

## والباب الثالث

### تطوير أصناف واجهة الاستخدام

#### DEVELOPING GRAPHICAL USER INTERFACE (GUI) CLASSES

- الفصل العاشر: نظرة عامة على مكونات واجهة المستخدم الرسومية في لغة فيجوال بيسك .نت
- الفصل الحادي عشر: استخدام نماذج متعددة مع أصناف مجال المشكلة
- الفصل الثاني عشر: نماذج الويب، ولغة HTML، وتقنية ASP.NET



### نظرة عامة على مكونات واجهة المستخدم الرسومية

#### في لغة فيجوال بيسك.نت

#### VB .NET GUI COMPONENTS OVERVIEW

##### أهداف الفصل:

- التعرف على أصناف واجهة المستخدم الرسومية في لغة فيجوال بيسك.نت.
- فهم أوامر فيجوال بيسك.نت الناتجة تلقائياً.
- معالجة أحداث فيجوال بيسك.نت.
- التعامل مع عناصر إضافية من GUI.

لقد تعرفنا في الباب الثاني من هذا الكتاب (من الفصل الأول إلى الفصل التاسع) كيفية برمجة أصناف مجال المشكلة حيث تعلمنا كيف نقوم بتطوير أصناف مجال مشكلة والتي تتكون من صفات وإجراءات، كما تعرفنا كيف ننشئ كلاً من علاقات الترابط وعلاقات التوارث بين أصناف مجال المشكلة. أما في هذا الباب (من الفصل العاشر إلى الفصل الثاني عشر) سنتعلم كيف تطور أصناف واجهة المستخدم الرسومية (Graphical User Interface Classes) التي تستخدم لاستقبال البيانات من المستخدم وإدخالها. سوف يقدم هذا الفصل مكونات واجهة المستخدم الأساسية، ويوضح الفصل الحادي عشر كيف تطور أصناف واجهة المستخدم لنظام شركة برادشو مارينا.

تتكون واجهة المستخدم الرسومية لأي تطبيق من كائنات أصناف واجهة المستخدم (GUI Classes) المعرفة داخل لغة فيجوال بيسك.نت. كما تعلمنا في الفصل الثاني إذا أردنا أن ننشئ نموذجاً (Form)، فيجب علينا استخدام مصمم نماذج الويندوز (Windows Form Designer) لتنسيق عناصر النموذج (مثل الأزرار Buttons والعناوين Labels) حيث يجب اختيار عناصر واجهة المستخدم من صندوق الأدوات ووضعها على النموذج ثم ضبط

خصائصها (مثل اللون والنص) مستخدماً نافذة الخصائص. عندما تنتهي من تصميم نموذج ما، ينشئ برنامج فيجوال ستوديو.نت (بالنيابة عنك) الأوامر المطلوبة تلقائياً لإنشاء هذا النموذج، ومن ثم يمكن أن تكمل بقية تطوير البرنامج وذلك بإضافة إجراءات معالجة الأحداث التي تصدر بواسطة المستخدم (مثل حدوث الضغط على زر الفأرة الأيسر أو مثل حدوث الضغط على أحد مفاتيح لوحة المفاتيح). سوف تتعلم خلال هذا الفصل الكثير عن الأحداث وكيفية تفاعل البرنامج مع المستخدم وذلك بإضافة مجموعه من الإجراءات التي تستدعى بشكل تلقائي عند صدور تلك الأحداث.

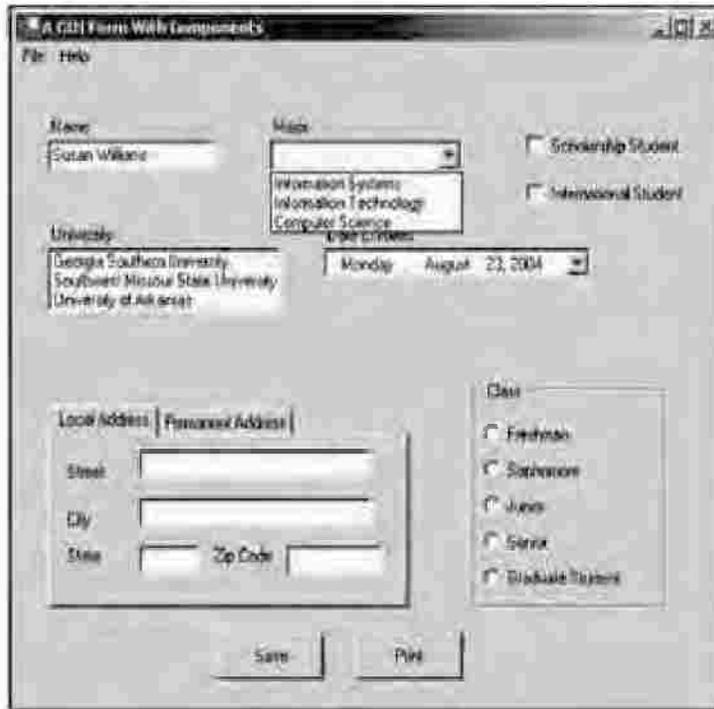
لقد قدمنا في الفصول الأولى من هذا الكتاب أساليب البرمجة المرئية لتطوير تطبيقات تحتوي على واجهات استخدام رسومية حيث تعلمنا في الفصل الثاني كيف تطور تطبيقات ويندوز مستخدمين نافذة صندوق الأدوات (Toolbox) لاختيار عناصر واجهة المستخدم ووضعها على النموذج، ثم استخدام نافذة الخصائص لضبط خصائص النموذج وعناصره. كما تعلمنا في الفصل السادس كيفية استخدام تلك التطبيقات لاختبار أصناف مجال المشكلة. يقدم هذا الفصل كيفية تصميم تطبيقات ذات واجهة رسومية مستخدماً العديد من العناصر المختلفة مثل الأزرار والعناوين وصناديق النصوص والقوائم وغيرها، وسوف تفهم أيضاً معنى الأوامر التي تنشئ تلقائياً نتيجة تصميم النموذج وكيفية كتابة إجراءات معالجة الأحداث المرتبطة بها.

### أصناف واجهة المستخدم الرسومية

#### Introducing the GUI Classes in VB .NET

يوضح الشكل رقم (١٠,١) نافذة تحتوي على العديد من عناصر مختلفة لواجهة المستخدم، وكما تعلمت في الفصل الثاني فإن النافذة يطلق عليها "نموذج" الذي يمثل كائناً من الصنف Form والذي ينتمي مثله مثل باقي أصناف واجهة المستخدم المقدمة في هذا الفصل إلى الفضاء المسمى "System.Windows.Forms". يحتوي النموذج الموضح في الشكل رقم (١٠,١) على العديد من عناصر واجهة المستخدم الرسومية مثل الأزرار (Buttons) والعناوين (Labels) وصناديق النصوص (Text Boxes) وصناديق التأشير (Check Boxes) وأزرار الاختيار (Radio Buttons) والصفحات المبهوبة (Tab Pages) وعناصر القائمة (Menu Items). يصف الجدول رقم (١٠,١) جميع هذه العناصر وغيرها من العناصر الأخرى، كما يوضح الشكل رقم (١٠,٢) جميع العناصر والمكونات المتاحة داخل صندوق أدوات تصميم النماذج.

نظرة عامة على مكونات واجهة المستخدم الرسومية (لغة الجيوان بيسك نت



الشكل رقم (١٠،١). نموذج بصوي على عناصر واجهة المستخدم متصلة.



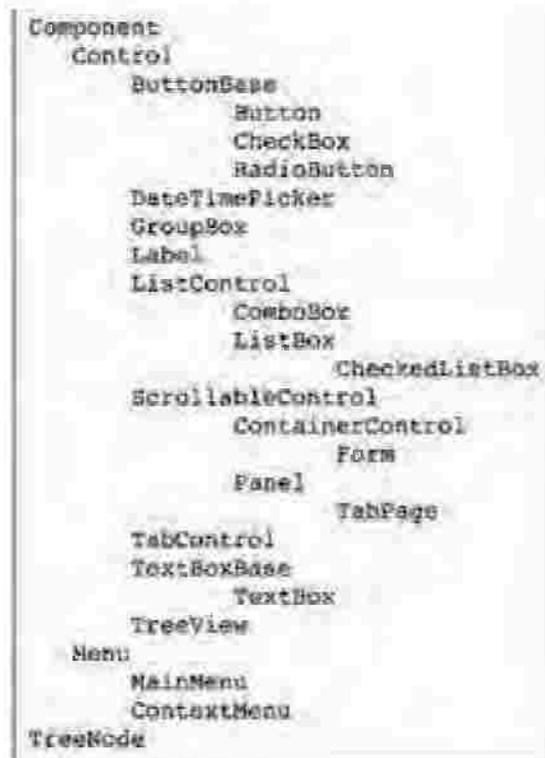
الشكل رقم (١٠،٢). عناصر ومكونات صندوق الأدوات.

الجدول رقم (١٠،١). وصف لعناصر مختارة لواجهة الاستخدام.

الوصف	العنصر
زر: يضغط المستخدم على هذا العنصر لإنجاز حدث محدد.	Button
صندوق تأشير: يستخدم هذا العنصر لاستقبال القيمة نعم (True) عند تأشير أو القيمة لا (False) عند عدم تأشير.	CheckBox
قائمة اختيار بالتأشير: يستخدم هذا العنصر لاختيار أكثر من عنصر في آن واحد، وعند اختيار أحد عناصر القائمة تظهر علامة التأشير داخل صندوق التأشير.	CheckedListBox
قائمة منسدلة: يستخدم هذا العنصر لاختيار قيمة أو أكثر من قائمة محددة من القيم.	ComboBox
مكون: يمثل هذا العنصر الصنف الأساسي لجميع عناصر واجهة الاستخدام.	Component
عنصر: يمثل هذا العنصر الصنف الأساسي لجميع عناصر واجهة الاستخدام المرئية.	Control
أداة التاريخ والوقت: يستخدم هذا العنصر لاختيار تاريخ من نتيجة مرئية.	DateTimePicker
نموذج: يمثل هذا العنصر نافذة حوارية التي تحتوي هذه النافذة على عناصر واجهة استخدام.	Form
صندوق تجميع: يستخدم هذا العنصر لتنظيم مجموعة من العناصر المرتبطة داخل إطار وتجميعها.	GroupBox
عنوان: يستخدم هذا العنصر لإظهار نص.	Label
قائمة: يستخدم هذا العنصر لاختيار قيمة أو أكثر من قائمة محددة من القيم.	ListBox
قائمة رئيسية: يستخدم هذا العنصر لإنشاء قائمة رئيسية تتفرع منها قوائم فرعية.	MainMenu
عنصر قائمة: يمثل هذا العنصر أحد عناصر القوائم الفرعية.	MenuItem
إطار: يستخدم هذا العنصر لتنظيم مجموعة من العناصر المرتبطة داخل إطار وتجميعها، وعادة يحتوي على أعمدة تصفح، ولكن لا يحتوي على عنوان.	Panel
زر اختيار: يقدم هذا العنصر إمكانية اختيار عنصر واحد من عدة عناصر.	RadioButton
عنصر التبويب: يقدم هذا العنصر إمكانية عمل الصفحات المبوبة (Tab Pages).	TabControl
صفحة مبوبة: يمثل هذا العنصر صفحة مبوبة.	TabPage
صندوق نصي: يستخدم هذا العنصر لاستقبال نص أو إظهاره.	TextBox
عقدة شجرية: يمثل هذا العنصر أحد عقد الشجرة داخل الرؤية الشجرية.	TreeNode
رؤية شجرية: يستخدم هذا العنصر لإظهار عناصر داخل شجرة.	TreeView

يجب أن تعلم عزيزي القارئ قبل البدء في تطوير التطبيقات التي تحتوي على واجهة رسومية أن أصناف واجهة المستخدم ترتبط بعلاقة توارث، وهذا ما يوضحه الشكل رقم (١٠،٣) حيث يوضح أن جميع أصناف واجهة المستخدم الرسومية هي أصناف فرعية للصنف Component عدا الصنف TreeNode، وأن معظم هذه الأصناف هي أيضاً أصناف فرعية للصنف Control، مع العلم أن الفرق بين الصنف Component والصنف Control هو أن كائنات الصنف Control تظهر في شكل رسومي ومرئي، بينما كائنات الصنف Component لا تظهر في شكل مرئي ورسومي، وأن الصنف Component يمثل الصنف الأساسي لجميع أصناف واجهة المستخدم والصنف Control

يقدم الوظائف الأساسية للعناصر المرئية فقط والتي تستخدم لتصميم النماذج يحتوي كل من الصف `Component` والصف `Control` على صفات وإجراءات هامة التي تورث لأصنافها الفرعية، ومن ثم يمكن استخدامها أثناء تصميم النماذج، كما يمكنك الاطلاع على المزيد من أصناف واجهة المستخدم من خلال إمكانية المساعدة وذلك بعدما تصبح متمكناً من المفاهيم الأساسية لهذه الأصناف.



