

الفصل الثالث والعشرون استخدام الحاويات Containers

لا تقوم الحاويات **Containers** بتنفيذ وظائف مستقلة بحد ذاتها وإنما وظيفتها تتمثل في تجميع العناصر الأخرى لكي لا تظهر العناصر مبعثرة على الشاشة. وهذا نوع من تحسين واجهات المستخدم يضاف لما سبق أن عرفته في الفصول السابقة. بانتهاء هذا الفصل ستتعرف على:

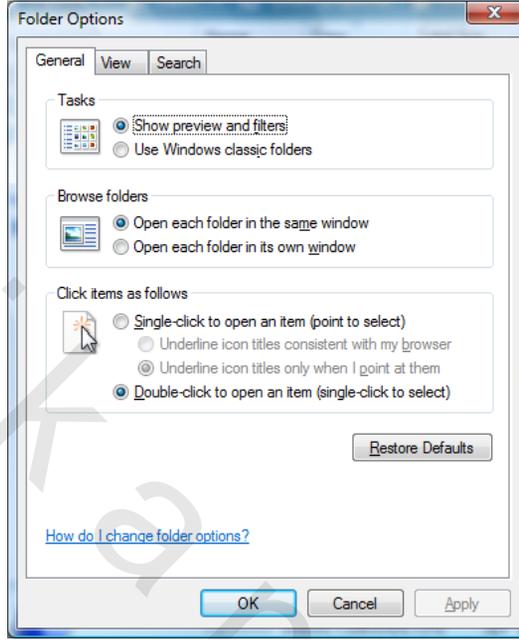
- ◆ استخدام مربع المجموعة **GroupBox** لعمل مجموعات من أزرار الاختيار.
- ◆ استخدام الحاويات لتحسين استخدام مساحة النافذة.
- ◆ التحكم في الوظائف المتاحة للمستخدم عن طريق الحاويات.
- ◆ استخدام أداة التبويب **TabControl**.
- ◆ إنشاء المعالجات لتسهيل عمل المستخدم.

الوظيفة الأساسية للحاويات هي تجميع الكائنات ذات الوظيفة الواحدة في مكان واحد ويحقق ذلك ميزة استغلال أفضل لمساحة الشاشة. أيضاً تسمح الحاويات بالتعامل مع الكائنات كمجموعات مما يقلل من الجهد المستخدم في التعامل مع كل كائن على حدة، يضاف إلى ذلك ضرورة استخدامها في حالة وجود عدة مجموعات من أزرار الاختيار على نفس النموذج.

نتناول في هذا الفصل الحاويات Containers واستخداماتها المختلفة في Visual Basic سنشرح أداة مربع المجموعة GroupBox ومربع الرسم PictureBox ثم نتناول أداة التبويب TabControl بالتفصيل.

تقسيم النافذة الواحدة إلى عدة صفحات

عادةً ما يفضل استخدام نافذة واحدة لكثير من البرامج، وهذا هو اختيار المستخدم في كثير من الحالات. ولكن إذا زادت المدخلات أو المخرجات عبر هذه النافذة فسيصبح نشر العديد من الكائنات وأدوات التحكم عليها أمر مزعج للمستخدم. الحل في ذلك (مع الاحتفاظ بنافذة واحدة للبرنامج) أن تقوم باستخدام الحاويات. ولعلك قابلت ذلك مراراً من خلال تعاملك مع نظام التشغيل Windows (انظر شكل ٢٣-١).



شكل ٢٣-١ استخدام الحاويات لتجميع الوظائف المتشابهة

يشتمل Visual Basic علي أربع أدوات تعمل كحاويات.

- مربع المجموعة **GroupBox**
- أداة اللوحة **Panel**
- مربع الرسم **PictureBox**
- أداة التبويب **TabControl**

بالطبع تعرف الأدوات الثلاثة الأولى من متابعتك للفصول السابقة من الكتاب أما الأداة الأخيرة فهي التي تقوم بإظهار التبويبات المختلفة الموجودة على أى نموذج (راجع شكل ٢٣-١).

حاول دائماً ألا تضع عناصر كثيرة على نافذة واحدة، فهذا يجعل النافذة بطيئة جداً، وكحد أقصى لا تزيد عن 200 أداة. وإن احتجت لأكثر من ذلك فاستخدم نافذة أخرى.



