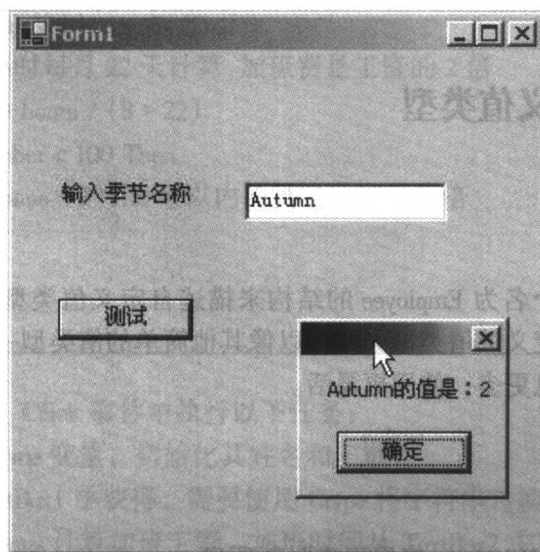


图 1-2 获得枚举类型的数值

## 【参考代码】

```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click  
    Dim q As quarter '定义 Quarter 枚举变量  
    Dim q_string As String = TextBox1.Text '枚举变量的文本  
    Dim q_int As Integer '枚举变量的数值  
  
    '根据输入文本获得枚举文本值  
    Select Case q_string  
        Case "Spring", "spring"  
            q = quarter.Spring  
        Case "Summer", "summer"  
            q = quarter.Summer
```

```

Case " Autumn ", " autumn "
    q = quarter. Autumn
Case " Winter ", " winter "
    q = quarter. Winter
End Select

q_int = q 获得枚举的数值
MessageBox.Show(q_string & "的值是:" & q_int.ToString)
End Sub

```

## 练习 2 创建自定义值类型

### 【摘 要】

本练习通过创建一个名为 Employee 的结构来描述自定义值类型的使用。从练习中我们可以看到，结构这种自定义的值类型同样可以像其他简单的值类型一样直接使用，只不过结构比简单值类型包含信息更多，使用更灵活。

### 【操作步骤指导】

#### 步骤 1: 创建一个“Windows 应用程序”新项目

打开 Visual Studio.NET，然后新建一个项目，项目名称为 Lab1 - 2，项目类型选择“Visual Basic 项目”，模板选择“Windows 应用程序”。

图 1-3 创建自定义值类型的设计界面

## 步骤2: 设计用户界面

用户界面如图 1-3 所示。

## 步骤3: 创建一个名为 Employee 的结构

```
Public Structure Employee
```

```
    Public Name As String '姓名
```

```
    Public EmployeeNumber As Integer '编号
```

```
    Public Salary As Double '工资
```

```
    Public Bonus As Double '加班费
```

```
    Sub CalculateBonus(ByVal hours As Single)
```

```
        '按照每天 8 小时每月 22 天计算,加班费是工资的 2 倍
```

```
        Bonus = Salary * hours / (8 * 22)
```

```
        If EmployeeNumber < 100 Then
```

```
            Bonus = Bonus * 2 '100 号以内的员工加班费翻倍
```

```
        End If
```

```
    End Sub
```

```
End Structure
```

## 步骤4: 在 Button1\_Click 事件中执行以下任务:

- (1) 定义一个 Employee 变量, 初始化其姓名和工资;
- (2) 雇员编号从 TextBox1 中获得, 需要使用 CType 将字符串转换为整数;
- (3) 调用 CalculateBonus 计算加班工资, 加班时间从 TextBox2 中获得, 需要使用 CType 将字符串转换为整数;
- (4) 显示该雇员的加班费。