الشكل رقم (١٠،٣). شجرة تورث أصناف واجهة المستخدم معقدة

### فهم أوامر فيجوال بيسك .نت المتوفرة آلياً أثناء مرحلة تصميم النماذج

#### Understanding the Code Generated by VB .NET

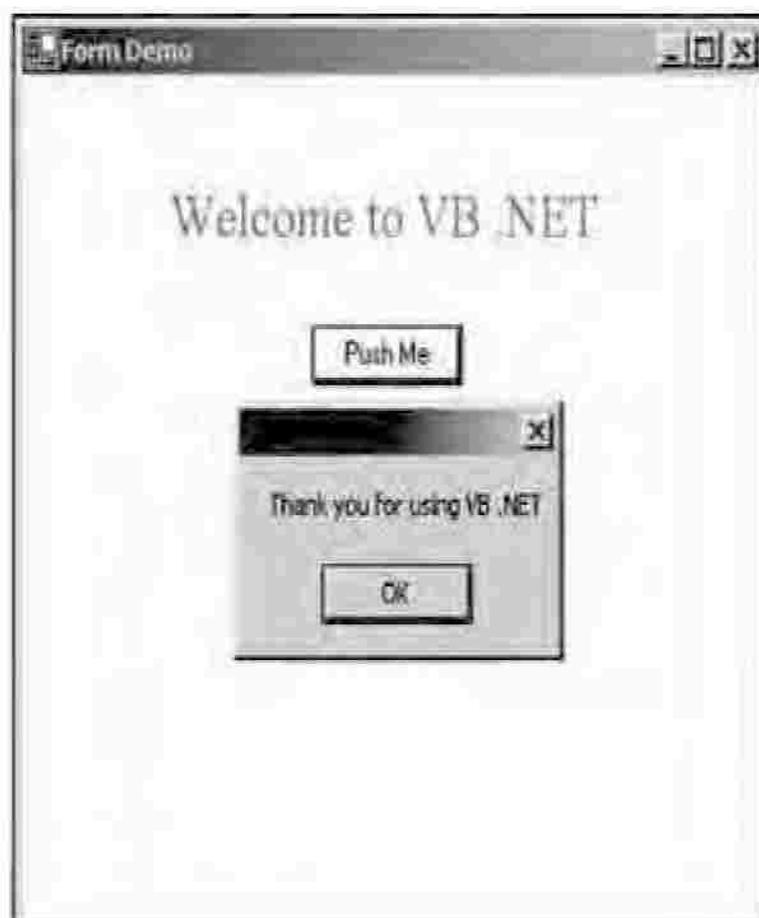
عندما تقوم بتطوير تطبيق يحتوي على واجهة مستخدم رسومية، فنحن نستخدم أسلوب البرمجة المرئية (Visual Programming) التي تتكون من تصميم عناصر النموذج داخله وإضافتها، ثم ضبط خصائص كل من النموذج وعناصره، ثم إضافة الإجراءات المطلوبة التي تعالج الأحداث (Events) التي تصدر من المستخدم عند تفاعله مع النموذج، مع العلم أن لغة فيجوال بيسك نت تنتج أوامر أصناف واجهة المستخدم الرسومية المطلوبة تلقائياً لتنفيذ تصميم النموذج وذلك أثناء تصميم عناصر النموذج وضبط خصائصها. يتبع تعريف أصناف واجهة المستخدم تركيب أوامر تعريف أصناف مجال المشكلة نفسه حيث إن الفرق بين صف واجهة المستخدم وصفح مجال

المشكلة هو وجوب إنشاء كائنات من أصناف واجهة المستخدم (مثل الأزرار وصناديق النصوص) داخل النموذج لكي تظهر على الشاشة أمام المستخدم (وظيفة الأوامر المتجهة تلقائياً بواسطة لغة فيجوال بيسك تحت).

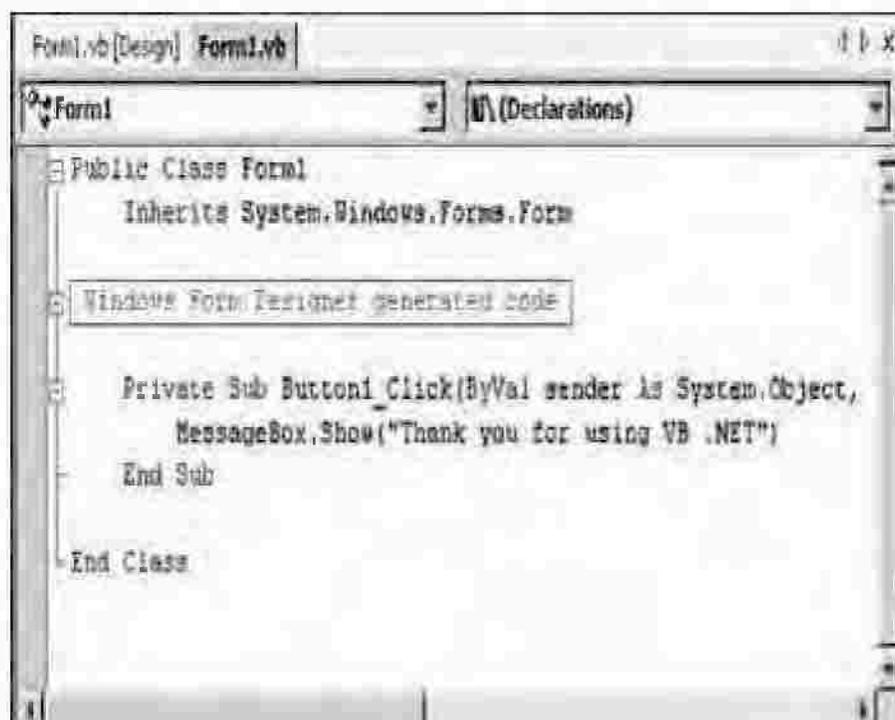
### اكتشاف برنامج FormDemo

#### Exploring the FormDemo Program

يشمل البرنامج FormDemo أول برنامج ذات واجهة رسومية داخل هذا الكتاب (راجع الفصل الثاني) والذي يحتوي على عنوان وعلى زر، وعند الضغط على هذا الزر تظهر الرسالة الموضحة في الشكل رقم (١٠.٤). يوضح الشكل رقم (١٠.٥) أوامر فيجوال بيسك تحت التي تولد ألياً لإنشاء كائنات أصناف واجهة المستخدم (الزر والعنوان) ومن ثم تنفيذ هذا النموذج.



الشكل رقم (١٠.٤). نموذج برنامج FormDemo.



الشكل رقم (١٠.٥). أوامر برنامج FormDemo.

وإذا عدنا إلى الشكل رقم (١٠.٣) نجد أن شجرة التوارث للصف `Form` تتكون من الصف `ContainerControl`، ثم الصف `ScrollableControl`، ثم الصف `Control`، ثم الصف `Component`، وهذا يعني أن الصف الذي يرث الصف `Form` سوف يرث جميع أصناف هذه الأصناف الأربعة وإجراءاتها. يعكس أول أمر داخل تعريف الصف `FormDemo` هذه الحقيقة حيث يرث الصف `FormDemo` الصف `Form` (بمعنى آخر أن الصف `FormDemo` مشتق من الصف `Form`). لاحظ أن الأوامر الناتجة تلقائياً بواسطة لغة فيجواول بيك نت تستخدم الاسم الكامل للأصناف مثل `"System.Windows.Forms.Form"` حيث كان من الممكن أن تستخدم الأمر `Import` لإدراج الفضاء المسمى `"System.Windows.Forms"`، ومن ثم تستخدم أسماء الأصناف مباشرة دون الحاجة لإضافة اسم الفضاء للمسمى (يصح الأمر هكذا `Inherits Form`).

أما بقية الأوامر الناتجة تلقائياً فهي مغطاة داخل الرابط `"Windows Form Designer Generated Code"` والذي يمكن أن تظهر بالنقر على الزر "+". يوضح الشكل رقم (١٠.٦) هذه الأوامر عند ظهورها، ولاحظ أن بعض سطور الأوامر طويلة جداً عن حدود عرض الصفحة؛ ولذلك لم تقسم هذه الأوامر باستخدام الرمز "..." والذي يعني أن الأمر مازال مستمراً.

```

#Region " Windows Form Designer generated code "
Public Sub New()
    MyBase.New()
    'This call is required by the Windows Form Designer.
    InitializeComponent()
    'Add any initialization after the InitializeComponent() call
End Sub

'Form overrides dispose to clean up the component list.
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
End Sub

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form
'Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents Button1 As System.Windows.Forms.Button
<System.Diagnostics.DebuggerStepThrough()> _
Private Sub InitializeComponent()
    Me.Label1 = New System.Windows.Forms.Label
    Me.Button1 = New System.Windows.Forms.Button
    Me.SuspendLayout()
    '
    'Label1
    '
    Me.Label1.Font = New System.Drawing.Font _
        ("Times New Roman", 18.0!, _
        System.Drawing.FontStyle.Regular,
        System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.Label1.ForeColor = System.Drawing.Color.Red
    Me.Label1.Location = New System.Drawing.Point(48, 32)
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(272, 48)
    Me.Label1.TabIndex = 0
    Me.Label1.Text = "Welcome to VB .NET"
    Me.Label1.TextAlign = _
        System.Drawing.ContentAlignment.MiddleCenter
    '
    'Button1
    '

```

الشكل رقم (٦، ١٠). أوامر البرنامج بعد الضغط على زر التوسعة.

```

Me.Button1.BackColor =
    System.Drawing.Color.FromArgb(CType(255, Byte) ' _
    CType(255, Byte), CType(128, Byte))
Me.Button1.Location = New System.Drawing.Point(144, 96)
Me.Button1.Name = "Button1"
Me.Button1.TabIndex = 1
Me.Button1.Text = "Push Me"
'
' Form1
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.BackColor = System.Drawing.Color.White
Me.ClientSize = New System.Drawing.Size(368, 325)
Me.Controls.AddRange(New System.Windows.Forms.Control() _
    {Me.Button1, Me.Label1})
Me.Controls.Add(Me.Label1)
Me.Name = "Form1"
Me.StartPosition = _
    System.Windows.Forms.FormStartPosition.CenterScreen
Me.Text = "Form Demo"
Me.ResumeLayout(False)

```

End Sub

#End Region

تابع الشكل رقم (٦، ١٠).

