

الفصل الخامس

إنشاء القوائم

يتركز اهتمام هذا الفصل، على آلية دمج القوائم في برامجك. وسوف نتعرف من خلاله على الطريقة التي يتم بها تصميم القائمة وكيفية ربطها مع البرنامج.

برنامج الألوان

سنبدأ الآن بكتابة برنامج يتضمن قائمة Menu، وسندعوه برنامج الألوان. يسمح لك برنامج الألوان باختيار لون من قائمة، وتلوين خلفية البرنامج باللون المنتقى. كما يسمح باختيار حجم إطار البرنامج من قائمة أخرى. يلزمنا قبل البدء بكتابة برنامج الألوان، تحديد كيف ستبدو القائمة في البرنامج وما الذي يفترض أن تفعله:

الشكل ٥-١

برنامج الألوان



□ سيظهر لك عند تشغيل برنامج الألوان، شريط قوائم. يحمل عنواني قائمتين هما:

قائمة الألوان وقائمة الحجم. (انظر الشكل ٥-١).

- تحتوي قائمة الألوان على بندين هما: بند تحديد اللون وبند خروج (انظر الشكل ٥-٢).

الشكل ٥-٢

بندي قائمة الألوان



- كما تحوي قائمة الحجم على بندين أيضاً، هما: البند صغير والبند كبير.

الشكل ٥-٣

قائمة الحجم.



- تظهر قائمة أخرى لدى اختيار البند تحديد اللون من قائمة الألوان. تُظهر هذه القائمة لائحةً بالألوان التي يمكنك الانتقاء من بينها. يتغير لون خلفية نموذج البرنامج إلى اللون الجديد فور اختياره. (انظر الشكل ٥-٤).
- كذلك، يتغير حجم إطار برنامج الألوان، تبعاً للبند الذي تنتقيه من قائمة الحجم.
- يتسبب اختيار البند خروج، من قائمة الألوان بإنهاء البرنامج.

التمثيل المرئي لبرنامج الألوان

سنبدأ كعادتنا عند تصميم البرنامج، ببناء نموذج البرنامج:

- أنشئ الدليل الفرعي التالي: C:\VB5Prg\Ch05.
- ابدأ مشروعاً جديداً Project من النوع Standard EXE.
- احفظ نموذج المشروع باسم Colors.Frm في الدليل الفرعي C:\VB5Prg\Ch05 ، ثم احفظ ملف المشروع باسم Colors.Vbp في ذات الدليل.
- أنشئ نموذج برنامج الألوان طبقاً للجدول ٥-١.

الجدول ٥-١. جدول خصائص برنامج الألوان.

القيمة	الخاصية	الكائن
frmColors	Name	Form
برنامج الألوان	Caption	
True	RightToLeft	
White	BackColor	

بناء قوائم برنامج الألوان

سنعمل الآن على إنشاء قوائم برنامج الألوان. تحتاج القائمة لنموذج ترتبط معه، لهذا يجب علينا اختيار النموذج الذي سترتبط به القائمة، قبل بناء هذه القائمة:

□ تأكد أن النموذج frmColors هو الإطار المنتقى. (قد تبدو هذه الخطوة لا لزوم لها في برنامجنا هذا، على اعتبار أن مشروعنا لا يحتوي سوى نموذجاً واحداً فقط). ولكن تغدو هذه الخطوة ضرورية عندما يحوي المشروع على أكثر من نموذج واحد، لأنه سيتوجب عليك ربط القائمة بالنموذج الذي تحدده.

ملاحظة

يؤدي النقر بالزر الأيسر للفأرة داخل النموذج أو أي عنصر آخر إلى اختيار هذا النموذج.

الآن، وبعد أن انتهينا من اختيار النموذج frmColors، سيربط فيجول بيسك القائمة بالنموذج frmColors.

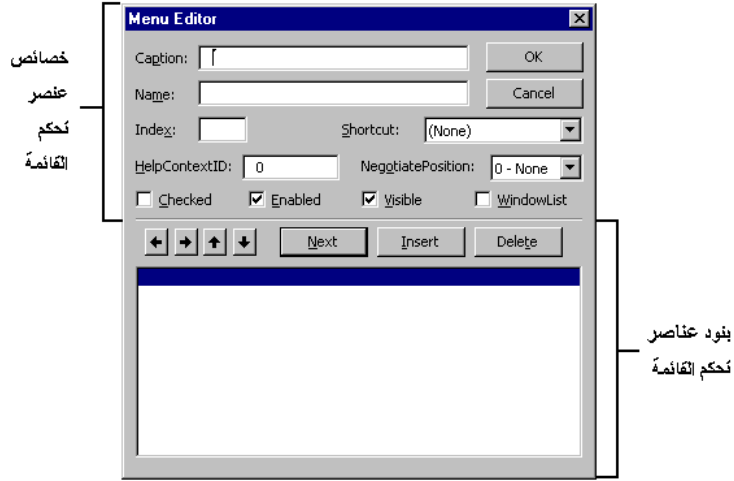
□ اختر البند Menu Editor من القائمة Tools.

يستجيب فيجول بيسك بإظهار إطار Menu Editor المبين في الشكل ٥-٥.

سنستخدم إطار Menu Editor من الآن فصاعداً، لبناء نظام القائمة المبين في الشكل ٥-١.

الشكل ٥-٥

إطار Menu Editor
محرر القائمة.