إنشاء عدة مجموعات من أزرار التحكم

سبق أن أوضحنا عند حديثنا عن أدوات التحكم الأساسية استخدام مربع المجموعة للفصل بين مجموعات مختلفة من أزرار التحكم. يعتبر ذلك ضرورة عند الرغبة مثلاً في اختيار أكثر من عنصر في مجموعات مختلفة، حيث لن نستطيع ذلك إلا باستخدام حاوية.

استخدام مربع المجموعة GroupBox

يعتبر مربع المجموعة GroupBox بديل لأداة مربع المجموعة Frame Control الموجودة بالإصدارات القديمة من Visual Basic والتي تسبق Visual Basic.NET. ومربع المجموعة هو أبسط الحاويات وأقلها استخداماً للذاكرة، لذا يفضل استخدامه عن أقرانه عند عدم وجود حاجة إليها. الاستخدام الأول لمربع المجموعة كما علمنا هو الفصل بين مجموعات أزرار الاختيار المختلفة (انظر شكل ٢٣-٢).



شكل ٢٣-٢ استخدام إطارات متعددة لإنشاء عدة مجموعات من أزرار الاختيار

إعداد مربع المجموعة

مربع المجموعة كما قلنا كائن بسيط جداً. لرسم مربع مجموعة علي النموذج انقر الأداة  GroupBox من مربع الأدوات ثم ارسم الأداة علي النموذج بالحجم الذي يروق لك. بعد رسم مربع المجموعة قد تحتاج إلى تغيير خصائص قليلة فيه أهمها الاسم Name والعنوان Text (العنوان الذي يظهر أعلى مربع المجموعة) و FlatStyle (نمط مربع المجموعة محاط بمربع أو لا) والخاصية RightToLeft التي تحدد اتجاه مربع المجموعة. يوضح شكل ٢٣-٣ التالي التأثير المختلف للخصائص السابقة.



شكل ٢٣-٣ تأثير قيم الخاصية FlatStyle وRightToLeft

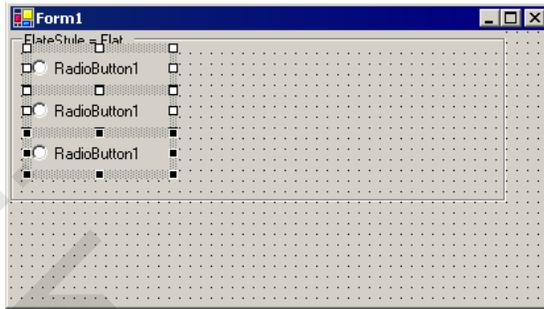
إضافة العناصر إلى مربع المجموعة

لإضافة عنصر داخل مربع المجموعة قم برسم العنصر داخل مربع المجموعة تماماً كما تفعل عند رسم الأداة على النموذج. هذه هي الطريقة التي تجعل مربع المجموعة يحتوي العنصر. وهذا على عكس الإصدارات القديمة من Visual Basic (التي تسمى Visual Basic.NET)، حيث لا تعترف هذه الإصدارات بالعمليات الآتية:

- سحب عنصر مرسوم على الشاشة إلى داخل مربع المجموعة
- أو رسم مربع المجموعة على عناصر مرسومة أصلاً على النموذج
- أو النقر المزدوج على الأداة مع تعليم مربع المجموعة

تعلمت في ما سبق من فصول الكتاب كيف يمكن اختيار مجموعة عناصر للتعامل مع خصائصها أو لحاذاؤها. نفس الأمر يمكن تطبيقه على عدة عناصر داخل حاوية، ولكن عليك أولاً أن تنقر على النموذج خارج أي عنصر (يجب ألا يكون أي عنصر محددًا) بعد ذلك اضغط **Ctrl** أو **Shift** ثم ابدأ بالسحب حول العناصر التي تريد تحديدها كما في شكل

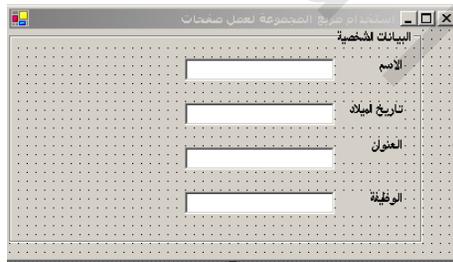
٢٣-٤ . يمكنك بعد ذلك استخدام **Ctrl** مع النقر لكي تختار أو تزيل الاختيار عن عنصر وحيد.