عزيزي القارئ لاحظ أن الأوامر الناتجة تلقائياً تبدأ بالتعليمة Region وتنتهي بالتعليمة End Region واللذان يشيران إلى أن الأوامر التي بينهما تنكمش إلى أمر واحد داخل نافذة محرر البرنامج، مع العلم أن انكماش هذه الأوامر يعطيك فرصة التركيز على الأجزاء الأخرى من البرنامج، وفي الحقيقة سوف لا نهتم كثيراً بهذه الأوامر؛ ولذلك ننصح أن تتركها في وضعها المنكمش.

تبدأ الأوامر (كما هو واضح في الأوامر التالية) بإجراء إنشاء الصنف FormDemo الذي يقوم باستدعاء إجراء إنشاء الصنف الرئيس له مستخدماً الأمر MyBase.New()، حيث تشير الكلمة المحجوزة MyBase إلى الصنف الرئيس للكائن الحالي، كما يمثل كائن الصنف FormDemo الكائن الحالي ويمثل الصنف Form الرئيس، ثم يتم استدعاء الإجراء InitializeComponent() المسؤول عن إسناد قيم ابتدائية لخصائص عناصر النموذج.

```

Public Sub New()
    MyBase.New()

```

```

' This call is required by the Windows Form Designer.
InitializeComponent()
' Add any initialization after the InitializeComponent() call

```

End Sub

ثم يظهر تعريف الإجراء (Dispose) المسؤول عن تحرير موارد النظام (الحاسب الآلي) عند إنهاء البرنامج ، مع العلم أن هذا الإجراء هو عكس إجراء الإنشاء حيث يطلق عليه اسم إجراء التدمير (Destructor Method) ، ويرث جميع الأصناف الفرعية للصف Component تعريف افتراضي للإجراء (Dispose) ، ولكن كما نرى فقد تم إعادة تعريف هذا الإجراء لإضافة بعض المهام الإضافية له مثل تحرير موارد النظام المستخدمة من قبل عناصر النموذج ومن النموذج ذاته. عزيزي القارئ لا توجد أي حاجة الآن لمعرفة تفاصيل هذه الأوامر ولكن المطلوب فقط منك هو معرفة الغرض منها.

```
'Form overrides dispose to clean up the component list.
Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
End Sub
```

تحتاج أداة مصمم نماذج الويندوز الأمر التالي الذي يعرف كائناً من الصف IContainer والذي يوفر مهاماً أساسية للحاويات (Containers) ، مع العلم أن الحاوية تمثل كائناً يحتوي على مكونات أخرى.

```
'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer
```

توضح الأوامر التالية عناصر النموذج التي تم استخدامها في تصميم النموذج (العنوان Label والزر Button) ، ولاحظ استخدام أسماء الأصناف كاملة (مثل System.Windows.Forms.Label بدلاً من استخدام الاسم Label فقط).

```
'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents Button1 As System.Windows.Forms.Button
```

يتم توصيف عناصر النموذج أثناء عملية تصميمه باستخدام الكلمة Friend والكلمة WithEvents حيث تشير الكلمة Friend كما تعلمنا في الفصل السادس إلى إمكانية التعامل مع العنصر خلال التجميع نفسه (Assembly)

التي تحتويه فقط ، أما الكلمة WithEvents فتشير إلى إمكانية إصدار أحداث (Events) من هذا العنصر ، ومن ثم سوف تعالج هذه الأحداث بالإجراءات التي تحتوي على العبارة "Handles controlName.eventName" ، فعلى سبيل المثال ربما يصدر الحدث Click من العنصر Button1 ، وبما أنه قد تم توصيف العنصر Button1 بالكلمة WithEvents فهذا يعني أن الضغط على هذا الزر سوف يستدعي الإجراءات الذي يشمل على العبارة Handles Button1.Click حيث نطلق على مثل هذه الإجراءات اسم معالجة الأحداث (Event Handlers).

تعرف باقي الأوامر الإجراء InitializeComponents() الذي يعرف كائناً من الصنف Label وكائناً من الصنف Button ، ثم يستدعي الإجراء SuspendLayout() الذي يوقف إصدار الأحداث مؤقتاً أثناء ضبط خصائص العناصر المختلفة خلال عملية التصميم (مثل الحجم Size والموقع Location) ، أما الأمر المحاط بعلامات الزاوية والمتواجد في تعريف رأس الإجراء فيقوم بإخبار برنامج مصحح الأخطاء (Debugger) بعدم الوقوف عند هذا الإجراء خلال تنفيذ البرنامج وأثناء عملية تصحيح الأخطاء .تذكر عزيزي القارئ أن الكلمة المحجوزة Me تشير إلى الكائن الحالي (الجاري تنفيذ أوامره الآن - الكائن FormDemo) ، كما يجب أن تلاحظ استخدام الرمز "\_" (Underscore) والذي يفيد أن الأمر مازال مستمراً في السطر التالي.

```
<System.Diagnostics.DebuggerStepThrough() > _
Private Sub InitializeComponent()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.Button1 = New System.Windows.Forms.Button()
    Me.SuspendLayout()

```

تسند الأوامر التالية خصائص عناصر النموذج وفقاً للقيم التي قمنا بإسنادها داخل نافذة الخصائص ، فعلى سبيل المثال تسند هذه الأوامر الصفة Font والصفة Foreground Color والصفة Location والصفة Name والصفة Size والصفة Tab Index والصفة Text والصفة Alignment. تشير الصفة Tab Index إلى ترتيب التعامل مع عناصر النموذج عند التنقل بينها أثناء تنفيذ البرنامج باستخدام المفتاح Tab حيث تأخذ الصفة Tab Index القيمة المناظرة لترتيب وضع العناصر على النموذج والتي تبدأ بالقيمة صفر ، ولكن يمكنك تغيير هذا الترتيب بتغيير قيم هذه الصفة لعناصر النموذج داخل نافذة الخصائص.

☞ لاحظ أنه يمكنك أيضاً تغيير ترتيب التنقل بين عناصر النموذج بالمفتاح Tab من القائمة View التي تسند قيم الصفة Tab Index طبقاً لترتيب اختيارها.

```

'
'Label1
'
Me.Label1.Font = New System.Drawing.Font("Times New Roman", _
    18.0!, System.Drawing.FontStyle.Regular, _
    System.Drawing.GraphicsUnit.Point, _ CType(0, Byte))
Me.Label1.ForeColor = System.Drawing.Color.Red
Me.Label1.Location = New System.Drawing.Point(48, 32)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(272, 48)
Me.Label1.TabIndex = 0
Me.Label1.Text = "Welcome to VB .NET"
Me.Label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter

```

تسند بقية الأوامر قيم صفات الكائن Button وكائن الصنف FormDemo حيث يقوم كل أمر بتنفيذ مهمة تناظر أحد الأحداث التي قمت بها أثناء تصميم النموذج مستعيناً بأداة صندوق الأدوات (Toolbox) أو بنافذة الخصائص. يقوم الإجراء ResumeLayout() بتطبيق أحداث التنسيق المعلقة التي تم إيقافها مؤقتاً بواسطة الإجراء SuspendLayout() ، وأخيراً تشير التعليمة #EndRegion إلى نهاية الأوامر التي تم إنتاجها تلقائياً بواسطة أداة مصمم نماذج الويندوز.

```

'
'Button1
'
Me.Button1.BackColor = System.Drawing.Color.FromArgb _
    (CType(255, Byte), CType(255, Byte), CType(128, Byte))
Me.Button1.Location = New System.Drawing.Point(144, 96)
Me.Button1.Name = "Button1"
Me.Button1.TabIndex = 1
Me.Button1.Text = "Push Me"
'
'Form1
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.BackColor = System.Drawing.Color.White
Me.ClientSize = New System.Drawing.Size(368, 325)
Me.Controls.AddRange(New System.Windows.Forms.Control() _
    (Me.Button1, Me.Label1))
Me.Name = "Form1"
Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
Me.Text = "Form Demo"
Me.ResumeLayout(False)

End Sub

#End Region

```

تعرف أوامر البرنامج الأخيرة إجراء معالجة الحدث Click الذي قمنا بتطويره أثناء الفصل الثاني والذي يستدعى بواسطة الزر Button1.

```
Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles _ Button1.Click
    MessageBox.Show("Thank you for using VB .NET")
End Sub
```

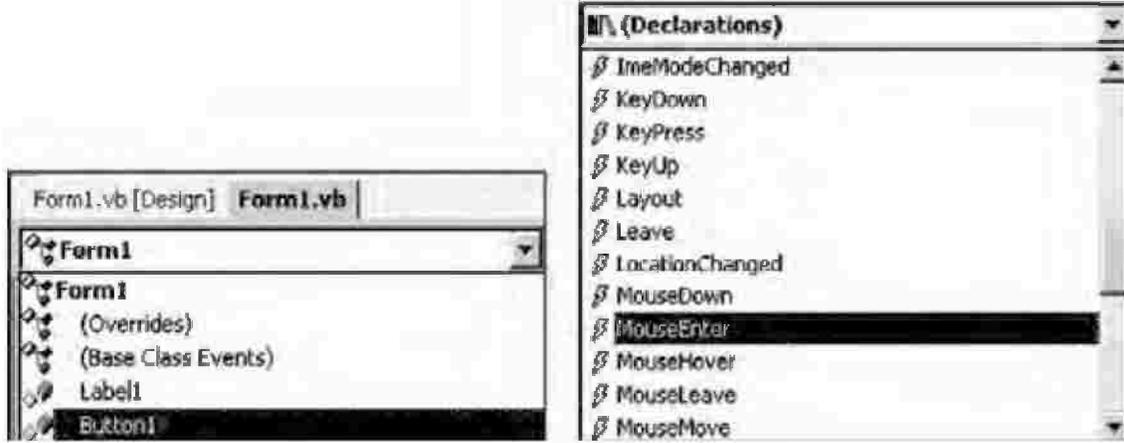
عزيزي القارئ لاحظ للمرة الثانية أن رأس الإجراء يحتوي على العبارة Handles Button1.Click، ولاحظ أيضاً الاسم الافتراضي للإجراء Button1\_Click() الذي يستجيب بإظهار رسالة على الشاشة تحتوي على الجملة "Thank you for using VB .NET".

