

Práctica B: Uso de estructuras y matrices

Objetivos

En este laboratorio, aprenderá a:

- Declarar y utilizar una estructura.
- Declarar y utilizar una matriz.

Nota Este laboratorio se centra en los conceptos de este módulo y, por ello, podría no cumplir las recomendaciones de seguridad de Microsoft.

Requisitos previos

Para trabajar en este laboratorio, debe:

- Estar familiarizado con el entorno de desarrollo de Visual Studio .NET.
- Saber declarar variables.
- Saber modificar formularios y controles.

Nota Si desea más información sobre estructuras, lea “Estructuras: sus propios tipos de datos”, en la documentación de Visual Studio .NET. Encontrará más información sobre matrices en la documentación de Visual Basic .NET.

Escenario

En este laboratorio, creará y utilizará una estructura en una aplicación existente. También añadirá una matriz para permitir que la estructura se utilice y muestre tres veces. Aunque el contexto de este laboratorio es una aplicación de préstamo, no forma parte del escenario de aplicación de préstamo utilizado en otros laboratorios.

Archivos de solución

Los archivos de solución para este laboratorio se encuentran en las carpetas

PracticaB\Ex01\Solution y PracticaB\Ex02\Solution dentro del archivo labs03.zip.

Tiempo estimado para
realizar este laboratorio:
30 minutos

Ejercicio 1

Crear una estructura

En este ejercicio, declarará una estructura que contendrá cuatro miembros para simular datos del solicitante de un préstamo. Asignará datos a los miembros de la estructura y mostrará todos los datos en un cuadro de mensaje.

🔍 Abrir el código de inicio para el proyecto

1. Abra Visual Studio .NET.
2. Abra el proyecto StructuresArrays.sln, que se encuentra en la carpeta PracticaB\Ex01\Starter dentro del archivo labs03.zip

Importante Aunque el contexto de este laboratorio es una aplicación de préstamo, no forma parte del escenario de aplicación de préstamo utilizado en otros laboratorios.

🔍 Añadir una estructura de datos

1. Abrir el Editor de código de Form1.vb.
2. Sobre la sentencia **MessageBox** comentada, cree una estructura *loanApplicant* con los siguientes miembros y tipos de datos:

Nombre del miembro	Tipo de datos
<i>ApplicantNumber</i>	Integer
<i>Name</i>	String
<i>Salary</i>	Decimal
<i>LoanRequested</i>	Decimal

Su código debería tener un aspecto similar al siguiente:

```
Public Structure loanApplicant
    Dim ApplicantNumber As Integer
    Dim Name As String
    Dim Salary As Decimal
    Dim LoanRequested As Decimal
End Structure
```

3. Declare una variable *applicant* a nivel de módulo como una estructura *loanApplicant*. Su código debería ser parecido al siguiente:

```
Dim applicant as loanApplicant
```

🔗 Rellene los miembros de la estructura con datos del formulario

1. En el gestor de eventos **AddButton_Click**, asigne un número de inicio al miembro *ApplicantNumber* de la estructura *loanApplicant* para utilizarlo cuando muestre la información (el código que debe añadirse aparece en negrita en el ejemplo siguiente). Asignar este número de inicio también le ayudará a realizar un seguimiento de los elementos de la matriz en el Ejercicio 2.

```
Private Sub AddButton_Click(...)
    applicant.ApplicantNumber = 1
End Sub
```

2. En el gestor de eventos **AddButton_Click**, escriba código para asignar valores de las propiedades del cuadro de texto a los miembros de la estructura, como sigue (el código que debe añadirse aparece en negrita).

```
Private Sub AddButton_Click(...)
    applicant.ApplicantNumber = 1
    applicant.Name = NameTextBox.Text
    applicant.Salary = CDec(SalaryTextBox.Text)
    applicant.LoanRequested = _
        CDec(LoanRequestedTextBox.Text)
End Sub
```