شكل ٢٣-٤ عناصر متعددة تم تحديدها داخل إطار

إنشاء صفحات متعددة على النموذج والتبديل بينها

يستخدم مربع المجموعة أيضاً لاستغلال مساحة صغيرة في عرض معلومات كثيرة وذلك بتنظيمها في صفحات يمكن تقلبها كالكتاب للوصول إلى الصفحة المرادة. لكي يتضح لك كيفية عمل ذلك قم بإضافة مربع مجموعة وأضف عليه عدة عناصر ليبدو كما في شكل ٢٣-٥.



شكل ٢٣-٥ مربع مجموعة يحتوي على بعض العناصر كصفحة أولى

بعد ذلك أضف إطاراً آخر، لكي تفعل ذلك قم بإزاحة مربع المجموعة الأول لتستطيع رسم مربع المجموعة الثاني (تذكر أن ترسمه بدءاً من خارج مربع المجموعة الأول)، بعد ذلك أضف إليه عدة عناصر أيضاً ليبدو كما في شكل ٢٣-٦.



شكل ٢٣-٦ مربع المجموعة الثاني بعد وضعه على النموذج

ستحتاج إلى التبديل بين مربعي المجموعة أثناء التصميم. لكي تحضر مربع إلى الواجهة لتستطيع التعامل مع عناصره. انقره بزر الفأرة الأيمن ثم اختر **Bring To Front** أو اختر **Send To Back** لإرساله إلى الخلف.

للتبديل بين الصفحات من خلال الكود. اجعل قيمة الخاصية **Visible** للصفحة المرادة **True** واجعلها **False** لبقية الصفحات (مربع المجموعة الثاني في هذه الحالة). مع الأخذ في الاعتبار أن البرنامج لا بد أن يقوم في بدايته بإخفاء جميع مربعات المجموعة عدا واحداً فقط. قم بإضافة الكود الآتي إلى الإجراء **Form_load**، حيث يقوم هذا الكود بإخفاء الصفحة الثانية:

```
Private Sub Form1_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
    GroupBox1.Visible = True
    GroupBox2.Visible = False
End Sub
```

للتبديل بين الصفحات أضف زر أمر إلى النموذج ثم اكتب الكود التالي في الإجراء **Click** الخاص به:

```
Private Sub Button1_Click(sender As System.Object, e As
System.EventArgs) Handles Button1.Click
    If GroupBox1.Visible Then
        GroupBox2.Visible = True
        GroupBox1.Visible = False
        Button1.Text = "عرض البيانات الشخصية"
    Else
```

```
GroupBox1.Visible = True
GroupBox2.Visible = False
Button1.Text = "عرض الدرجة العلمية"
End If
```

End Sub

يمكنك الآن تشغيل البرنامج بـ F5 ليظهر كما في الشكل ٢٣-٧، لاحظ أن عنوان الزر يتغير تبعاً للصفحة المعروضة.



شكل ٢٣-٧ البرنامج عند تشغيله حيث يمكن التبديل بين الصفحات.

تشبه أداة اللوحة PanelTool إلى حد كبير مربع المجموعة إلا أنها قد تظهر بدون إطار كما أنها لا تحتوى على عنوان. وعلى هذا فكل ما شرحناه عن مربع المجموعة ينطبق تماماً على اللوحة Panel.