### معالجة أحداث لغة فيجوال بيسك .نت

#### Handling VB .NET Events

عندما نطور برنامج ويندوز يجب علينا إضافة إجراءات معالجة الأحداث التي تحدث التفاعل بين البرنامج والمستخدم كما فعلنا في الفصل الثاني بإضافة الإجراء Button1\_Click() الذي يتم استدعاؤه تلقائياً عند الضغط على الزر "Push Me"، ولكي نضيف إجراء معالجة الحدث Click نقوم بالضغط المزدوج على الزر Button1، فعندئذ يتم فتح نافذة محرر الأوامر وإضافة إجراء Button1\_Click() الفارغ تلقائياً، ومن ثم نقوم بإضافة أمر إظهار رسالة للمستخدم، وبشكل عام عندما نضغط ضغطاً مزدوجاً على أي عنصر داخل النموذج أثناء مرحلة التصميم، تفتح لغة فيجوال بيسك .نت نافذة محرر البرنامج، ثم تضيف رأس إجراء معالجة أشهر حدث يرتبط مع هذا العنصر، وذلك مثلما حدث مع الزر Button1 .

تستطيع عناصر واجهة المستخدم أن تكون مصدراً للعديد من الإجراءات الأخرى والتي تختلف عن الأحداث الأكثر استخداماً معها، فعلى سبيل المثال يكفي أن تجعل برنامج FormDemo يستجيب للحدث MouseEnter الخاص بالزر Button1 وذلك بعرض الرسالة "MouseEnter event detected" (يحدث هذا الحدث عندما يدخل رأس مؤشر الفأرة داخل حدود الزر Button1)، وكما هو واضح من الشكل رقم (١٠٧) يمكن إضافة إجراء معالجة هذا الحدث للبرنامج FormDemo وذلك بالضغط على المفتاح F7 لكي تفتح نافذة محرر البرنامج، ثم اختيار العنصر Button1 من قائمة Class Name، ثم اختيار الحدث MouseEnter من قائمة Method Name، ثم إضافة الأمر .MessageBox.Show()



الشكل رقم (١٠٧). تعريف إجراء معالجة حدث.

### التعامل مع عناصر إضافية في واجهة المستخدم الرسومية

#### Working with Additional GUI Controls

سوف نتعلم في هذه الفقرة كيف نتعامل مع عناصر واجهة استخدام أخرى مثل زر الاختيار Radio Button ومسدوق التأشير Check Box ومسدوق القوائم List Boxes وعنصر التبويب Tab Control وعنصر الترميم Calendar، ولكن قبل أن نتعرض لهله العناصر يجب أن نتعلم مبادئ تصميم النماذج، مع العلم أن معظم التطبيقات العملية تتكون من العديد من النماذج حيث يطلق على واجهة البرنامج شعار "Look and Feel" الذي يعني مبدأ توحيد التصميم والمظهر لنماذج التطبيق مما يعطي قدراً من التوافق والألفة للمستخدم عندما يتجول بين نماذج التطبيق الواحد، ومن ثم يؤدي إلى الإحساس بالراحة لدى المستخدم. فعلى سبيل المثال إن موقع أزرار النماذج وألوانها التي تؤدي نفس المهمة يجب أن تكون موحدة. إن اختيار الألوان وأنواع خطوط عناصر النماذج يجب أيضاً أن تكون متوافقة خلال التطبيق حيث إن حصر عدد الألوان والخطوط المستخدمة في عدد قليل يحسن الصورة المرئية للبرنامج مما يؤدي إلى عدم تشتت المستخدم عند استخدام البرنامج.

يعد سهولة استخدام البرنامج (Base of Use) الهدف الثاني للتصميم الجيد حيث يجب أن يكون هدف كل عنصر وتنسيقه داخل النموذج واضحاً للمستخدم، فمثلاً يجب أن تحدد العناوين المتواجدة على النموذج كل من البيانات المطلوب إدخالها وكيفية التفاعل مع النظام، هذا بالإضافة إلى وضع العناصر بشكل منطقي وتجميعها على النموذج حيث يجب أن يظهر النموذج مليئاً بالعناصر.

كما أن إعلام المستخدم بحدوث بعض الأحداث الهامة مثل إضافة سجل إلى قاعدة البيانات ومثل حدوث خطأ عند تعديل سجل داخل قاعدة البيانات يعد هدفاً ثالثاً للتصميم الجيد لواجهة النظام حيث يجب أن تظهر تلك الرسائل الإخبارية بشكل متوافق (كإصدار صوت مناسب مع الرسالة إذا احتاج الأمر)، وأخيراً يجب علينا أن

نصمم واجهة التطبيق بأسلوب يؤدي إلى تقليل استخدام لوحة المفاتيح ، ومن ثم إلى تقليل فرصة حدوث أخطاء عند إدخال البيانات (ومما لا شك فيه تجعل حياة المستخدم أكثر يسراً).

ولكي تحقق هذه الأهداف عزيزي القارئ يجب عليك التروي والتخطيط الجيد ، فمن الشائع أن تقوم مجموعة البرمجة بوضع مجموعة من المقاييس لتصميم النماذج خارج إطار المشروع والتي يجب أن تتبع بصرامة أثناء عملية تطوير النظام.

كما يتبع المبرمجون المحترفون نظاماً محدداً عند تسمية متغيرات البرنامج ، فعلى الرغم من أن المستخدم النهائي لا يعلم شيئاً عن هذه المتغيرات ، فإن التسمية الجيدة للمتغيرات تحسن قراءة أوامر البرنامج وتسهل عملية صيانة البرنامج. تحتوي لغة فيجوال بيسك .نت على العديد من عناصر واجهة المستخدم التي تسند إليها أسماء افتراضية عند إدراجها داخل النموذج أثناء عملية التصميم ، فمثلاً تسند لغة فيجوال بيسك .نت الاسم Button1 والاسم Button2 للأزرار ، كما تسند الاسم Label1 والاسم Label2 للعناوين. يهتم المبرمج المحترف بإعادة تسمية هذه العناصر بأسماء ذات معنى ، ولتغيير أسماء هذه العناصر يجب تغيير قيمة الخاصية Name داخل نافذة الخصائص حيث يجب أن يحتوي اسم العنصر على نوع العنصر (هل هو زر Button أم عنوان Label ، وهكذا) والغرض من العنصر ، فعلى سبيل المثال يحتوي البرنامج TesterFive (المقدم في الفصل السادس) على الزر btnCenter والزر btnDisplay والزر btnClose ، يشير الجزء "btn" إلى نوع العنصر (في هذه الحالة العنصر ينتمي إلى الصنف Button). سوف نستخدم هذا الأسلوب في تسمية بقية عناصر واجهة المستخدم خلال هذا الكتاب ، ويوضح الجدول رقم (١٠،٢) أسلوب تسمية عناصر واجهة المستخدم ومكوناتها بناء على نوعها.

الجدول رقم (١٠،٢). تسمية عناصر مختارة من واجهة المستخدم الرسومية.

العنصر	بداية الاسم
Button	Btn
CheckBox	Chk
CheckedListBox	Chklst
ComboBox	Cmb
DateTimePicker	Dtp
Form	Frm
GroupBox	Grp
Label	Lbl
ListBox	Lst
MainMenu	Mnu
MenuItem	Mnuitm
Panel	Pnl
RadioButton	Rad
TabControl	Tab
TabPage	Tabpg
TextBox	Txt
TreeNode	Trn
TreeView	Trv

## العودة إلى النماذج (Forms) والأزرار (Buttons) والعناوين (Labels)

## Reviewing Forms, Buttons, and Labels

قدم لنا برنامج FormDemo العديد من الإجراءات والخصائص الشائعة للأزرار والنماذج والعناوين ، ولكن تذكر أن هذه العناصر وغيرها ما هي إلا أصناف فرعية للصنف Control ؛ ولذلك فإنها ترث جميع الصفات والإجراءات والأحداث المعرفة داخل هذا الصنف. يوضح الجدول رقم (١٠,٣) مجموعة مختارة من الصفات والأحداث والإجراءات الشائعة والمعرفة داخل الصنف Control.

الجدول رقم (١٠,٣). مجموعة مختارة من صفات الصنف Control وإجراءاته.

الوصف	الصفة
تحدد هذه الصفة وضع تنسيق العنصر داخل العنصر الأب الذي يحتويه (الوضع الافتراضي هو الركن الأيسر العلوي)	Anchor
تحدد لون خلفية العنصر	BackColor
تحدد مساحة العنصر	ClientSize
تحدد مجموعة العناصر المتواجدة داخل العنصر الأب	Controls
تحدد إلى أي جانب من جوانب العنصر الأب يلتصق العنصر	Dock
تحدد هل يمكن التعامل مع العنصر أم لا	Enabled
تحدد هل العنصر نشط للكتابة أم لا	Focused
تحدد نوع خط نص العنصر	Font
تحدد لون واجهة العنصر	ForeColor
تحدد موقع العنصر	Location
تحدد اسم العنصر	Name
تحدد حجم العنصر (طول وعرض)	Size
تحدد فهرس العنصر داخل العنصر الأب	TabIndex
تحدد النص المصاحب للعنصر	Text
تحدد العنصر مرئي أم لا	Visible
يجرر الموارد المخصصة للعنصر	Dispose
يجعل العنصر نشطاً لاستقبال الكتابة من المستخدم	Focus
يخفي العنصر	Hide
يسترجع أحداث عناصر التصميم المعلق بواسطة الإجراء SuspendLayout	ResumeLayout
يظهر العنصر	Show
يعلق أحداث عناصر التصميم مؤقتاً	SuspendLayout

كما يوضح الجدول رقم (١٠,٤) والجدول رقم (١٠,٥) والجدول رقم (١٠,٦) مجموعة مختارة من الصفات والإجراءات والأحداث الشائعة والمعرفة داخل الصنف Form والصنف Button والصنف Label على التوالي.

الجدول رقم (١٠,٤). مجموعة مختارة من صفات الصنف Form وإجراءاته وأحداثه.

الوصف	الصفة
تحدد الزر الذي سيضغط عند الضغط على المفتاح الإدخال	AcceptButton
تحدد قيمة ضبط حجم النموذج تلقائياً طبقاً لأحجام الخطوط	AutoScale
تحدد قيمة الحجم الأساسي المستخدم بواسطة الصفة AutoScale	AutoScaleBaseSize
تحدد الزر الذي سيضغط عند الضغط على المفتاح Esc	CancelButton
تحدد القائمة الرئيسة للمصاحبة للنموذج	Menu
تحدد نص عنوان النموذج	Text
الوصف	الإجراء
يقوم بغلاق النموذج وتحرير جميع موارده	Close
الوصف	الحادث
يحدث هذا الحادث قبل إظهار النموذج مباشرة	Load

الجدول رقم (١٠,٥). مجموعة مختارة من صفات الصنف Button وأحداثه.

الوصف	الصفة
تحدد نص عنوان الزر	Text
تحدد تنسيق نص عنوان الزر	TextAlign
الوصف	الحادث
يحدث عندما يضغط المستخدم على الزر	Click

الجدول رقم (١٠,٦). مجموعة مختارة من صفات الصنف Label وأحداثه.