إنشاء عناصر تحكم القائمة

حسب ما يوضحه الشكل ٥-٥ فإن مربع حوار Menu Editor يقسم إلى جزئين: الجزء السفلي وهو لائحة بأسماء عناصر تحكم القائمة Menu Control List Box، والجزء العلوي ويعرض خصائص كل عنصر Menu Control Properties. يحتوي الجزء السفلي عادةً، كل عناصر تحكم القائمة الموجودة في البرنامج، مثلاً، سيحتوي هذا الجزء في مثالنا الحالي على كل عناصر تحكم القائمة الموجودة في برنامج الألوان، السؤال هو: ما هي عناصر تحكم القائمة؟. يمكن أن تكون عناصر تحكم القائمة عبارة عن عناوين قوائم. (مثل عنوان قائمة الألوان وعنوان قائمة الحجم، المبينين في الشكل ٥-١). أو قد تكون بنود قوائم (مثل البند تحديد اللون والبند خروج، المبينين في الشكل ٥-٢). يمتلك برنامج الألوان كما رأينا قائمتين منسدلتين هما: قائمة الألوان المبينة في الشكل ٥-٢ وقائمة الحجم المبينة في الشكل ٥-٣. تمتلك قائمة الألوان بدورها ثلاثة عناصر تحكم قائمة هي:

الألوان (عنوان القائمة).

تحديد اللون (بند قائمة).

خروج (بند قائمة).

حسب ما يوضحه الشكل ٥-٢.

يظهر حتى هذه النقطة، شريط غامق في الجزء السفلي من محرر القائمة. كما هو واضح في الشكل ٥-٥. وهذا يعني أن فيجول بيسك صار جاهزاً لتشكيل عناصر تحكم القائمة.

اتبع الخطوات التالية لإنشاء عنوان قائمة الألوان:

□ اكتب "&الألوان" في الحقل النصي Caption. بدون كتابة إشارات التنصيص.

□ اكتب "mnuColors" في الحقل النصي Name. بدون كتابة إشارات التنصيص.

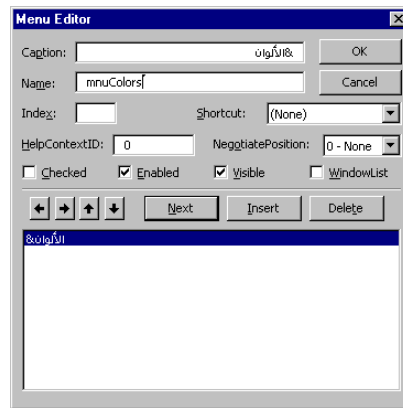
اتفقنا أن الرمز & قبل أحد الأحرف (للخاصية Caption فقط)، يعني أن الحرف الذي يأتي وراء هذا الرمز سيظهر وتحت خط. فمثلاً ستظهر قائمة الألوان مع خط قصير تحت الحرف "ا"، هكذا "ا". مما يعني أنه يمكن الضغط على المفاتيح "Alt+ا"، عند تنفيذ البرنامج لاختيار قائمة الألوان بدلاً من النقر على عنوانها بواسطة الزر الأيسر للفأرة.

انتهينا للتو من إنشاء عنوان لقائمة الألوان. يفترض أن يظهر الـ Menu Editor الآن كما في الشكل ٥-٦.

الشكل ٥-٦

إنشاء عنوان

قائمة الألوان



الخطوة التالية الآن إنشاء بندي قائمة الألوان.

اتبع الخطوات التالية لإنشاء البند تحديد اللون في قائمة الألوان:

□ انقر الزر Next في إطار Menu Editor.

يستجيب فيجول بيسك بإضاءة السطر التالي في الجزء السفلي من إطار *Menu Editor*.

□ اكتب "&تحديد اللون" في الحقل النصي Caption من الإطار Menu Editor.

□ اكتب "mnuSetColor" في الحقل النصي Name من الإطار Menu Editor.

باعتبار أن بند تحديد اللون موجود في قائمة الألوان، إذاً لا بد من إزاحته وجعله كفرع ثانوي من الفرع الأصلي. اتبع الخطوات التالية لإزاحة بند تحديد اللون:

□ انقر زر السهم اليميني في الإطار Menu Editor.

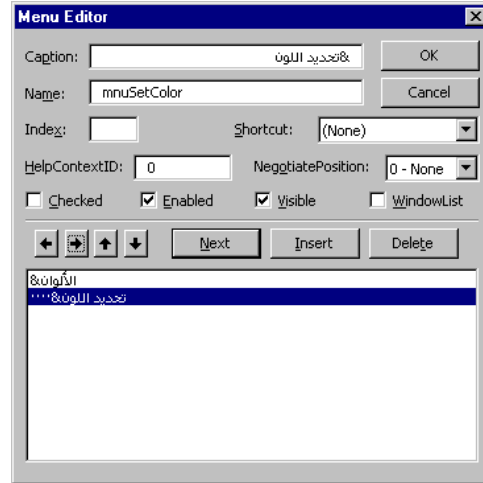
استجابة لذلك، يزاح بند تحديد اللون إلى اليمين. (تظهر ثلاث نقاط على يساره).

يفترض الآن أن يبدو إطار *Menu Editor* كما في الشكل ٧-٥.

الشكل ٧-٥

إنشاء بند تحديد اللون

في قائمة الألوان.



يضع فيجول بيسك خطأً قصيراً تحت الحرف "ت"، نتيجة لاستخدام الرمز & في كتابة قيمة الخاصية Caption لبند تحديد اللون. وعند تشغيل البرنامج يؤدي الضغط على حرف "ت" بعد إظهار قائمة الألوان، إلى نفس النتيجة التي تنشأ عن اختيار بند تحديد اللون عن طريق الفأرة.