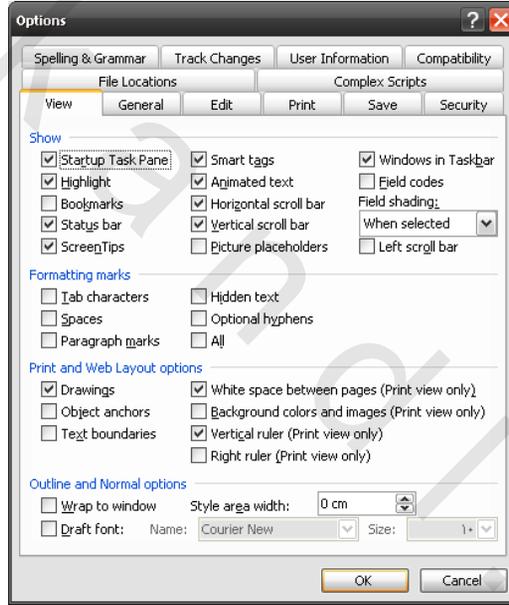


استخدام مربع الرسم PictureBox

يشبه مربع الرسم مجموعة تماماً من حيث طريقة استخدامه، إلا أنه يفوقه في أن مربع الرسم يمكن أن يحتوى على الصور والرسوم، لذا فهو يستخدم المزيد من مساحة الذاكرة مقارنةً بمربع المجموعة أو اللوحة، إلا أنها على أية حال أداة متعددة الاستخدامات ومفيدة جداً. إلا أن فوائدها الكبيرة لا تتبع من استخدامها كحاوية. وكما قلنا من السهل جداً استخدامها كحاوية كمربع المجموعة تماماً.

استخدام أداة التبويب TabControl

كما ترى في المثال السابق الذي استخدمنا فيه مربع المجموعة والأزرار لمحاكاة الصفحات أن المثال تضمن عدة عيوب. أولها هو صعوبة العمل أثناء التصميم حيث تخفي مربعات المجموعة بعضها. ثانيها الحاجة إلى ضبط مواقع مربع المجموعة بالكود عند تشغيل البرنامج، ثالثها وهو الأهم عدم إحساس المستخدم بوجود عدة صفحات كالإحساس الذي تجده عند رؤيتك لصفحات الخيارات في برنامج Word مثلاً (انظر شكل ٢٣-٨).

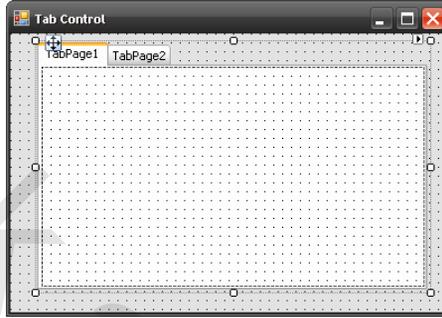


شكل ٢٣-٨ صفحات الخيارات داخل Word كنموذج للصفحات المتعددة

يشتمل Visual Basic علي أداة مهمة ومفيدة للغاية وهي أداة التبويب TabControl والتي تتيح لك إظهار صفحات كالتالي تراها في شكل ٢٣-٨ السابق تماماً. مع سهولة متناهية في العمل معها وقت التصميم أثناء التشغيل، حيث تعتبر أداة مثالية للبرامج كثيرة المدخلات.

إنشاء صفحات مبهوبة داخل البرنامج

كما تعودت في كافة الأدوات، اختر الأداة **TabControl** من مربع الأدوات ثم ارسمها على النموذج وقم بضبط حجمها تبعاً لعدد التبويبات والمحتويات المزمع إضافتها (انظر شكل ٢٣-٩).



شكل ٢٣-٩ وضع أداة التبويب على النموذج

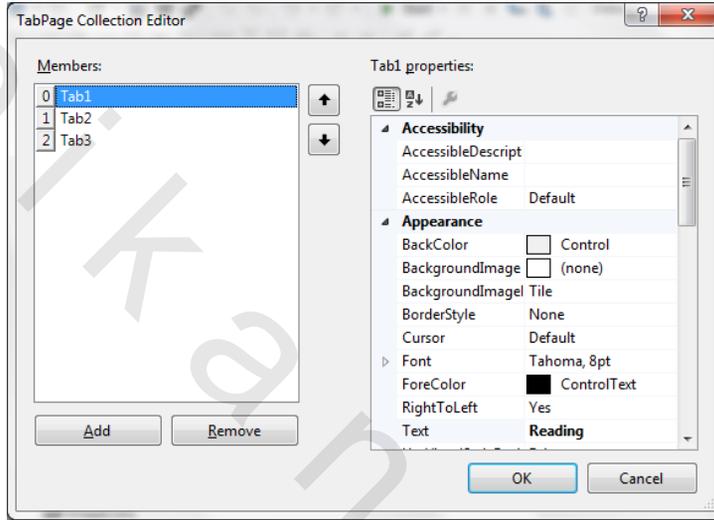
ضبط مربع الأداة

الخطوة الأولى في العمل مع أداة التبويب هو تعيين التبويبات التي تشتمل عليها الأداة. لأداء ذلك، تابع معنا الخطوات الآتية:

١. تأكد من تنشيط أداة التبويب ثم انقر الزر المجاور للخاصية **TabPage** داخل مربع الخصائص، يظهر المربع الحوارى **TabPage Collection Editor** الذى يمكنك من خلاله تعيين التبويبات المختلفة بالإضافة إلى ضبط خصائص كل تبويب على حده، حيث يظهر افتراضياً تبويين بالأداة (انظر شكل ٢٣-١٠).
٢. لإضافة تبويب جديد، انقر زر **Add**، وحينئذٍ يتم إضافة التبويب الجديد بالعمود الأيسر من المربع الحوارى كما تظهر خصائص التبويب بالعمود الأيمن (انظر شكل ٢٣-١٠).
٣. قم بتغيير خصائص التبويب من العمود الأيمن. ولعل أهم هذه الخصائص يتمثل فى الخاصية **Name** التى تحتوى اسم التبويب كى تتمكن من التعامل معه من خلال الكود، والخاصية **Text** التى تحدد عنوان التبويب داخل النموذج وكذلك الخاصية

ToolTip Text التي يمكنك من خلالها تحديد التلميح الذي يظهر بمجرد مرور مؤشر الفأرة على التبويب.

٤. قم بتكرار الخطوتين السابقتين لإضافة المزيد من التبويبات.



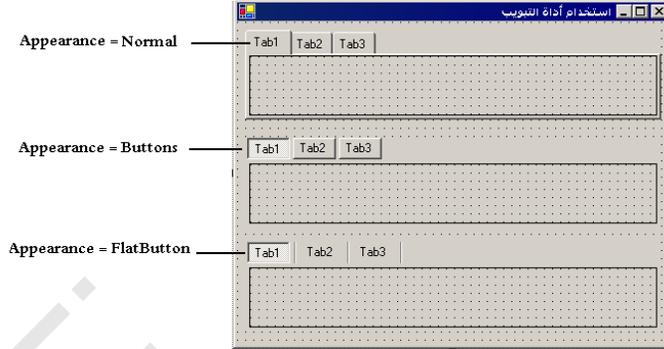
شكل ٢٣-١٠ تعيين التبويبات من خلال المربع الحوارى TabPage Collection Editor

تغيير مظهر الصفحات

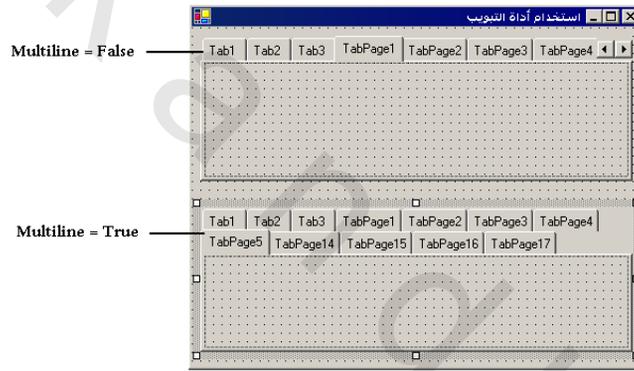
هناك عدة خصائص تحكم الشكل العام لأداة التبويب بالكامل كما أن هناك بعض الخصائص التي تحكم الشكل العام لكل تبويب على حده. ولعل أهم خصائص الأداة ككل ما يلي:

- الخاصية **Appearance** وتحدد مظهر رؤوس التبويبات على الأداة وتحتوى على القيم **Normal** لإظهار التبويبات بشكلها المعتاد و **Buttons** لإظهار رؤوس التبويبات على شكل أزرار و **FlatButtons** لإظهارها على شكل أزرار محفورة (انظر شكل ٢٣-١١).

- الخاصية **Multiline** وتحتوى على قيمتين **True** أو **False**. تستخدم القيمة **True** إذا أردت إنشاء أكثر من صف من التبويبات داخل الأداة أما القيمة **False** فتستخدم إذا أردت إظهار صف واحد فقط أعلى الأداة (انظر شكل ٢٣-١٢).