🔗 Mostrar la información de la estructura al usuario

1. En el gestor de eventos **DisplayButton_Click**, quite los comentarios y, a continuación, examine la línea de código que se proporciona en el código de ejemplo. Este código muestra cada miembro de la estructura en líneas separadas en un cuadro de texto. El código comentado se reproduce a continuación.


```
MessageBox.Show("Loan Applicant No: " & _
    applicant.ApplicantNumber & ControlChars.CrLf & _
    "Loan Applicant Name: " & applicant.Name & _
    ControlChars.CrLf & "Salary: " & _
    applicant.Salary & ControlChars.CrLf & _
    & "Loan Requested: " & applicant.LoanRequested)
```

Nota Este código utiliza la constante **ControlChars.CrLf** para crear retornos de carro en el texto del cuadro de mensaje. Encontrará más información sobre **ControlChars.CrLf** en “Imprimir y visualizar constantes”, en la documentación de Visual Basic .NET.

2. En el gestor de eventos **DisplayButton_Click**, cree un segundo cuadro de mensaje para informar al usuario de que la aplicación se cerrará, y escriba el código para cerrar la aplicación. Su código debería ser similar al siguiente:

```
MessageBox.Show("La aplicación se cerrará ahora.")
Application.Exit( )
```

⏏ Ejecutar y probar la aplicación

1. Para ejecutar la aplicación, en la barra de herramientas estándar, haga clic en el botón **Iniciar** .
2. Escriba datos de ejemplo en los cuadros **Nombre**, **Salario** y **Préstamo Solicitado**.
3. Haga clic en **Add to Structure**.
Cuando haga clic en **Add to Structure**, no ocurrirá nada visible.
4. Haga clic en **Display** y los elementos de la estructura aparecerán en un cuadro de mensaje.
5. Haga clic en **OK** para cerrar el cuadro de mensaje y, a continuación, haga clic de nuevo en **OK** para cerrar la aplicación.

Ejercicio 2

Crear y utilizar matrices

En este ejercicio, añadirá una matriz basada en la estructura *loanApplicant*. Modificará el gestor de eventos **AddButton_Click** para permitir que puedan añadirse múltiples solicitantes a la matriz. También modificará el gestor de eventos **DisplayButton_Click** para contener la nueva matriz.

🔍 Crear una matriz basada en la estructura *loanApplicant*

1. Abra el archivo StructuresArrays.sln que creó en el Laboratorio 3.2, Ejercicio 1 (si no está abierto).
Si todavía no ha finalizado el Ejercicio 1 del Laboratorio 3.2, abra el archivo StructuresArrays.sln de la carpeta PracticaB\Ex02\Starter dentro del archivo lab03.zip.

2. Abra el Editor de código de Form1.vb.
3. Localice el código que declara la variable *loanApplicant*. En su lugar, declare una matriz de tipo *loanApplicant* de tres elementos, como sigue:

```
Dim applicant(2) As loanApplicant
```

🔍 Seguimiento del número de solicitantes

1. En el gestor de eventos **AddButton_Click**, escriba el código para declarar una variable estática como contador, y para inicializarla a 1, como sigue:

```
Static countClicks As Integer
```

2. En el gestor de eventos **AddButton_Click**, después de la declaración de la variable estática, cambie las referencias a la variable *loanApplicant* para hacer referencia a la matriz utilizando *countClicks* para adquirir el número del elemento de matriz actual, como sigue:

```
applicant(countClicks).ApplicantNumber = countClicks + 1  
applicant(countClicks).Name = NameTextBox.Text  
applicant(countClicks).Salary = CDec(SalaryTextBox.Text)  
applicant(countClicks).LoanRequested = _  
    CDec(LoanRequestedTextBox.Text)
```

3. En el gestor de eventos **AddButton_Click**, tras el código que establece cada conjunto de propiedades de la matriz, escriba el código para incrementar *countClicks* cada vez que el usuario haga clic en **Add**, y para capturar los valores actuales incrementando el contador. Su código debería ser similar al siguiente:

```
countClicks +=1
```