استخدم الخطوات التالية لإنشاء البند الثاني في قائمة الألوان:

□ انقر الزر **Next** في الإطار Menu Editor.

يستجيب فيجول بيسك بإضاءة السطر التالي.

□ اكتب "&خروج" في الحقل النصي Caption من الإطار Menu Editor.

- ❑ اكتب "mnuExit" في الحقل النصي Name من الإطار Menu Editor. يستجيب فيجول بيسك بإضافة هذا البند آلياً تحت سابقه، بمعنى أنك لست بحاجة إلى إزاحته كما فعلت مع البند الأول. هذا كل شيء بالنسبة لقائمة الألوان. والآن جاء دور إنشاء قائمة الحجم. تتضمن هذه القائمة حسب ما أوضحه الشكل ٣-٥ ، ثلاثة عناصر تحكم قائمة هي:
- الحجم (عنوان القائمة).
- صغير (بند قائمة).
- كبير (بند قائمة).

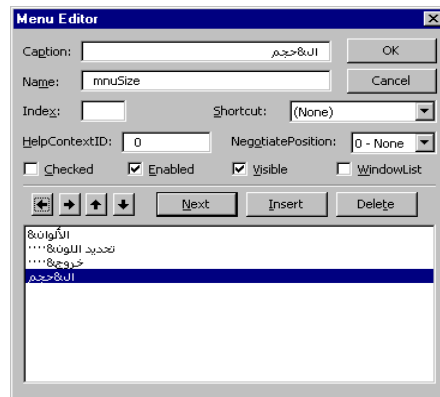
استخدم الخطوات التالية لإنشاء عنوان قائمة الحجم:

- ❑ انقر الزر Next في الإطار Menu Editor. يستجيب فيجول بيسك بإضاءة السطر التالي لسطر آخر بند تم إنشاؤه (بند خروج).
- ❑ اكتب "ال&حجم" في الحقل النصي Caption.
- ❑ اكتب "mnuSize" في الحقل النصي Name.
- يزيح فيجول بيسك هذا البند تلقائياً، ولكن بما أن هذا البند ليس بنداً عادياً، بل هو عنوان قائمة، فإنه لا بد من التراجع عن هذه الإزاحة (نقر زر السهم اليساري). اتبع هذه الخطوة لإزالة الإزاحة الموجودة قبل عنوان قائمة الحجم:
- ❑ انقر زر السهم الأيسر في الإطار Menu Editor، مثلما هو مبين في الشكل ٨-٥.

الشكل ٨-٥

إنشاء عنوان

قائمة الحجم.



اتبع الخطوات التالية لإنشاء بندي قائمة الحجم:

❑ انقر الزر **Next** في الإطار Menu Editor.

يستجيب فيجول بيسك بإضاءة السطر التالي لسطر آخر بند تم إنشاؤه (عنوان قائمة الحجم).

❑ اكتب "&صغير" في مربع النص Caption.

❑ اكتب "mnuSize" في الحقل النصي Name.

بما أن البند صغير هو بند في قائمة الحجم، فلا بد من إزاحته، وجعله في المستوى الثاني. اتبع ما يلي:

❑ انقر زر السهم اليميني في الإطار Menu Editor. (تظهر ثلاث نقاط على يسار البند صغير).

❑ استخدم الآن الخطوات التالية لإنشاء البند الثاني في قائمة الحجم:

❑ انقر الزر **Next** في الإطار Menu Editor.

❑ اكتب "&كبير" في مربع النص Caption.

❑ اكتب "mnuSize" في مربع النص Name.

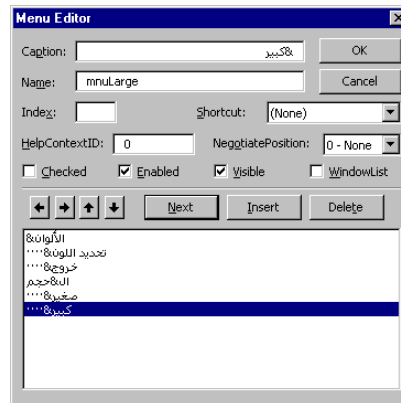
لا داعي لإزاحة هذا البند، لأن فيجول بيسك يعتبره بنفس مستوى البند السابق له، وهو المطلوب.

أصبحت قائمة الحجم جاهزة هي الأخرى! ويفترض أن يظهر الإطار Menu Editor كما في الشكل ٥-٩.

الشكل ٥-٩

إنشاء بنود

قائمة الحجم.



لم تنته مرحلة تصميم القائمة بعد، ولكن دعنا نشاهد ما أنجزناه حتى هذه اللحظة!

❑ انقر الزر **Ok** في الإطار Menu Editor.

❑ احفظ العمل المنجز باختيار البند **Save Project** من قائمة **File**.

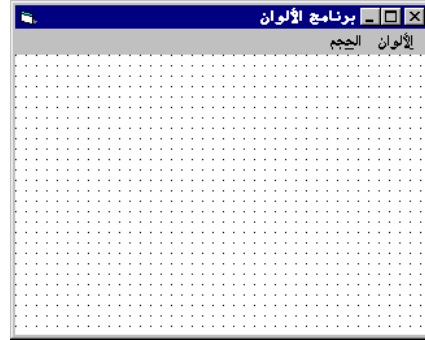
يوضح الشكل ١٠-٥ كيف يظهر النموذج frmColors حتى الآن.