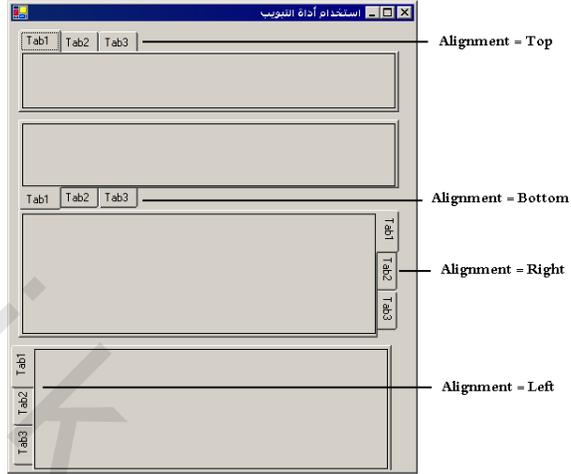


شكل ٢٣-١١ أنماط مختلفة للصفحات



شكل ٢٣-١٢ مواضع مختلفة للتبويبات

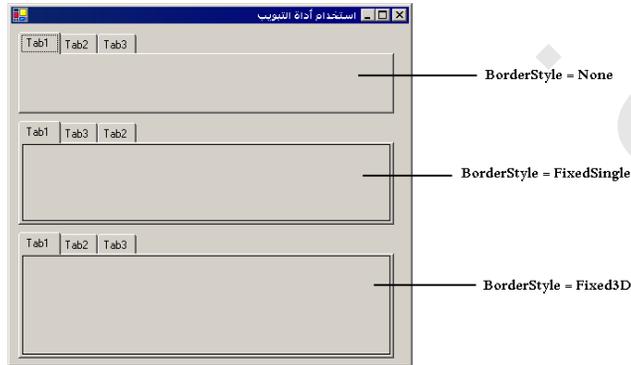
- الخاصية **ShowToolTips** وتحتوى على قيمتين أيضاً **True** أو **False**. تستخدم القيمة **True** إذا أردت إظهار التلميحات المحددة من خلال الخاصية **ToolTipText** عند التوقف للحظات بمؤشر الفأرة على رأس التبويب. أما القيمة **False** فتستخدم بالطبع إذا لم ترغب في إظهار هذه التلميحات.
- الخاصية **ImageList** وتستخدم لتحديد اسم قائمة الصور التي تقوم باستخدام عناصرها كصور مصاحبة لنص رأس التبويب إذا رغبت في ذلك.
- الخاصية **Alignment** وتحدد محاذاة أداة التبويب على النموذج وتأخذ إحدى قيم الجوانب الأربعة **Top** و **Bottom** و **Left** و **Right** (انظر شكل ٢٣-١٣).



شكل ٢٣-١٣ القيم المختلفة للخاصية **Alignment**

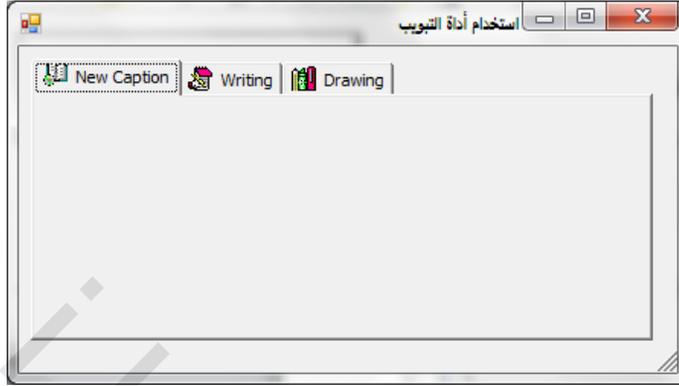
بالإضافة إلى الخصائص السابقة التي تخص الأداة بالكامل، هناك بعض الخصائص التي تعمل مع كل تبويب على حدة والتي أهمها ما يلي:

- الخاصية **BorderStyle** وتستخدم لتحديد مظهر الإطار الذي يظهر حول التبويب ويأخذ إحدى القيم **None** لعدم إنشاء إطار من الأساس و **FixedSingle** لإنشاء إطار عادي بسمك نقطة واحدة و **Fixed3D** لإنشاء إطار ثلاثي الأبعاد (انظر شكل ٢٣-١٤).