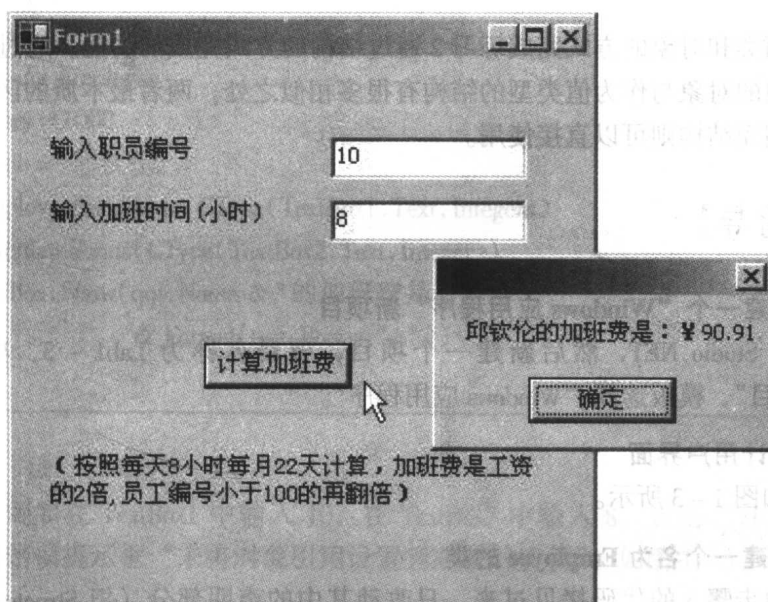


图 1-4 测试自定义值类型

### 步骤 5: 测试

运行程序。例如在 TextBox1 中输入 10, 在 TextBox2 中输入 8, 计算加班费后的结果如图 1-4 所示。

### 【参考代码】

```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click  
    Dim e1 As Employee '定义 Employee 结构变量  
    '赋值  
    e1.Salary = 1000  
    e1.Name = "邱钦伦"  
    e1.EmployeeNumber = (CType(TextBox1.Text, Integer))  
  
    e1.CalculateBonus(CType(TextBox2.Text, Integer))  
    MessageBox.Show(e1.Name & "的加班费是:" _  
        & Format(e1.Bonus, "c"))  
End Sub
```

## 练习 3 创建一个对象

### 【摘要】

本练习使用类和对象的方式完成练习 2 通过结构的方式完成的功能。通过本练习可以看到作为引用类型的对象与作为值类型的结构有很多相似之处, 两者最本质的区别是对象必须先创建才能使用而结构则可以直接使用。

### 【操作步骤指导】

#### 步骤 1: 创建一个“Windows 应用程序”新项目

打开 Visual Studio.NET, 然后新建一个项目, 项目名称为 Lab1-3, 项目类型选择“Visual Basic 项目”, 模板选择“Windows 应用程序”。

#### 步骤 2: 设计用户界面

用户界面如图 1-3 所示。

#### 步骤 3: 创建一个名为 Employee 的类

将练习 2 中步骤 3 的代码拷贝过来, 只改动其中的声明部分 (把 Structure 改为 Class),

改动后的代码如下：

```
Public Class Employee
    Public Name As String '姓名
    Public EmployeeNumber As Integer '编号
    Public Salary As Double '工资
    Public Bonus As Double '加班费
    Sub CalculateBonus(ByVal hours As Single)
        '按照每天 8 小时每月 22 天计算,加班费是工资的 2 倍
        Bonus = Salary * hours / (8 * 22)
        If EmployeeNumber < 100 Then
            Bonus = Bonus * 2 '100 号以内的员工加班费翻倍
        End If
    End Sub
End Class
```

**步骤 4：在 Button1\_Click 事件中执行以下任务：**

- (1) 定义一个 Employee 对象，初始化其姓名和工资；
- (2) 雇员编号从 TextBox1 中获得，需要使用 CType 将字符串转换为整数；
- (3) 调用 CalculateBonus 计算加班工资，加班时间从 TextBox2 中获得，需要使用 CType 将字符串转换为整数；
- (4) 显示该雇员的加班费。

直接拷贝练习 2 中步骤 4 的代码，没有任何语法错误，如下所示：

```
Private Sub Button1_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs)Handles Button1.Click
    Dim qql As Employee
    qql.Salary = 1000
    qql.Name = "邱钦伦"
    qql.EmployeeNumber = (CType(TextBox1.Text, Integer))
    qql.CalculateBonus(CType(TextBox2.Text, Integer))
    MessageBox.Show(qql.Name & " 的加班费是：" _
        & Format(qql.Bonus, "c"))
End Sub
```

**步骤 5：测 试**

运行程序。例如在 TextBox1 中输入 10，在 TextBox2 中输入 8，点击“计算加班费”按钮，程序会弹出错误提示框“未将对象引用设置到对象的实例”，如图 1-5 所示。

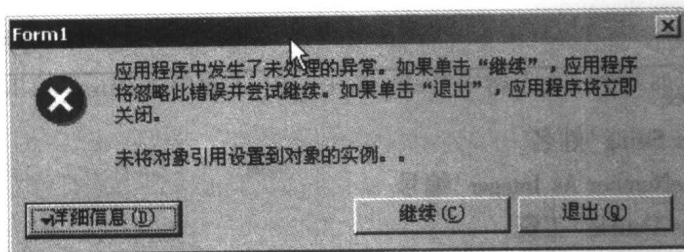


图 1-5 错误对话框

上述错误正是引用类型（这里是类 Employee）与值类型（练习 2 中的结构 Employee）的区别所在，类作为引用类型，在定义之后还必须通过 New 获得该类的一个具体实例，才能访问这个对象的内容。因此我们需要将第一句定义对象的语句修改为：

```
Dim qql As New Employee()
```

修改后的程序运行结果如图 1-4 所示。

## 练习 4 值类型和引用类型的赋值

### 【摘 要】

本练习通过结构和类的赋值操作比较值类型和引用类型的不同。

对于类，两个变量引用同一个对象是可能的，因此对一个变量进行的操作可能影响另一个变量所引用的对象。对于结构，每个变量都有自己的数据副本，因此对一个变量进行的操作不可能影响另一个变量。