🔍 Prepare el cuadro de texto para la siguiente secuencia de adición

- En el gestor de eventos **AddButton_Click**, justo tras el código que incrementa *countClicks*, escriba el código para establecer el foco en el control **NameTextBox** y seleccione todo su texto. Su código debería ser similar al siguiente:

```
NombreTextBox.Focus( )  
NombreTextBox.SelectAll( )
```

✍ Escribir código para contemplar múltiples solicitudes de préstamos

1. En el gestor de eventos **DisplayButton_Click**, edite la sentencia **MessageBox** para utilizar la matriz reemplazando las referencias a `applicant` con `applicant(0)`.
2. Abra el archivo `MessageBoxCode.txt` de la carpeta `PracticaB\Ex02\Starter` dentro del archivo `labs03.zip`, corte las tres líneas de código y péguelas en su código.

También puede copiar toda la sentencia **MessageBox**, y pegarlo dos veces detrás de la sentencia **MessageBox** inicial. Si elige este método alternativo, incremente los números de referencia de la matriz a `applicant(1)` en la segunda sentencia y a `applicant(2)` en la tercera sentencia.

3. Verifique que las dos sentencias finales en el procedimiento de eventos **DisplayButton_Click** son la sentencia **MessageBox** indicando que la aplicación se cerrará y la sentencia que cierra la aplicación.

Su código para todo el procedimiento de eventos **DisplayButton_Click** debería ser como sigue:


```
Private Sub DisplayButton_Click(...)
    MessageBox.Show("Loan Applicant No: " & _
        applicant(0).ApplicantNumber & ControlChars.CrLf & _
        "Loan Applicant Name: " & applicant(0).Name & _
        ControlChars.CrLf & "Salary: " & _
        applicant(0).Salary & ControlChars.CrLf & _
        "Loan Requested: " & applicant(0).LoanRequested)

    MessageBox.Show("Loan Applicant No: " & _
        applicant(1).ApplicantNumber & ControlChars.CrLf & _
        "Loan Applicant Name: " & applicant(1).Name & _
        ControlChars.CrLf & "Salary: " & _
        applicant(1).Salary & ControlChars.CrLf & _
        "Loan Requested: " & applicant(1).LoanRequested)

    MessageBox.Show("Loan Applicant No: " & _
        applicant(2).ApplicantNumber & ControlChars.CrLf & _
        "Loan Applicant Name: " & applicant(2).Name & _
        ControlChars.CrLf & "Salary: " & _
        applicant(2).Salary & ControlChars.CrLf & _
        "Loan Requested: " & applicant(2).LoanRequested)

    MessageBox.Show("Application will close now.")
    Application.Exit( )
End Sub
```

⏏ Ejecutar y probar la aplicación

1. Para ejecutar la aplicación, en la barra de herramientas estándar, haga clic en el botón **Iniciar** .
2. Introduzca datos de ejemplo en los cuadros de **Nombre**, **Salario** y **Préstamo Solicitado**.
3. Haga clic en **Add to Structure**.
4. Repita los pasos 2 y 3 dos veces más, hasta un total de tres veces.
5. Haga clic en **Display** y, a continuación, haga clic en **OK** en todos los cuadros de mensaje.
6. Cierre Visual Studio .NET.

Nota Cuando haga clic en **Add to Structure**, aparentemente no ocurrirá nada en el formulario, y volverá al primer cuadro de texto. Sin embargo, los miembros de la estructura se están poblando en memoria. A continuación, cuando haga clic en **Display**, verá que la información de cada solicitante aparece en otro cuadro de mensaje. La aplicación generará una excepción si hace clic en **Add to Structure** más de tres veces. La estructura correcta para gestionar esto es una estructura **If. .Then** o un bucle. Lo estudiaremos en el Módulo 5, “Bucles y estructuras de decisión”.