الشكل ١٠-٥

النموذج frmColors

مع شريط قائمة

(مرحلة التصميم).



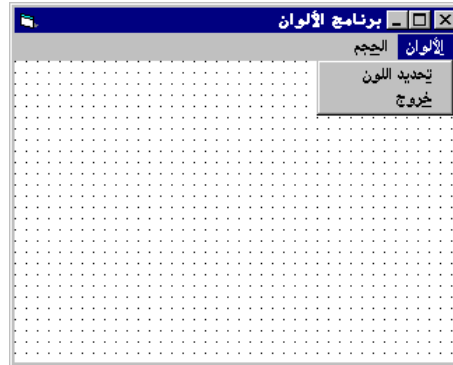
❑ انقر على عنوان قائمة الألوان فوق شريط القوائم، تتسدل عندها لائحة ببند هذه

القائمة (انظر الشكل ١١-٥).

الشكل ١١-٥

بنود قائمة الألوان

(مرحلة التصميم).



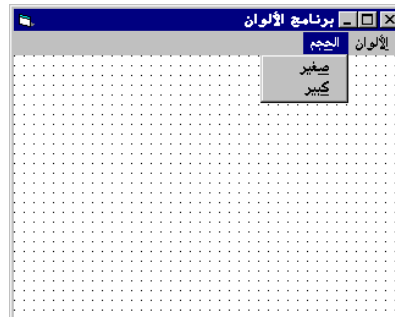
❑ انقر على عنوان قائمة الحجم فوق شريط القوائم، لفتح هذه القائمة (انظر الشكل ٥

١٢-).

الشكل ١٢-٥

قائمة الحجم

(مرحلة التصميم).



دعنا ننفذ برنامج الألوان:

□ نفذ برنامج الألوان. (اضغط المفتاح F5).

يظهر إطار برنامج الألوان. وكما تلاحظ يحتوى إطار البرنامج على شريط قوائم، يحمل عنواني قائمتين: قائمة الألوان وقائمة الحجم.

□ انقر عنوان قائمة الألوان.

يستجيب برنامج الألوان بإظهار قائمة الألوان.

□ انقر عنوان قائمة الحجم.

يستجيب برنامج الألوان بإظهار قائمة الحجم.

تستطيع طبعاً نقر أي بند من بنود القائمتين لاختياره، ولكن لن يحدث أي شيء، والسبب طبعاً أننا لم نكتب حتى الآن أي نص برمجي، ولم نربط بعد أي نص برمجي مع بنود القائمتين.

□ انقر على الرمز الذي يظهر في أقصى الزاوية اليمنى العليا من برنامج الألوان

(إشارة x)، وذلك لإنهاء عمل البرنامج.

يستجيب برنامج الألوان وينهي نفسه. (طبعاً، قمنا بهذا العمل لأننا لم نربط أي نص برنامج مع البند خروج، ولهذا لا فائدة حالياً من نقر بند خروج الموجود في قائمة الألوان لأنه لن يؤدي إلى إنهاء تنفيذ البرنامج).

إنشاء قائمة ثانوية

سوف نستأنف الآن عملية التصميم المرئي لنظام القائمة في برنامج الألوان. لنبدأ الآن

بقائمة الألوان (انظر الشكل ٥-١١).

تتضمن هذه القائمة بندين هما:

تحديد اللون.

خروج.

سنعمل لاحقاً، على ربط البند خروج بنص البرنامج المناسب، بحيث يؤدي النقر على هذا البند إلى قيام البرنامج بإنهاء نفسه. ما الذي سيحدث لدى قيام المستخدم بالنقر

على بند **تحديد اللون**؟ الجواب هو ظهور قائمة أخرى (تدعى قائمة ثانوية أو فرعية Submenu). سنشرع فوراً ببنائها.

يجب أن تكون محتويات قائمة تحديد اللون الثانوية كالتالي:

أحمر.

أزرق.

أبيض.

□ يجب طبعاً أن تكون نافذة النموذج frmColors منتقاة. اختر **Menu Editor** من القائمة **Tools** لفيجول بيسك.

يستجيب فيجول بيسك بإظهار الإطار *Menu Editor*.

البند **أحمر** هو أحد بنود القائمة الفرعية التابعة للبند **تحديد اللون**. ولهذا سنحتاج إلى حشره (إضافته) بين بند **تحديد اللون** والبند **خروج**.

□ اختر البند **خروج** (دع الإضاءة تتوضع عنده). لأنك على وشك حشر بند إضافي في السطر الذي يقع فوق البند **خروج**.

□ انقر الزر **Insert** في الإطار *Menu Editor*.

يستجيب فيجول بيسك بحشر سطر غامق تحت البند **تحديد اللون**.

□ اكتب "أح&مر" في مربع النص **Caption**.

□ اكتب "mnuRed" في مربع النص **Name**.

كما تلاحظ، يقع البند **أحمر** الآن في نفس مستوى بند **تحديد اللون**، إلا أنه يجب إزاحته، لأنه في الواقع ينتمي إلى قائمة **تحديد اللون** الثانوية.

استخدم الخطوة التالية لإزاحة البند **أحمر**.

□ انقر زر السهم اليميني في الإطار *Menu Editor*.

استخدم الخطوات التالية لإضافة البند **أزرق**:

□ اختر البند **خروج**، ثم انقر الزر **Insert** في الإطار *Menu Editor*، لإضافة

وإضاءة سطر غامق (خالي) تحت بند القائمة الثانوية المسمى **أحمر**.

□ اكتب "أز&ق" في مربع النص **Caption**.