الوصف	الصفة
تحدد نص عنوان العنصر Label	Text
تحدد تنسيق نص العنوان	TextAlign
الوصف	الحادث
يحدث عندما يضغط المستخدم على العنوان	Click

تحتوي جميع عناصر واجهة الاستخدام الرسومية الخاصة بلغة فيجوال بيسك نت على العديد من الإجراءات والصفات الهامة التي يجب الاطلاع عليها والتي لا يمكن أن نذكرها كاملة داخل هذا الفصل ، ولذلك يمكن أن نستعين بأداة المساعدة Help لكي نعلم الكثير عن أمتاف واجهة المستخدم الرسومية (GUI Classes) ، فعلى سبيل المثال لكي نتعرض لمعلومات حول أعضاء الصنف Button ، اتبع الخطوات التالية :

- ١- اختر الأمر Index من القائمة الفرعية Help ، فستتخذ تظهر نافذة الفهرس Index .
- ٢- اختر قائمة Filtered by من نافذة الفهرس ، ثم اختر Visual Basic and Related .
- ٣- اكتب الكلمة Button داخل صندوق النص Look for .
- ٤- تصفح نافذة النتائج حتى تروى العبارة "Button class (System.Windows.Forms)".
- ٥- اختر الخيار members اللذي يظهر مباشرة أسفل العبارة السابقة ، فستتخذ تظهر نافذة الشكل رقم (١٠٨)
- ٦- لتعلمك الكثير عن صفات الصنف Button وإجراءاته وأحداثه .
- ٦- أخلق نافذة أعضاء الصنف Button .



الشكل رقم (١٠٨). معلومات مساعدة عن أعضاء الصنف Button.

سوف نتعلم الكثير عن عناصر واجهة استخدام أخرى داخل هذا الفصل مثل الصفات الرئيسية، وإجراءات كل عنصر، وكيفية استخدام كل عنصر وذلك من خلال مجموعة من الأمثلة البسيطة، أما الفصل الحادي عشر فسوف يوضح كيفية استخدام هذه العناصر داخل نظام شركة برادشو مارينا.

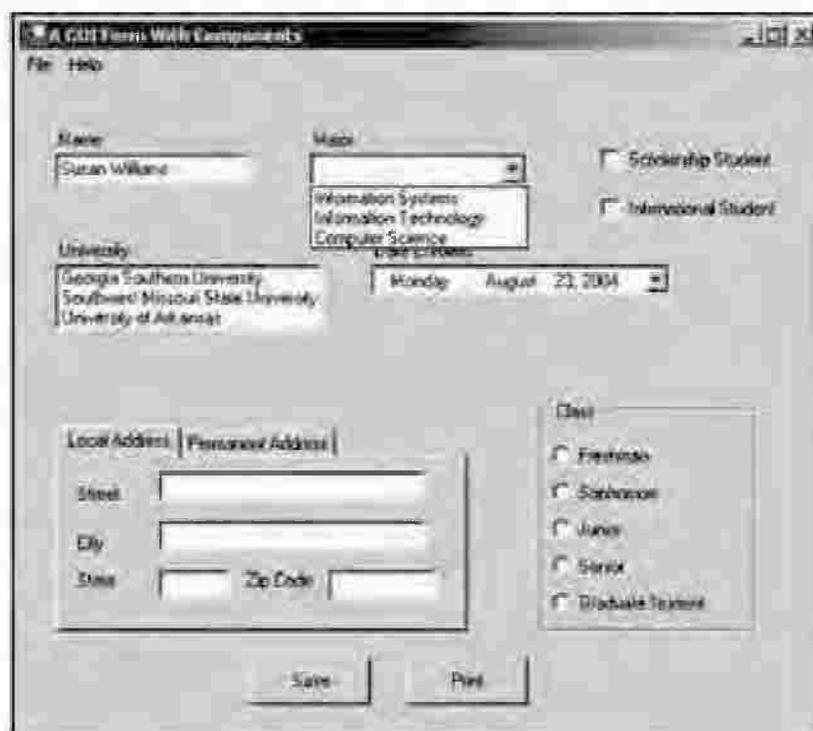
### استخدام صناديق النصوص (Text Boxes) والقوائم المنسدلة (Combo Boxes)

#### Using Text Boxes and Combo Boxes

تستخدم الصناديق النصية (Text Boxes) لاستقبال نصوص من المستخدم أو لإظهار معلومات نصية له، أما القائمة المنسدلة (Combo Box) فتقوم بدور صندوق النص نفسه وتزيد عليه إمكانية اختيار عنصر من مجموعة من العناصر المعروفة مسبقاً، وتأخذ القائمة المنسدلة شكل صندوق النص نفسه ولكن تحتوي على سهم في أقصى اليمين، وعندما يضغط المستخدم على هذا السهم تنسدل منه قائمة من العناصر. تقدم لغة فيجوال بيسك. نت كلاً من الصنف TextBox والصنف ComboBox لإنشاء عنصر صندوق النص وعنصر القائمة المنسدلة على التوالي.

إذا كان لدينا علم مسبق أن المستخدم سوف يدخل قيمة واحدة داخل صندوق النص من مجموعة محددة من القيم، فعندئذ يجب علينا استخدام عنصر القائمة المنسدلة بدلاً من صندوق النص، كما يمكن للمستخدم أن يكتب قيمة داخل القائمة المنسدلة إذا لم تكن موجودة بالفعل، ومن ثم فإن عنصر القائمة المنسدلة يقلل إدخال البيانات من لوحة المفاتيح، وبذلك نجعل البرنامج أكثر سهولة ونقلل احتمالات حدوث الأخطاء. يوضح الشكل رقم (١٠،٩) نموذجاً يحتوي على العديد من عناصر واجهة المستخدم الرسومية، كما يحتوي على عنصر القائمة المنسدلة Major التي يستخدم لإدخال تخصص الطالب.

وكما فعلنا سابقاً يمكنك إدراج قائمة منسدلة أو صندوق نص داخل نموذج وذلك باختياره من صندوق الأدوات وجره لوضعه داخل النموذج ثم ضبط خصائصه مستخدماً نافذة الخصائص، ويوضح الجدول رقم (١٠،٧) والجدول رقم (١٠،٨) أهم خصائص الصنف TextBox والصنف ComboBox وإجراءاتهما وأحداثهما على التوالي.



الشكل رقم (٩، ١٠). اختيار عنصر من قائمة متسلسلة.

الجدول رقم (٧، ١٠). مجموعة متعارفا من صفات المتبب Text وواجهته وأمتاب.

الوصف	الخاصة
تحدد هل ضغط مفتاح الإدخال وهي سطرًا جديدًا أم ضغط لزر الإقراض	AcceptReturn
تحدد هل ضغط مفتاح الجدولة (Tab) يكتب لثاني مسافات أم ضغط الزر الإقراض	AcceptTab
تحدد أقصى عدد حروف يمكن كتابته داخل صندوق النص	MaxLength
تحدد هل صندوق النص يمتهي على سطر واحد أو أكثر	MultiLine
تحدد الحرف الذي سيكتب عند كتابة حروف كلمة المرور	PasswordChar
تحدد هل صندوق النص للقراءة فقط أم لا	ReadOnly
تحدد نص عنوان صندوق النص	Text
تحدد تسيق نص عنوان صندوق النص	TextAlign
الوصف	الإجراء
يأخذ نص جديد بالنص المتواجد داخل الصندوق النص.	AppendText
يقوم بمسح النص المتواجد داخل الصندوق النص.	Clear
الوصف	الحدث
يحدث عندما يغير نص المتواجد داخل صندوق النص.	TextChanged

الجدول رقم (٨، ١٠). مجموعة مختارة من صفات الصنف ComboBox وأحداثه.

الوصف	الصفة
تحدد العناصر المتواجدة داخل القائمة المنسدلة	Items
تحدد أقصى عدد عناصر يمكن أن يظهر عند انسدال القائمة المنسدلة	MaxDropDownItems
تحتوي على قيمة فهرس العنصر المختار حالياً من القائمة المنسدلة	SelectedIndex
تحتوي على العنصر المختار حالياً من القائمة المنسدلة	SelectedItem
تحدد هل عناصر القائمة المنسدلة مرتب أم لا	Sorted
تحدد نص عنوان القائمة المنسدلة	Text
الوصف	الحدث
يحدث عندما يتغير العنصر المختار حالياً من القائمة المنسدلة	SelectedIndexChanged

### استخدام صناديق التأشير (Check Boxes) وأزرار الاختيار (Radio Buttons)

#### Using Check Boxes and Radio Buttons

تقدم صناديق التأشير (Check Boxes) وأزرار الاختيار (Radio Buttons) إمكانية اختيار عنصر من عدة عناصر حيث تأخذ كل من صناديق التأشير وأزرار الاختيار حالة من اثنين إما مؤشر (Checked) أو غير مؤشر (Not Checked)، ومثل الصنف Button، فإن كلا من الصنف CheckBox والصنف RadioButton تمثل أصنافاً فرعية من الصنف ButtonBase، ولكن يحتويان على إجراءات أخرى التي تمكن المبرمج من تغيير أو معرفة حالة كل من صندوق التأشير وزر الاختيار، مع العلم أن التأشير وعدم التأشير لتلك العناصر يصدر أحداثاً.

يظهر صندوق التأشير على شكل مربع أبيض صغير، كما يحتوي على عنوان يوضح الغرض منه، ويمكن إسناد قيمة له بواسطة الخاصية Text، وعندما يتم تأشيرها تظهر علامة وسط الصندوق. تستخدم صناديق التأشير عادة للحصول على معلومات مرتبطة ببعضها، فعلى سبيل المثال يمكن استخدام صناديق التأشير في نظام برادشو مارينا لمعرفة هل رصيف ما يحتوي على كهرباء أم لا أو يحتوي على مياه أم لا، ومن ثم يستطيع المستخدم التأشير على الصندوقين معاً أو التأشير على أحدهما فقط أو عدم التأشير. تعمل صناديق التأشير بشكل مستقل بمعنى أن اختيار صندوق تأشير ما لا يؤثر على صندوق آخر؛ ولذلك يمكن اختيار أي أو جميع الصناديق في وقت واحد أو حتى عدم اختيارها جميعاً.

تشابه أزرار الاختيار مع صناديق التأشير ولكن تختلف في بعض الأمور الهامة حيث يظهر زر الاختيار على شكل دائرة صغيرة بيضاء، ويحتوي على عنوان يوضح الغرض منه، وعندما يتم اختيار زر الاختيار تظهر نقطة سوداء داخل الدائرة (مثل صناديق التأشير). إن استخدام مجموعة من أزرار الاختيار يعني ارتباط مجموعة

اختيارات ، وعلى أي حال إن أهم اختلاف بينها هو أن أزرار الاختيار تستخدم عندما نريد أن يكون هناك زر واحد مختار فقط ؛ ولذلك فإن اختيار أحد الأزرار ينفي اختيار الزر الحالي ، فعلى سبيل المثال يمكن استخدام أزرار الاختيار في نظام برادشو مارينا لتمثيل نوع وقود محرك المركب الآلي الذي يأخذ القيمة "ديزل" أو القيمة "بنزين". يوضح الجدول رقم (١٠,٩) والجدول رقم (١٠,١٠) أهم خصائص هذين العنصرين وإجراءاتهما وأحداثهما.

الجدول رقم (١٠,٩). مجموعة مختارة من صفات الصنف CheckBox وأحداثه.

الوصف	الصفة
تحدد هل صندوق التأشير مختار حالياً أم لا	Checked
تحدد حالة صندوق التأشير مختار حالياً (Checked)، أو غير مختار (Unchecked)، أو بين الحالتين (Checked & Shaded)	CheckState
تحدد نص عنوان صندوق التأشير	Text
الوصف	الحديث
يحدث عندما تتغير حالة صندوق التأشير	CheckedChanged

الجدول رقم (١٠,١٠). مجموعة مختارة من صفات الصنف RadioButton وأحداثه.