### 【操作步骤指导】

#### 步骤 1：创建一个“Windows 应用程序”新项目

打开 Visual Studio.NET，然后新建一个项目，项目名称为 Lab1-4，项目类型选择“Visual Basic 项目”，模板选择“Windows 应用程序”。

#### 步骤 2：设计用户界面

用户界面如图 1-6 所示。Button 按钮“测试结构”的名称为 BtnStructure，Button 按钮“测试类”的名称为 BtnClass。

#### 步骤 3：创建类和结构

为了简便起见，将练习 2 的结构和练习 3 的类中声明代码简化，代码如下：

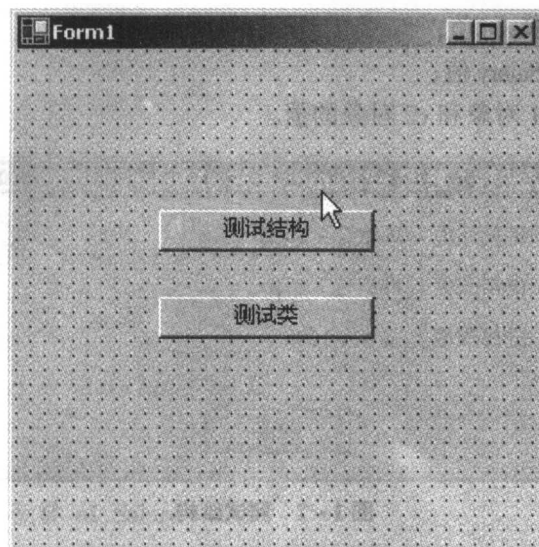


图 1-6 错误对话框

```
Public Structure S_Employee
    Public Name As String '姓名
    Public Salary As Double '工资
End Structure
```

```
Public Class C_Employee
    Public Name As String '姓名
    Public Salary As Double '工资
End Class
```

**步骤 4:** 在 **BtnStructure** 事件中执行以下任务:

- (1) 定义一个 **S\_Employee** 对象 **e1**, 初始化其姓名和工资;
- (2) 定义一个 **S\_Employee** 对象 **e2**;
- (3) 将变量 **e1** 赋值给变量 **e2**;
- (4) 修改变量 **e2** 的 **Name** 字段;
- (5) 修改变量 **e1** 的 **Salary** 字段;
- (6) 显示修改后的 **e1** 变量和 **e2** 变量的值。

**步骤 5:** 测试结构值的改变

运行后, 点击 **BtnStructure** 按钮, 运行结果如图 1-7 所示。

**步骤 6:** 在 **BtnClass** 事件中执行以下任务:

- (1) 定义一个 **C\_Employee** 对象 **c1** 并实例化, 初始化其姓名和工资;
- (2) 定义一个 **S\_Employee** 对象 **c2**;
- (3) 将对象 **c1** 指向对象 **c2**;

- (4) 修改对象 c2 的 Name 值;
- (5) 修改对象 c1 的 Salary 值;
- (6) 显示修改后的 c1 对象和 c2 对象的值。

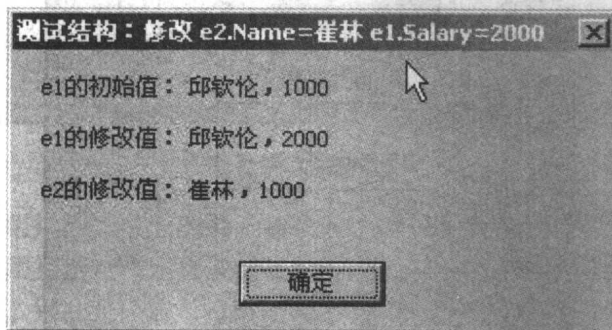


图 1-7 测试结构

#### 步骤 7: 测试对象值的改变

运行后, 点击 BtnClass 按钮, 运行结果如图 1-8 所示。

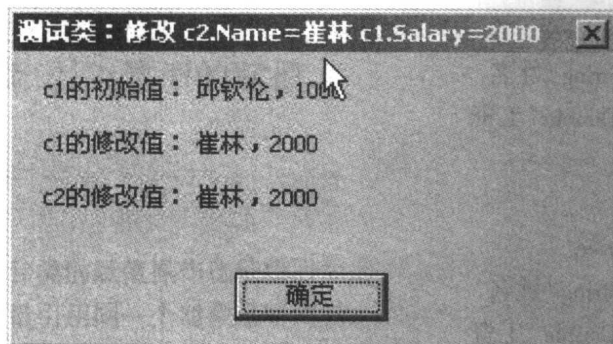


图 1-8 测试类

### 【参考代码】

测试结构的代码:

```
Private Sub BtnStructure_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnStructure.Click
    Dim e1 As S_Employee
    e1.Salary = 1000
    e1.Name = "邱钦伦"

    Dim str As String = "e1 的初始值: " & e1.Name
    & ", " & e1.Salary & Chr(13) & Chr(13)

    Dim e2 As S_Employee
```