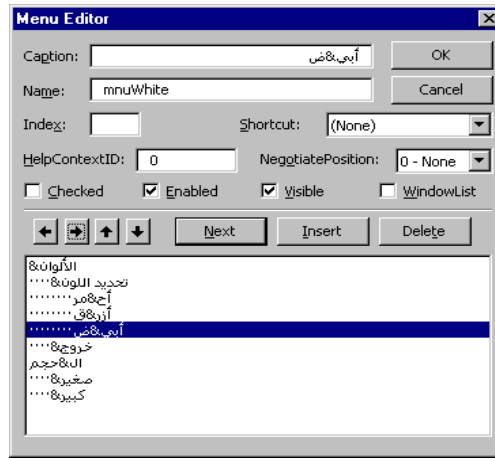
- ❑ اكتب "mnuBlue" في مربع النص Name.
- ❑ انقر زر السهم اليميني في الإطار Menu Editor لوضع البند أزرق تحت نفس مستوى إزاحة البند أحمر.
- اتباع الخطوات التالية لإضافة البند أبيض:
- ❑ اختر البند خروج، ثم انقر الزر Insert في الإطار Menu Editor، لإضافة وإضاءة سطر غامق (خالي) تحت بند القائمة الثانوية المسمى أزرق.
- ❑ اكتب "أبي&ض" في حقل النص Caption.
- ❑ اكتب "mnuWhite" في حقل النص Name.
- ❑ انقر زر السهم اليميني في الإطار Menu Editor لإزاحة البند أبيض إلى نفس سوية البندين أحمر و أزرق.
- ❑ كرر نفس الخطوات لإضافة البندين صغير وكبير إلى قائمة الحجم.
- تهانينا! لقد انتهينا من بناء قائمة برنامج الألوان! ويفترض أن يظهر الإطار Menu Editor كما في الشكل ٥-١٣.
- استخدم الخطوات التالية للخروج من Menu Editor:
- ❑ انقر الزر Ok في الإطار Menu Editor.
- يستجيب فيجول بيسك بإغلاق Menu Editor، ويظهر النموذج frmColors. وكما تشاهد تم ربط القائمة التي انتهيت لتوك من تصميمها بالنموذج.
- اكتمل الآن التصميم المرئي لبرنامج الألوان.
- ❑ احفظ المشروع باختيار البند Save Project من قائمة File.

الشكل ٥-١٣

إطار Menu Editor

بعد اكتمال قوائم

برنامج الألوان.



إنشاء قائمة وفقاً لجدول معين

لقد شرحنا خطوة خطوة، كيفية بناء قوائم برنامج الألوان، لأنه أول برنامج لك في هذا الصدد. لا يمكننا طبعاً الاستمرار على هذا المنوال في الفصول المقبلة، وإنما سنلجأ من الآن فصاعداً إلى تزويدك بجدول يساعدك على إنشاء تلك القوائم.

يتألف جدول القائمة من عمودين، يدرج في العمود اليميني عناوين Captions ومستويات إزاحة البنود، بينما تدرج أسماء البنود (أسماء عناصر تحكم النافذة) في العمود اليساري.

يوضح الجدول ٥-٢ نموذجاً لجدول قائمة. وسنكتشف بسرعة، أن هذا الجدول هو جدول برنامج الألوان.

الجدول ٥-٢. جدول قائمة برنامج الألوان.

العنوان Caption	الخاصية Name
&الألوان	mnuColors
...&تحديد اللون	mnuSetColors
.....&أحمر	mnuRed
.....&أزرق	mnuBlue
.....&أبيض	mnuWhite
...&خروج	mnuExit
ال&حجم	mnuSize
...&صغير	mnuSmall
...&كبير	mnuLarge

إدخال نص برنامج الألوان

يمتلك كل بند من بنود القائمة التي صممناها للتو، (بغض النظر عن مستواه) حادثة نقر Click Event. تتفد هذه الحادثة لدى اختيار بند القائمة المرافق لها، فمثلاً تتفد الحادثة mnuExit_Click() تلقائياً، عند اختيار البند خروج من قائمة الألوان. يبدأ اسم الإجراء لهذه الحادثة بالرموز mnuExit، التي تمثل الاسم الذي أطلقته على بند القائمة خروج في الإطار Menu Editor.

رغم أن الحادثة تدعى Click (أي نقر) إلا أنها تتفد سواء تم اختيار البند بنقره بواسطة الفأرة، أو ضغط أحرف الوصول لهذا البند، بواسطة لوحة المفاتيح. لإظهار الإجراء الخاص ببند قائمة ما، انقر على ذلك البند (في مرحلة التصميم). مثلاً، انقر على البند خروج، إذا أردت مشاهدة الإجراء الذي ينفذ عند اختيار هذا البند.

هناك طريقة أخرى لإظهار الإجراء الخاص ببند قائمة ما، وتكون بإظهار إطار نص البرنامج (Code Window) التابع للنموذج، بالنقر المزدوج على أي مكان في النموذج، ثم اختيار بند القائمة المطلوب، من مربعي السرد الموجودين في قمة إطار نص البرنامج.

اكتب الآن نص برنامج الألوان:

□ تحقق أن قسم التصاريح العامة General Declarations للنموذج frmColors يحوي العبارة Option Explicit، أي:

```
يجب التصريح عن كل المتحولات'
Option Explicit
```

سيطلب منك في الخطوة التالية إدخال نص البرنامج داخل الإجراء Form_Load(). ولإظهار هذا الإجراء انقر نقراً مزدوجاً على النموذج، ثم ضع الإضاءة في مربع السرد اليساري على Form، وضع مربع السرد الأيمن على Load، فيُظهر فيجول بيسك حينها الإجراء Form_Load().