الوصف	الصفة
تحدد هل زر الاختيار مختار حالياً أم لا	Checked
تحدد نص عنوان زر الاختيار	Text
الوصف	الحديث
يحدث عندما تتغير حالة زر الاختيار	CheckedChanged

## استخدام صناديق التجميع والإطارات

### Using Group Boxes and Panels

تستخدم كل من صناديق التجميع (Group Boxes) والإطارات (Panels) لتنظيم مجموعة من العناصر المرتبطة داخل إطار وتجميعها ؛ لذلك عندما يتم إنشاء صندوق تجميع (كائن من الصنف Group Box) أو إطار (كائن من الصنف Panel) ، وإدراج عناصر لواجهة الاستخدام (مثل الأزرار والعناوين) داخل حدوده ، فإن جميع هذه العناصر ستحول معاً كوحدة واحدة عند نقل صندوق التجميع (أو الإطار) من مكان إلى آخر داخل النموذج. يحتوي صندوق التجميع على حدود تلقائياً ، أما الإطار فلا يحتوي على حدود بشكل تلقائي ، ولكن يمكن إظهار حدود له عن طريق الخاصية BorderStyle. ربما تحتوي الإطارات على أزرار تصفح أفقية أو رأسية ولكن لا تحتوي على عنوان. أما صناديق

التجميع فلا تحتوي على أزرار تصفح وعادة ما تحتوي على عنوان الذي يمكن تغيير قيمته عن طريق الخاصية Text. وأخيراً يجب أن تعلم أنه يمكن وضع صندوق التجميع كعنصر داخل الإطار والعكس صحيح. وبالرغم أن كلاً من صندوق التجميع والإطار يمكن أن يحتوي على أي عنصر من عناصر واجهة المستخدم، فإنه عادة ما يتم استخدامهما لتجميع مجموعة من أزرار الاختيار، حيث يمكن استخدام أكثر من صندوق تجميع (أو إطار) داخل النموذج الواحد لإظهار عدة مجموعات مختلفة من أزرار الاختيار. وبذلك فإنه يتم اختيار عنصر واحد من كل مجموعة على حده. ربما يصدر عن كل من صناديق التجميع والإطارات أحداث لا نهتم عادة أن نستجيب لها بتعريف إجراءات معالجة. يوضح كل من الجدول رقم (١٠،١١) والجدول رقم (١٠،١٢) أهم خصائص هذين العنصرين وإجراءاتهما وأحداثهما.

الجدول رقم (١٠،١١). مجموعة مختارة من صفات الصنف groupBox.

الوصف	الصفة
تحتوي على مجموعة عناصر واجهة المستخدم المتواجدة داخل الإطار	Controls
تحدد نص عنوان صندوق التجميع	Text

الجدول رقم (١٠،١٢). مجموعة مختارة من صفات الصنف Panel وإجراءاته وأحداثه.

الوصف	الصفة
تحدد هل الإطار يسمح للمستخدم باستخدام أعمدة تصفح لإظهار عناصر الإطار	AutoScroll
تحدد شكل حدود الإطار	BorderStyle
تحتوي على مجموعة عناصر واجهة المستخدم المتواجدة داخل الإطار	Controls

### استخدام قوائم الاختيار وقوائم الاختيار بالتأشير

#### Using List Boxes and Checked List Boxes

تقدم كل من قوائم الاختيار (List Boxes) وقوائم الاختيار بالتأشير (Checked List Boxes) إمكانية اختيار عنصر أو أكثر من مجموعة قيم معرفة مسبقاً، وعند إنشاء قائمة اختيار يتم إنشاء كائن من الصنف ListBox الذي يستخدم لاختيار عنصر واحد فقط من عدة اختيارات. ويمكن تغيير إمكانية اختيار أكثر من عنصر بواسطة ضبط الخاصية SelectionMode. أما قائمة الاختيار بالتأشير فتعمل مثل قائمة الاختيار، ولكن تضيف صندوق تأشير على يسار كل اختيار. وعند إنشاء هذه القائمة يتم إنشاء كائن من الصنف CheckedListBox الذي يستخدم لاختيار أكثر من عنصر في آن واحد. وعند اختيار احد عناصر القائمة تظهر علامة التأشير داخل صندوق التأشير.

تستخدم قائمة الاختيار أو قائمة الاختيار بالتأشير عندما نريد أن نجبر المستخدم أن يختار عنصراً أو أكثر من مجموعة معرفة مسبقاً دون توفير إمكانية إضافة عنصر جديد. فعلى سبيل المثال يمكن استخدام قائمة الاختيار في نظام برادشو مارينا لكي نوجه المستخدم أن يختار مرسى من قائمة المراسي المتاحة فقط. يلخص الجدول رقم (١٠،١٣) والجدول رقم (١٠،١٤) أهم الخصائص والإجراءات والأحداث المعرفة داخل كل من قائمة الاختيار وقائمة الاختيار بالتأشير على التوالي.

الجدول رقم (١٠،١٣). مجموعة مختارة من صفات الصنف `ListBox` وإجراءاته وأحداثه.

الوصف	الصفة
تحتوي على العناصر المتواجدة داخل قائمة الاختيار	Items
تحدد هل قائمة الاختيار تدعم ظهور أكثر من عمود أم لا	MultiColumn
تحتوي على قيمة فهرس العنصر المختار حالياً من قائمة الاختيار	SelectedIndex
تحتوي على قيم فهرس العناصر المختارة حالياً من قائمة الاختيار	SelectedIndices
تحتوي على العنصر المختار حالياً من قائمة الاختيار	SelectedItem
تحتوي على العناصر المختارة حالياً من قائمة الاختيار	SelectedItems
تحدد هل يستطيع المستخدم أن يختار اختياراً واحداً فقط أو يختار أكثر من اختيار	SelectionMode
تحدد هل عناصر قائمة الاختيار مرتبة أم لا	Sorted
الوصف	الإجراء
إلغاء اختيار عناصر القائمة المختارة	ClearSelected
تحديد حالة عنصر ما، هل هو مختار بالفعل أم لا	GetSelected
اختيار (أو إلغاء الاختيار) عنصر ما	SetSelected
الوصف	الحدث
يحدث عندما يتغير العنصر المختار حالياً	SelectedIndexChanged

الجدول رقم (١٠,١٤). مجموعة مختارة من صفات الصنف `CheckedListBox` وإجراءاته وأحداثه.

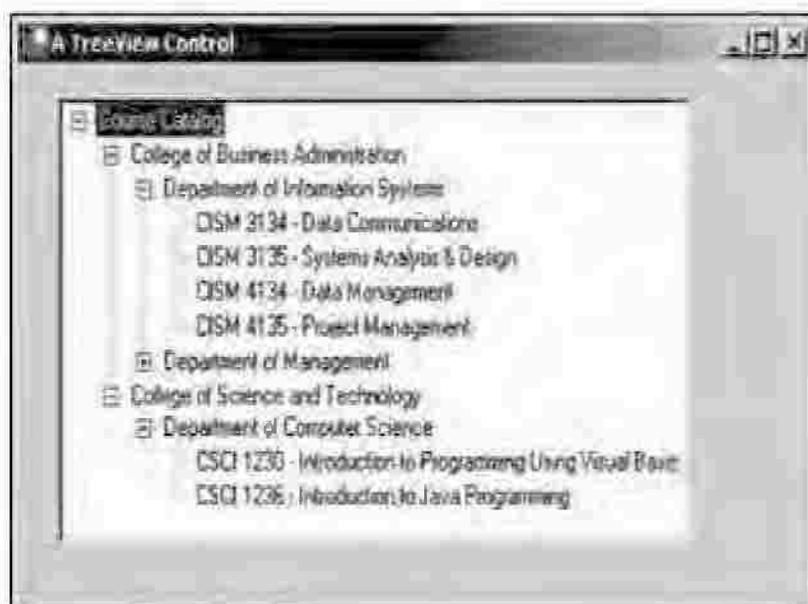
الوصف	الصفة
تحدد قيم فهارس العناصر المختارة حالياً من قائمة الاختيار بالتأشير	<code>CheckedIndices</code>
تحدد العناصر المختارة حالياً من قائمة الاختيار بالتأشير	<code>SelectedItems</code>
تحديد حالة عنصر ما، هل هو مختار بالفعل أم لا	<code>GetItemChecked</code>
الوصف	الإجراء
معرفة حالة عنصر ما	<code>GetItemCheckedState</code>
اختيار عنصر ما عند فهرس معين	<code>SetItemChecked</code>
تغيير حالة عنصر ما عند فهرس معين	<code>SetItemCheckedState</code>
الوصف	الحدث
يحدث عندما يتغير حالة عنصر ما	<code>ItemCheck</code>
يحدث عندما يتغير العنصر المختار حالياً	<code>SelectedIndexChanged</code>

### استخدام الشجرة المرئية والعقد الشجرية

#### Using Tree Views and Tree Nodes

تظهر الشجرة المرئية (Tree View) مجموعة من العناصر في شكل شجري، حيث يمثل كل عنصر عقدة في الشجرة. وتمثل الشجرة المرئية كائناً من الصنف `TreeView`. كما يمثل كل عنصر داخل الشجرة كائن من الصنف `TreeNode`. وكما هو واضح من الشكل رقم (١٠,١٠) أن الشجرة المرئية والعقد المصاحبة لها تظهر في شكل شجري. نستطيع أن نستخدم عنصر الشجرة المرئية في نظام برادشو مارينا لعرض معلومات عن الأرصفة والمراسي التي تنتمي إليها.

يلخص كل من الجدول رقم (١٠,١٥) والجدول رقم (١٠,١٦) أهم الخصائص والإجراءات والأحداث المعرفة داخل هذين العنصرين.



الشكل رقم (١٠,١٠). عنصر الشجرة المرئية.

الجدول رقم (١٠,١٥). مجموعة مخطوطات من صفات المصنف `TreeView` وإجراءاته وأحداثه.

الوصف	الصفة
تحديد هل تظهر مستطيق التأشير للحوار عند الشجرة أم لا	<code>CheckBox</code>
تحديد مساحة لتسوية كل مستوى في الشجرة	<code>Indent</code>
تحديد نصي على مجموعة من عند الشجرة	<code>Margin</code>
تحديد تقنية للمحاذاة سلباً	<code>SelectedNode</code>
تحديد ظهور المخطوط بين العقد أم لا	<code>ShowLines</code>
تحديد هل تظهر كلاً من علامة + وعلامة - للحوار عند الشجرة أم لا	<code>ShowPlusMinus</code>
تحديد ظهور المخطوط بين جزر الشجرة والعقد الأخرى	<code>ShowRootLines</code>
تحديد هل عند الشجرة مرئية أم لا	<code>Sorted</code>
الوصف	الإجراء
تكشف عند الشجرة	<code>CollapseAll</code>
توسع عند الشجرة	<code>ExpandAll</code>
استرجع عقدة عند اللمس عند	<code>GetNodeAt</code>
استرجع عدد عند الشجرة	<code>GetNodeCount</code>
الوصف	الحدث
يحدث بعد اختيار عقدة	<code>AfterSelect</code>

الجدول رقم (١٠,١٦) مجموعة مختارة من صفات الصنف `TreeNode` وإجراءاته.