شكل ٢٣-١٤ القيم المختلفة للخاصية **BorderStyle**

- الخاصية **AutoScroll** وتستخدم لعرض شرائط التمرير تلقائياً عند الحاجة وتأخذ إحدى القيمتين **True** أو **False**.
 - الخاصية **ImageIndex** وتستخدم لتعيين صورة لرأس التبويب من قائمة الصور المعرفة داخل الخاصية **imageList** بأداة التبويب نفسها (إن وجدت).
 - الخاصية **ToolTipText** وتستخدم لتعيين تلميح نصي يظهر بمجرد الوقوف لشواني بزر الفأرة على رأس التبويب.
ضبط التبويبات الموجودة بمربع الأداة
يمكنك إعطاء مظهر متميز لكل صفحة من خلال خاصيتين تخصصان كل صفحة وهما خاصية العنوان **Text** والصورة التي يتم وضعها في رأس التبويب من خلال الخاصية **ImageIndex**. للتعرف على كيفية إضافة صورة لرأس كل تبويب، تابع معنا الخطوات الآتية:
١. انقر النموذج الذي يحتوي على أداة التبويب لتنشيطه ثم قم بإضافة أداة قائمة صور **ImageList** إليه.
 ٢. انقر أداة قائمة الصور لتنشيطها ثم انقر الخاصية **Images** وقم بإضافة الصور المناسبة إلى القائمة كما تعلمت من قبل.
 ٣. قم بتنشيط أداة التبويب ثم انقر الخاصية **ImageList** واختر قائمة الصور التي قمنا بإضافتها للنموذج (**ImageList1** مثلاً).
 ٤. قم بتنشيط كل تبويب داخل الأداة على حده ثم انقر الخاصية **ImageIndex** وقم باختيار الصورة التي ترغب في إظهارها في رأس التبويب (انظر شكل ٢٣-١٥).



شكل ٢٣-١٥ إضافة الصور إلى التبويبات

أثناء العمل في مرحلة التصميم، يمكنك اختيار أداة التبويب كلها كما يمكنك اختيار كل تبويب على حدة وفي هذه الحالة يحتوى مربع الخصائص على خصائص العنصر المختار سواء الأداة نفسها أو صفحة من صفحاتها. لاختيار أداة التبويب، انقر أي من رؤوس التبويبات الموجودة بالأداة. ولاختيار تبويب من التبويبات، انقر رأس التبويب أولاً ثم انقر داخل صفحة التبويب لاختيارها.



التحكم في التبويبات من خلال الكود

يمكنك من خلال الكود التحكم في خصائص الأداة نفسها أو في خصائص كل تبويب على حدة حيث يتم التعامل مع الخصائص التابعة للصفحات ككل مثل **Alignment** و **Appearance** بصورة طبيعية في الكود (بالتسمية المنقوطة طبعاً) بينما يتم التعامل مع الخصائص التابعة لكل تبويب على حده مثل الخاصية **Text** أو الخاصية **BorderStyle** من خلال استخدام دليل التبويب أو الصفحة داخل الخاصية **TabPage** هكذا **TabPage(0)** (حيث يبدأ دليل الصفحات من 0 وليس 1). فمثلاً لتغيير عنوان الصفحة الأولى من خلال الكود، يمكنك استخدام السطر التالي:

```
TabControl1.TabPages(0).Text = "New Caption"
```

بطريقة مماثلة يمكنك تغيير الصورة المستخدمة مع الصفحة الثانية كما يلي:

```
TabControl1.TabPages(1).ImageIndex = 2
```

حيث تعبر 2 عن ترتيب الصورة داخل قائمة الصور، أى الصورة الثالثة.
يمكنك أيضاً استرجاع عدد التبويبات الموجودة بالأداة من خلال الكود وذلك باستخدام الوظيفة (Count) كما يلي:

Dim num As Integer

num = TabControl1.TabPages.Count()

MessageBox.Show("Number of Tabs is " & num)

نقطة أخيرة وهامة قد تحتاج إليها وهى تقليب الصفحات بالكود، بمعنى إظهار صفحة معينة دون أن يضغظ المستخدم التبويب. يتم ذلك من خلال الخاصية **SelectedTab**، فلو أردت إظهار الصفحة **TabPage2** دون أن يطلب المستخدم ذلك، يمكنك استخدام السطر التالى:

TabControl1.SelectedTab = TabPage2

فكرة إنهاء المعالجات Wizards

لقد انتشرت فكرة المعالج Wizard كثيراً بعد أن أثبتت نجاحها، حيث أن المستخدم الذي يبدأ في استخدام البرنامج يزججه بشدة الكم الكبير من القوائم وشرائط الأدوات. فكل ما يريده هو الخطوات الأساسية التي تقوده إلى إنجاز عمل معين من أجله قام بشراء البرنامج. وهذا ما تقدمه البرامج المساعدة **Wizards**، مثال ذلك معالج التنصيب (التثبيت) **Setup wizard** الذي كثيراً ما استخدمناه في تثبيت البرامج.

يتميز أي معالج بوجود عدة صفحات متتالية في نفس الشاشة وأربعة أزرار أساسية هي:

- التالي: **Next**
- السابق: **Previous**
- إنهاء: **Finish**
- إلغاء الأمر: **Cancel**

بتتبع سلوك المعالجات نلاحظ أن زر "إنهاء" **Finish** يظل معتمولا يمكن إلا عند الوصول للصفحة الأخيرة. بينما يظل زر "إلغاء الأمر" **Cancel** ممكناً دائماً. أما زر "السابق" **Previous** فيظل ممكناً دائماً عدا في الصفحة الأولى، وعلى العكس فإن زر "التالي"

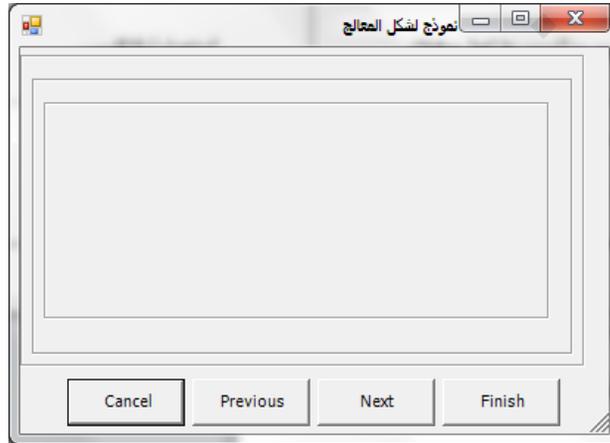
Next يظل ممكناً دائماً عدا في الصفحة الأخيرة.

عندما تبدأ في إنشاء المعالج، ضع عدداً من مربعات المجموعة يساوي عدد الصفحات التي تريدها في المعالج (عدد الخطوات). وأفضل الطرق استخدام القصواللصق لإنشاء مصفوفة من مربعات المجموعة مما يسهل كثيراً من عملية البرمجة وكتابة الكود. وقد تفضل جعل هذه المربعات محيطة بـ **Border** كي لا توهي بالأسلوب المستخدم في إنشائها (حينئذ ضع خطأ فاصلاً بينها وبين الأزرار). أيضاً تحتاج إلى وضع الأزرار الأربعة الأساسية أسفل مربعات المجموعة.

في الكود الخاص بالبدء، يتم ضبط موضع جميع مربعات المجموعة لتساوي الموضع الصحيح للصفحة ثم يتم إخفاء جميع مربعات المجموعة عدا الصفحة الأولى فقط، كما يتم إعتام زر الإنهاء.

وفي الكود الخاص بزر "التالي" **Next** وزر "السابق" **Previous** يتم عمل شيئين:

١. إخفاء مربع المجموعة الحالي وعرض مربع المجموعة التالي أو السابق.
 ٢. اختبار كون الصفحة الحالية الأولى أو الأخيرة لإعتام أحد الزرين ولتمكين زر الإنهاء.
- وفي الكود الخاص بزر الإنهاء من المفترض أن تتم الوظيفة الفعلية للبرنامج بعد اختبار المدخلات في الصفحات المختلفة.
- الصورة العامة التي سيكون عليها برنامجك أثناء التصميم تظهر كما فيشكل ٢٣-١٦.



شكل ٢٣-١٦ نموذج لبرنامج معالج Wizard أثناء تصميمه