□ أدخل نص البرنامج التالي في الإجراء Form_Load():

```
Private Sub Form_Load()
    ' بما أن لون الإطار الابتدائي هو الأبيض'
    ' فيجب تعطيل البند "أبيض" في القائمة'
    mnuWhite.Enabled = False

    ' بما أن حجم الإطار الابتدائي هو صغير'
    ' فيجب تعطيل البند "صغير" في القائمة'
    mnuSmall.Enabled = False
End Sub
```

ملاحظة

سيطلب منك في الخطوة التالية إدخال نص البرنامج للإجراء mnuRed_Click(). يمكنك إظهار هذا الإجراء، بإظهار النموذج frmColors، ثم اختيار بند تحديد اللون من قائمة الألوان، وأخيراً اختيار البند أحمر من القائمة الثانوية التي ستظهر.

أو تستطيع إظهار الإجراء `mnuRed_Click()` باتباع الخطوات التالية:

- إظهار إطار نص البرنامج.
- وضع مربع السرد اليساري من مربع إطار نص البرنامج على `mnuRed`.
- وضع مربع السرد اليميني من إطار نص البرنامج على `Click`.

□ أدخل نص البرنامج التالي في الإجراء `mnuRed_Click()`:

```
Private Sub mnuRed_Click()  
    'تغيير لون النموذج إلى الأحمر'  
    frmColors.BackColor = vbRed  
  
    'بما أن لون الإطار صار أحمر'  
    'يجب تعطيل البند "أحمر" وتفعيل'  
    'البندين "أبيض" و "أزرق" في القائمة'  
    mnuRed.Enabled = False  
    mnuBlue.Enabled = True  
    mnuWhite.Enabled = True  
End Sub
```

□ أدخل نص البرنامج التالي في الإجراء `mnuBlue_Click()`:

```
Private Sub mnuBlue_Click()  
    'تغيير لون النموذج إلى الأزرق'  
    frmColors.BackColor = vbBlue  
  
    'بما أن لون الإطار صار أزرق'  
    'يجب تعطيل البند "أزرق" وتفعيل'  
    'البندين "أبيض" و "أحمر" في القائمة'  
    mnuBlue.Enabled = False  
    mnuRed.Enabled = True  
    mnuWhite.Enabled = True  
End Sub
```

□ أدخل نص البرنامج التالي في الإجراء `mnuWhite_Click()`:

```
Private Sub mnuWhite_Click()  
    'تغيير لون النموذج إلى الأبيض'
```



```
frmColors.BackColor = vbWhite

' بما أن لون الإطار صار أبيض
' يجب تعطيل البند "أبيض" وتفعيل
' البندين "أحمر" و "أزرق" في القائمة
mnuWhite.Enabled = False
mnuBlue.Enabled = True
mnuRed.Enabled = True
```

End Sub

□ أدخل نص البرنامج التالي في الإجراء :mnuSmall_Click()

```
Private Sub mnuSmall_Click()
    ' تغيير حجم النموذج إلى صغير
    frmColors.WindowState = vbNormal

    ' تعطيل بند القائمة صغير
    mnuSmall.Enabled = False

    ' تفعيل بند القائمة كبير
    mnuLarge.Enabled = True
```

End Sub

□ أدخل نص البرنامج التالي في الإجراء :mnuLarge_Click()

```
Private Sub mnuLarge_Click()
    ' تغيير حجم النموذج إلى كبير
    frmColors.WindowState = vbMaximized

    ' تعطيل بند القائمة كبير
    mnuLarge.Enabled = False

    ' تفعيل بند القائمة صغير
    mnuSmall.Enabled = True
```

End Sub

□ أدخل نص البرنامج التالي في الإجراء :mnuExit_Click()

```
Private Sub mnuLarge_Click()
```

End

End Sub

□ اختر البند **Save Project** من القائمة **File** لحفظ العمل المنجز حتى الآن.

تنفيذ برنامج الألوان

نُفذ برنامج الألوان وامل على اختيار مختلف بنود القائمة، ستلاحظ المزايا التالية أثناء تنفيذك للبرنامج:

يكون لون البند أبيض المنتمي إلى القائمة الثانوية **تحديد اللون**، بلون باهت (بمعنى أنه معطلاً أو ليس فعالاً) عند تنفيذ البرنامج. وهذا طبيعي، لأن لون النموذج الافتراضي هو اللون الأبيض عند بداية التشغيل. ولذلك لا فائدة من إجراء تغيير لون النموذج إلى اللون الأبيض طالما أن لونه بالأساس هو اللون الأبيض. يقاس نفس الشيء بالنسبة للبند **صغير**، المنتمي إلى قائمة **الحجم**. وسيكون لونه عند بداية التشغيل باهتاً، لأن النموذج صغير بالأصل. يغير النموذج لونه بعد اختيار أحد ألوان قائمة البند **تحديد اللون**. ويصبح لون البند الذي اخترته من قائمة **تحديد اللون** باهتاً (بلون رمادي باهت). □ اختر البند **خروج** من قائمة **الألوان**، لإنهاء عمل برنامج الألوان.

كيف يعمل برنامج الألوان

يستخدم برنامج الألوان الإجراء `Form_Load()`، وأيضاً الحادثة `Click` لبنود القوائم المختلفة.