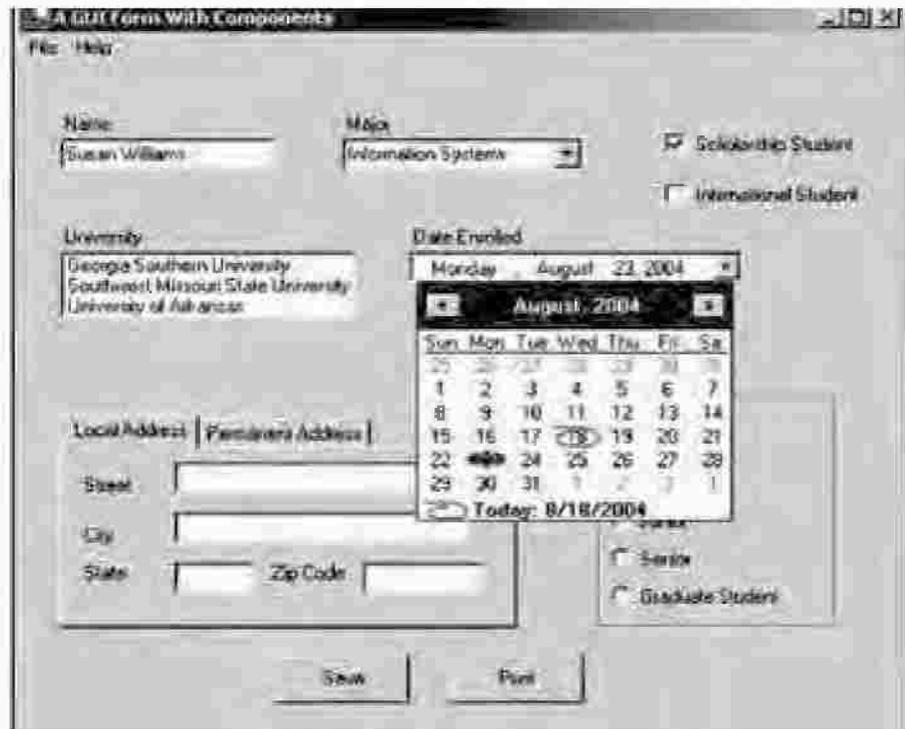
الوصف	الصفة
تحدد هل عقدة الشجرة مختارة حالياً أم لا	Checked
تعيد أول عقدة في الشجرة	FirstNode
تعيد المسار بداية من الجذر إلى العقدة الحالية	FullPath
تحدد موقع عقدة في الشجرة	Index
تعيد آخر عقدة في الشجرة	LastNode
تعيد العقدة التالية للعقدة الحالية في نفس المستوى	NextNode
تحتوي على مجموعة من عقد الشجرة المنتمة لعقدة ما	Nodes
تعيد العقدة ذات المستوى الأعلى (الأب)	Parent
تحدد العقدة المختارة حالياً	SelectedNode
تحدد نص عنوان عقدة ما	Text
الوصف	الإجراء
تكتمش عقدة الشجرة	Collapse
توسع عقدة الشجرة	Expand
توسع جميع عقد الشجرة	ExpandAll
تسترجع عدد عقد الأبناء	GetNodeCount
حذف العقدة الحالية	Remove

### استخدام أداة التاريخ والوقت

#### Using Date/Time Pickers

تمكن أداة التاريخ والوقت (Date/Time Picker) المستخدم من اختيار التاريخ والوقت من خلال إظهار نتيجة مرئية، وكذلك تستخدم لعرض التاريخ والوقت بأشكال مختلفة. يوضح الشكل رقم (١٠,١١) نتيجة الضغط على سهم أداة التاريخ والوقت لاختيار تاريخ تسجيل طالب (تم عرض هذا البرنامج سابقاً داخل هذا الفصل)، كما يمكنك استخدام هذه الأداة داخل نظام برادشو مارينا لتمكين المستخدم من إدخال تاريخ بداية العقد. عند إدراج أداة التاريخ والوقت داخل نموذج ما يتم إنشاء كائن من الصنف `DateTimePicker`.

يلخص الجدول رقم (١٠,١٧) أهم الصفات والإجراءات والأحداث الخاصة بهذه الأداة.



الشكل رقم (١٠،١٦). اعداد التاريخ من اداء التاريخ والوقت.

الجدول رقم (١٠،١٧). أهم صفات المصنف DateTimesFormat وإجراءاته وأحداثه.

الوصف	الصفة
تحديد نوع خط العنصر	CalendarFont
تحديد لون واجهة العنصر	CalendarForeColor
تحديد لون خلفية شهر العنصر	CalendarMonthBackground
تحديد لون خلفية عنوان العنصر	CalendarTitleBackColor
تحديد لون عنوان العنصر	CalendarTitleForeColor
تحديد أسلوب خاص تنسيق التاريخ والوقت	CustomFormat
تحديد أسلوب تنسيق التاريخ والوقت في العنصر	Format
تحديد أقصى تاريخ ووقت يمكن اختياره من العنصر	MaxDate
تحديد أقل تاريخ ووقت يمكن اختياره من العنصر	MinDate
الوصف	الإجراء
يعرف كيف يظهر المحرر نظمه، التاريخ والوقت	ShowOptions
تحديد قيمة التاريخ والوقت الحالي	Value
الوصف	الحدث
يحدث عندما تغير قيمة التاريخ والوقت الحالي	ValueChanged

## استخدام عناصر التبويب والصفحات الميوية

## Using Tab Controls and Tab Pages

يقدم عنصر التبويب (Tab Control) الذي هو كائن من الصنف TabControl وظائف عمل الصفحات الميوية (Tab Pages) حيث تمثل كل صفحة ميوية كائناً من الصنف TabPage (صنف فرعي من الصنف Panel). تستخدم الصفحات الميوية عندما يكون عدد العناصر المراد وضعها داخل نموذج كثيرة جداً بشكل يصعب وضعها داخل شاشة واحدة، ومن ثم يمكن تقسيم هذه العناصر تقسيماً منطقياً في مجموعات فرعية، ثم وضع كل مجموعة داخل صفحة وتخصيص مسطرة لكل صفحة، وعندئذ يتحول المستخدم من صفحة إلى أخرى بالضغط على التبويب الخاص بها. يمكنك استخدام عنصر التبويب في نظام برادشو مارينا لعرض صفحتين مبيتين لاستخلاص معلومات حول المراكب حيث تخصص صفحة المراكب الآلية (عدد المركبات، ونوع الوقود، إلخ) وتخصص الأخرى للمراكب الشراعية (عدد الأشرعة، وعمق المركب، ونوع المحرك). يلخص كل من الجدول رقم (١٠،١٨) والجدول رقم (١٠،١٩) أهم خصائص هذين العنصرين وأحداثهما.

الجدول رقم (١٠،١٨). مجموعة مختارة من صفات الصنف TabControl وأحداثه.

الوصف	الصفة
تحدد موضع تنسيق العنصر الميوي	Alignment
تحدد هل يمكن إظهار أكثر من صف من التبيويات أم لا	Multiline
تحتوي على قيمة فهرس عنصر تبويب الصفحة المختارة حالياً من العنصر الميوي	SelectedIndex
تحتوي على عنصر تبويب الصفحة المختارة حالياً	SelectedTab
تحدد هل تظهر نصيحة عند نقل مؤشر الفأرة لتبيويات الصفحات أم لا	ShowToolTips
تحدد عدد تبويبات الصفحات	TabCount
تحدد ترتيب كل صفحة داخل العنصر الميوي	TabIndex
مجموعة من تبويبات الصفحات المتواجدة داخل العنصر الميوي	TabPage
الوصف	الحدث
يحدث عندما يتغير بتبويب الصفحة المختارة حالياً	SelectedIndexChanged

الجدول رقم (١٠,١٩). مجموعة مختارة من صفات الصنف TabPage وأحداثه.

الوصف	الصفة
تحتوي على عنوان تبويب الصفحة	Text
تحتوي على نص عنوان نصيحة تبويب الصفحة التي تظهر عند نقل مؤشر الفأرة عليها	ShowToolTips
الوصف	الحدث
يحدث عندما يتم الضغط على مسطرة الصفحة بمؤشر الفأرة	Click

### استخدام القوائم الرئيسية وعناصر القوائم

#### Using Main Menus and Menu Items

يسمح لك كل من الصنف MainMenu والصنف MenuItem أن تنشئ قوائم رئيسة وقوائم فرعية داخل النموذج. يعمل عنصر القائمة الرئيسية كحاوية لاحتواء التركيب الرئيس للقائمة. وتمثل عناصر القائمة اختيارات القائمة داخل هذا التركيب. وسوف تدرج القائمة إلى النموذج بإسناد كائن الصنف MainMenu للصفة Menu الخاصة بالنموذج. يمكنك استخدام أسلوب القوائم في نظام برادشو مارينا لكي تُمكن المستخدم من اختيار أوامر إنجاز مهام النظام الأساسية مثل إضافة عميل جديد وتعديل بيانات عميل موجود وحذف عميل وهكذا. يلخص الجدول رقم (١٠,٢٠) والجدول رقم (١٠,٢١) أهم الخصائص والأحداث الخاصة لهذين العنصرين.

الجدول رقم (١٠,٢٠). مجموعة مختارة من صفات الصنف MainMenu.

الوصف	الصفة
تحتوي على مجموعة العناصر المصاحبة للقائمة الرئيسية	MenuItems

الجدول رقم (١٠,٢١). مجموعة مختارة من صفات الصنف MenuItem وأحداثه.

الوصف	الصفة
يحدد موضع عنصر القائمة داخل القائمة	Index
تحتوي على مجموعة العناصر المصاحبة للقائمة الحالية	MenuItems
تحدد المفاتيح المختصرة للمسؤولة عن تنشيط عنصر قائمة ما	ShortCut
تحدد هل تظهر المفاتيح المختصرة لعنصر القائمة أم لا	ShowShortCut
تحتوي على عنوان عنصر القائمة	Text
الوصف	الحدث
يحدث عندما يتم الضغط على عنصر القائمة بمؤشر الفأرة	Click

## ملخص الفصل

### Chapter Summary

- تمثل مكونات واجهة الاستخدام كائنات من أصناف واجهة المستخدم المتواجدة في الفضاء المسمى "System.Windows.Forms". يمثل الصنف Component الصنف الرئيس لجميع أصناف واجهة الاستخدام، ويمثل الصنف Control الصنف الرئيس لجميع أصناف واجهة الاستخدام المرئية.
- عندما نقوم بتطوير تطبيق يحتوي على واجهة استخدام رسومية، فنحن نستخدم أسلوب البرمجة المرئية (Visual Programming) التي تتكون من إضافة عناصر النموذج داخل النموذج وتصميمها، ثم ضبط خصائص كل من النموذج وعناصره، ثم إضافة الإجراءات المطلوبة التي تعالج الأحداث التي تصدر من المستخدم عند تفاعله مع النموذج.
- يتفاعل المستخدم مع واجهة البرنامج بواسطة إدخال بيانات أو الضغط على مكونات النموذج ومن ثم تصدر أحداث مثل حدوث الضغط على زر الفأرة الأيسر أو مثل حدوث الضغط على أحد مفاتيح لوحة المفاتيح التي يتم معالجتها بإضافة إجراءات معالجة الأحداث التي يتم استدعاؤها بشكل تلقائي عند حدوث هذه الأحداث.
- يتم توصيف عناصر النموذج أثناء عملية تصميمه مستخدماً الكلمة WithEvents حيث تشير الكلمة WithEvents إلى إمكانية إصدار أحداث (Events) من هذا العنصر، ومن ثم سوف تعالج هذه الأحداث بالإجراءات التي تحتوي على العبارة "Handles ControlName.EventName".
- يمثل النموذج نافذة مرئية وحاوية لمكونات مرئية أخرى، مع العلم أنه توجد حاويات أخرى تستخدم داخل النموذج لاحتواء مكونات أخرى مثل الإطارات.
- يستخدم العنوان (Label) لإظهار نص وتستخدم الصناديق النصية (Text Boxes) لاستقبال نصوص من المستخدم أو لإظهار معلومات نصية له، أما القائمة المنسدلة (Combo Box) فتقوم بنفس دور صندوق النص وتزيد عليه إمكانية اختيار عنصر من مجموعة من العناصر المعروفة مسبقاً.
- تقدم صناديق التأشير (Check Boxes) وأزرار الاختيار (Radio Buttons) إمكانية اختيار عنصر من عدة عناصر حيث تأخذ كل من صناديق التأشير وأزرار الاختيار حالة من اثنين وهما مؤشر (Checked) وغير مؤشر (Not Checked). تستخدم صناديق التأشير عادة للحصول على معلومات مرتبطة ببعضها حيث تعمل صناديق التأشير بشكل مستقل، أي أن اختيار صندوق تأشير لا يؤثر على صندوق آخر. بينما تستخدم أزرار الاختيار عندما نريد أن يكون هناك زر واحد مختار فقط؛ ولذلك فإن اختيار أحد الأزرار ينفي اختيار الزر الحالي.