نص برنامج الإجراء `Form_Load()`:

يعتبر الإجراء `Form_Load()` من أول الإجراءات التي تنفذ آلياً عند بدء تشغيل أي برنامج مطور في فيجول بيسك. يمكنك استخدام الإجراء `Form_Load()` لإنجاز مختلف مهام الابتدء التي ترغب أن ينجزها برنامجك عند بداية التنفيذ.

في مثالنا هذا، يفترض أن تكتب نص برنامج الإجراء Form_Load()، بشكل يلغي عمل (يعطل) البند أبيض المنتمي إلى القائمة الثانوية تحديد اللون. كما يلغي عمل البند صغير المنتمي إلى قائمة الحجم. يعطل الإجراء Form_Load() عمل البندين أبيض و صغير بإسناد القيمة False للخاصية Enabled لكلا البندين.

```
Private Sub Form_Load()
    ' بما أن لون الإطار الابتدائي هو الأبيض
    ' فيجب تعطيل البند "أبيض" في القائمة
    mnuWhite.Enabled = False

    ' بما أن حجم الإطار الابتدائي هو صغير
    ' فيجب تعطيل البند "صغير" في القائمة
    mnuSmall.Enabled = False
End Sub
```

نص برنامج الإجراء mnuRed_Click()

ينفذ الإجراء mnuRed_Click() لدى اختيار البند أحمر من القائمة الثانوية للبند تحديد اللون. ويؤول لون النموذج إلى اللون الأحمر، ويتم تعطيل البند أحمر في قائمة البند تحديد اللون:

```
Private Sub mnuRed_Click()
    ' تغيير لون النموذج إلى الأحمر
    frmColors.BackColor = vbRed

    ' بما أن لون الإطار صار أحمر
    ' يجب تعطيل البند "أحمر" وتفعيل
    ' البندين "أبيض" و "أزرق" في القائمة
    mnuRed.Enabled = False
    mnuBlue.Enabled = True
    mnuWhite.Enabled = True
End Sub
```

يعمل الإجراء `mnuRed_Click()` على تعطيل البند أحمر. وتفعيل البندين أزرق و أبيض، وباعتبار أن لون النموذج صار أحمر، فمن المفترض أن يكون لدى المستخدم القدرة على تغيير لون النموذج إلى الأزرق أو الأبيض. يعتبر عمل الإجراءات `mnuRed_Click()` و `mnuWhite_Click()` مشابهاً جداً لعمل الإجراء `mnuRed_Click()`.

نص برنامج الإجراء `mnuSmall_Click()`

ينفذ الإجراء `mnuRed_Click()` لدى اختيار البند صغير من قائمة الحجم، ويؤدي إلى تغيير حجم النموذج إلى مقاس صغير. ثم يعطل عمل البند صغير.

```
Private Sub mnuSmall_Click()  
    ' تغيير حجم النموذج إلى صغير  
    frmColors.WindowState = vbNormal  
  
    ' تعطيل بند القائمة صغير  
    mnuSmall.Enabled = False  
  
    ' تفعيل بند القائمة كبير  
    mnuLarge.Enabled = True  
End Sub
```

تُستخدم الخاصية `WindowState` لتحديد حجم النموذج، فالنموذج يتخذ الحجم الافتراضي أو الطبيعي (`Normal`) (الحجم الافتراضي هو الحجم الذي يظهر به النموذج أثناء مرحلة التصميم)، عندما تكون قيمة الخاصية `WindowState` للنموذج مساوية إلى الصفر (نفس قيمة الثابت `vbNormal`). بينما يتخذ النموذج الحجم الأعظمي (تكبير النموذج لكامل مساحة الشاشة) عند إسناد القيمة ٢ (نفس قيمة الثابت `vbMaximized`) للخاصية `WindowState`.

يُعطّل الإجراء `mnuSmall_Click()` عمل البند صغير، ويفعل البند كبير، لأنه لا معنى أن يكون البند صغير فعالاً عندما يكون النموذج بحجم صغير أصلاً، وإنما يفترض أن يكون البند كبير هو البند الفعال لإتاحة تكبير حجم النموذج.

يتشابه عمل الإجراء `mnuLarge_Click()` مع الإجراء `mnuSmall_Click()`.

نص برنامج الإجراء mnuExit_Click()

ينفذ الإجراء mnuExit_Click() لدى اختيار البند خروج من قائمة الألوان، ويؤدي إلى إنهاء تنفيذ البرنامج:

```
Private Sub mnuLarge_Click()  
    End  
End Sub
```

إضافة مفاتيح الاختزال

يمكننا تحسين عمل برنامج الألوان، بإضافة مفاتيح الاختزال. تمكّنك مفاتيح الاختزال من اختيار بند قائمة ما، بالضغط على مفاتيح الاختزال المتعلقة بهذا البند من لوحة المفاتيح.

فمثلاً يمكننا إرفاق البند أحمر (الموجود في قائمة الألوان) بمفتاح الاختزال Ctrl+R، يسمح هذا العمل بالضغط على Ctrl+R لتغيير لون النموذج، بدلاً من النقر على البند تحديد اللون بواسطة الفأرة ثم النقر على البند أحمر لتغيير لون النموذج إلى الأحمر. استخدم الخطوات التالية لإرفاق البند أحمر بالمفتاح الاختزالي Ctrl+R:

❑ اختر النموذج frmColors.

❑ اختر البند Menu Editor من قائمة Tools.

يستجيب فيجول بيسك بإظهار الإطار Menu Editor.

❑ اختر البند أحمر من المربع الموجود أسفل الإطار Menu Editor.