- وأخيراً تقدم كل من قوائم الاختيار (List Boxes) وقوائم الاختيار بالتأشير (Checked List Boxes) إمكانية اختيار عنصر أو أكثر من مجموعة قيم معرفة سابقاً.
- تظهر الشجرة المرئية (Tree View) مجموعة من العناصر في شكل شجري حيث يمثل كل عنصر عقدة في الشجرة، وتمثل الشجرة المرئية كائناً من الصنف TreeNode، كما يمثل كل عنصر داخل الشجرة كائناً من الصنف TreeNode.

### المصطلحات الأساسية

#### Key Terms

صندوق التأشير (Check Box)	الصنف Component
زر الاختيار (Radio Button)	الصنف Control
قائمة الاختيار (List Boxes)	البرمجة المرئية (Visual Programming)
الشجرة المرئية (Tree View)	حدث (Event)
الصندوق النصي (Text Box)	العنوان (Label)
القائمة المنسدلة (Combo Box)	

### أسئلة المراجعة

#### Review Questions

- تكمّن أصناف واجهة المستخدم الرسومية في الفيجوال بيسك. نت في الفضاء المسمى:
  - .System.Data
  - .System.IO
  - .System.Web
  - .System.Windows.Forms
- كائن صنف النموذج يكون أيضاً .....:
  - صنف Component
  - صنف Container
  - صنف Control
  - جميع ما سبق

- ٣- العنوان صنف فرعي في :
- (أ) صنف Component.
- (ب) صنف Container.
- (ج) صنف Control.
- (د) جميع ما سبق.
- ٤- أسلوب نموذج واجهة المستخدم الرسومية ومظهره ومكونات الواجهة يطلق عليها :
- (أ) جاذبية.
- (ب) شخصية عامة.
- (ج) مظهر وشعور.
- (د) أسلوب ومظهر.
- ٥- عندما تنشئ كائناً من صنف الزر Button في نافذة تصميم النماذج ، تقوم لغة فيجوال بيسك .نت بشكل تلقائي بإضافة الكود الذي :
- (أ) يعرف كائن الزر.
- (ب) يستدعى إجراء الإنشاء من صنف الزر Button.
- (ج) يضع الخصائص الافتراضية لكائن الزر Button.
- (د) جميع ما سبق.
- ٦- تشير الكلمة المحجوزة Me إلى :
- (أ) كائن زر جديد.
- (ب) كائن نموذج جديد.
- (ج) كائن المكون الجديد.
- (د) الكائن الحالي.
- ٧- عادة ما يكون حدث فيجوال بيسك .نت لعنصر معين :
- (أ) نتيجة تفاعل المستخدم مع هذا العنصر.
- (ب) تم تعريفه بواسطة eventName.controlName.
- (ج) يصعب اكتسابه.
- (د) جميع ما سبق.

٨- إجراء الحدث الذي يستجيب لحدث الضغط على الزر المسمى Button1 يسمى .....

(أ) ButtonClickEvent.

(ب) Button1\_Click.

(ج) Click\_Button1.

(د) ButtonEvent.

٩- عنصر التحكم يسمح باختيار العديد من الاختيارات من مجموعة اختيارات:

(أ) Radio button.

(ب) Group box.

(ج) List box.

(د) Text box.

١٠- يفضل استخدام Group box في حالة:

(أ) ظهور العناصر القابلة للتوسعة.

(ب) إمكانية اختيار عنصر أو أكثر.

(ج) ظهور نص وإدخاله.

(د) تجميع منطقي لمكونات أخرى.

١١- ماهي الخاصية التي تستخدم في تغيير لون نص العنوان؟

(أ) خاصية اللون الأمامي ForeColor.

(ب) خاصية اللون الخلفي BackColor.

(ج) خاصية الخط Font.

(د) خاصية اللون Color.

١٢- ما هو الإجراء الذي يستخدم في إطلاق مصادر النظام عند انتهاء البرنامج؟

(أ) Event Handler.

(ب) Control.

(ج) Dispose.

(د) Exit.

١٣ - الكائن Anchor ليس عنصر في الصنف :

(أ) ContainerControl .

(ب) Form .

(ج) Button .

(د) MainMenu .

١٤ - أي من العناصر التالية له الخاصية PasswordChar؟

(أ) العنصر Label .

(ب) العنصر TreeNode .

(ج) العنصر Panel .

(د) العنصر TextBox .

١٥ - عنصر التحكم ..... و ..... تشترك في الحدث SelectedIndexChanged :

(أ) combo box, checked list box .

(ب) Combo box, check box .

(ج) Check box, radio button .

(د) Checked list box, radio button .

١٦ - أي من الإجراءات التالية لا ينتمي إلى الصنف TreeNode :

(أ) Collapse .

(ب) Expand .

(ج) Remove .

(د) Text .

١٧ - TabPage صنف فرعي في :

(أ) TabControl .

(ب) Panel .

(ج) TreeNode .

(د) ListControl .

١٨ - ما هو الاسم المناسب عليه لعنصر القائمة المسمى ملف (File)؟

(أ) .mnuFile

(ب) .stFile

(ج) .mnuitmFile

(د) .tabFile

١٩ - عند تصميم تطبيقات واجهة المستخدم الرسومية ، هل من المهم التقيد بالمعايير التي .....:

(أ) تقود إلى مكونات مظهر وشعور متوافقة.

(ب) تحسن سهولة الاستخدام.

(ج) تقلل احتمالية الأخطاء في إدخال البيانات.

(د) جميع ما سبق.

٢٠ - عنصر التحكم ..... يشترط اختياراً وحيداً:

(أ) .combo box

(ب) .List box

(ج) .label

(د) .Radio button

### أسئلة المناقشة

#### Discussion Questions

١ - استعن بوثيقة لغة VB .NET لمناقشة الأمور التالية والمتعلقة بالعنصر Context Menu :

(أ) عرف العنصر Context Menu؟

(ب) فيما يستخدم العنصر Context Menu؟

(ج) ما هي شجرة التوارث الخاصة بالعنصر Context Menu؟

(د) اذكر خصائص العنصر Context Menu العامة وغير الموروثة من الأصناف الرئيسة؟

(هـ) اذكر أحداث العنصر Context Menu العامة وتحت أي ظروف تنشأ؟

٢ - استعن بوثيقة لغة VB .NET لمناقشة أحداث لوحة المفاتيح والفأرة التالية :

(أ) حدد خمسة أحداث للفأرة غير الحدث MouseEnter وظروف حدوثها؟

(ب) ما هي أحداث لوحة المفاتيح الشائعة وظروف حدوثها؟

- ٣- ما فائدة إجراء تدمير الكائن (Destructor Method)؟ وما هو اسم إجراء تدمير الكائن الافتراضي الموروث من الصنف Component؟ ولماذا يعاد تعريفه في معظم تطبيقات الوندوز؟
- ٤- اذكر مبادئ تصميم النماذج الأساسية التي تعلمتها خلال هذا الفصل. ومن منطلق الخبرة التي اكتسبتها، اقترح على الأقل مبدأ واحداً لتصميم النماذج الذي ترى أنه يجب اتباعه عند التصميم.
- ٥- لماذا يجب الالتزام بأسلوب تسمية قياسي عند تسمية المتغيرات التي تمثل عناصر واجهة الاستخدام والأحداث وإجراءات معالجة الأحداث؟

## مشاريع الفصل

### Projects

- ١- تُستخدم المعادلة التالية في حساب قسط القرض الشهري الذي يجب أن يدفعه صاحب القرض:
- $$\text{paymentAmount} = (\text{amountOfLoan} * (\text{rate}/12)) / (1 - 1/(1 + \text{rate}/12) ^ \text{months})$$
- حيث إن المتغير rate يمثل معدل الفائدة السنوي (رقم بين صفر وواحد)، والرمز ^ يعني الأس، والمتغير months يعني عدد أشهر القرض. صمّم برنامج GUI يستقبل قيمة القرض (amountOfLoan) ومعدل الفائدة السنوي rate وعدد أشهر سداد القرض months ثم قم بحساب قيمة القسط الشهري بناء على هذه المعادلة.
- ٢- أنشئ نموذجاً يحتوي على زرّين: الزرّ "Push Me" والزرّ "Exit" (الزرّ الافتراضي.. استخدم الخاصية AcceptButton)، وعندما يضغط على الزرّ الأول تظهر رسالة "Hello World"، وعندما يضغط على الزرّ الثاني يغلق النموذج:
- (أ) عدل البرنامج السابق حيث يتم ظهور رسالة عند الضغط على الزر Exit تحذر المستخدم أن البرنامج سيغلق، فإذا ضغط المستخدم على الزر Yes يغلق البرنامج، أما إذا ضغط على الزر No لا يتم غلق البرنامج.
- (ب) أضف قائمة منسدلة للنموذج السابق، ثم أضف إليه ثلاث قيم. استخدم الخاصية SelectedIndex لمعرفة القيمة المختارة وإظهارها في رسالة ذات معنى.
- ٣- أنشئ النموذج الموضح في الشكل رقم (١٠.١)، ثم أضف القيم لكل من القائمة والقائمة المنسدلة كما هو واضح في الشكل رقم (١٠.٩). أضف إجراءات معالجة الأحداث للنموذج للعناصر التالية: صندوق الاختيار Scholarship Student، والقائمة University، و زر الاختيار Freshman، والزر Save. يجب أن يستجيب كل إجراء برسالة تفيد مصدر إصدار الحدث.

- ٤- أنشئ برنامج VB.NET يحتوي على شجرة مرئية تناظر شجرة أصناف واجهة الاستخدام الموضحة في الشكل رقم (١٠٣). عرف إجراء معالجة الحدث AfterSelect الذي يميز أيّاً من عقد الشجرة قد تم اختيارها ثم يظهر رسالة عن بيانات العقدة الحالية والعقد المرتبطة بها.