يستجيب فيجول بيسك بإظهار خصائص البند أحمر في الجزء العلوي من الإطار Menu Editor.

لاحظ أن الحقل المدعو Shortcut في الجزء العلوي، يحوي القيمة None. أي لا يوجد مفاتيح اختزال مرفقة مع هذا البند.

استخدم الخطوات التالية لتحديد Ctrl+R كمفاتيح اختزال للبند أحمر:

❑ انقر على السهم النازل الموجود يمين الحقل Shortcut في الإطار Menu

Editor، ثم اختر القيمة Ctrl+R.

□ انقر الزر **Ok** لإغلاق الإطار Menu Editor.

استخدم الخطوات التالية لمشاهدة مفتاح الاختزال Ctrl+R الذي أرفقته بالبند أحمر قيد التنفيذ:

□ نفذ برنامج الألوان.

□ اضغط على المفاتيح Ctrl+R.

يستجيب برنامج الألوان بتبديل لون النموذج إلى الأحمر.

كرر الخطوات الآتية الذكر، لإرفاق مفتاح الاختزال Ctrl+B بالبند أزرق، ومفتاح الاختزال Ctrl+W بالبند أبيض.

يدرك المستخدمون أن بوسعهم الضغط على Ctrl+R، للحصول على نفس الأثر الناتج عن اختيار البند أحمر من القائمة، وذلك لأن نص مفتاح الاختزال Ctrl+R يظهر على يسار البند أحمر. وبشكل مشابه، تظهر العبارة النصية Ctrl+B على يسار البند أزرق، وتظهر العبارة Ctrl+W على يسار البند أبيض (انظر الشكل ١٤-٥)، ويؤدي الضغط على كل منهما من لوحة المفاتيح إلى اختيار البند المرافق له.

الشكل ١٤-٥

مفاتيح الاختزالات Shortcuts

تظهر على يسار بنود

القائمة تحديد اللون.



ملاحظة

لا يوفر فيجول بيسك (فعلياً، ولا النظام ويندوز نفسه) مفاتيح اختزال عربية، لذلك لا يمكننا إسناد مفتاح الاختزال Ctrl+م مثلاً، إلى بند القائمة أحمر.

إضافة خط فاصل بين بنود القائمة

يعتبر الخط الفاصل Separator Bar عبارة عن خط أفقي يفصل بين بنود القائمة الواحدة. والغاية منه هي إعطاء منظر جمالي للقائمة. يوضح الشكل ١٥-٥ قائمة الألوان بعدما أضيف إليها خط فاصل بين بند تحديد اللون والبند خروج.

الشكل ١٥-٥

خط فاصل Separator Bar

في قائمة الألوان.



اتبع الخطوات التالية لإضافة خط فاصل بين بند تحديد اللون والبند خروج:

- ❑ اختر النموذج frmColors.
- ❑ اختر البند Menu Editor من قائمة Tools.
- يستجيب فيجول بيسك بإظهار الإطار Menu Editor.
- ❑ ضع الإضاءة على البند خروج، ثم انقر الزر Insert في الإطار Menu Editor.
- يستجيب فيجول بيسك بإضافة سطر فارغ فوق البند خروج.
- ❑ اكتب إشارة الناقص العادية "-" في الحقل النصي Caption من الإطار Menu Editor.
- ❑ اكتب munSep1 في الحقل النصي Name من الإطار Menu Editor.
- ❑ انقر الزر Ok من الإطار Menu Editor لإغلاقها.
- لاحظ أن رمز إشارة الناقص العادية "-" الذي كتبت في الحقل Caption يمثل رمز الخط الفاصل Separator Bar.
- لاحظ أيضاً أنك أطلقت على الخط الفاصل الاسم mnuSep1، فإذا أردت إضافة المزيد من الخطوط الفاصلة، فسوف يكون بمقدورك تسميتها mnuSep2، mnuSep3 ... الخ.

طبعاً تستطيع إطلاق أي أسماء أخرى تريدها على الخطوط الفاصلة، لكننا نحاول دائماً استخدام أسماء تسمح للمرء تمييز مهمة أي كائن برمجي من خلال اسمه. لا يمكننا اختيار الخط الفاصل أثناء تنفيذ البرنامج، ولهذا فإن الإجراء `mnuSep1_Click()` لن يُنفذ أبداً. (السؤال الذي قد يتبادر إلى الذهن، ما هي الحاجة إلى تسمية بند قائمة لن ينفذ أبداً؟. الجواب: يتطلب فيجول بيسك تسمية كل بنود القائمة مهما كان نوعها). □ نفذ برنامج الألوان.

كما تشاهد لقد ظهر الخط الفاصل الذي أضفته للتو.

كيفية جعل أحد عناصر تحكم القائمة غير مرئي

لقد كانت عملية تفعيل أو تعطيل أحد بنود القائمة، تتم بكتابة التعليمة التي تسند إحدى القيمتين `True` أو `False` إلى الخاصية `Enabled` التابعة لأحد عناصر تحكم القائمة. يظهر البند الممثل لعنصر تحكم القائمة، عند تعطيله بلون باهت (عندما تكون الخاصية `Enabled` مساوية للقيمة `False`). وعندها لا يمكن اختياره، إلا أن تعطيل أحد البنود، لا يمنع من استمرار ظهوره في القائمة. قد نحتاج في بعض البرامج إلى إخفاء أحد بنود القائمة كلياً، وليس مجرد جعله باهتاً. تستخدم الخاصية `Visible` لتحقيق هذا الغرض. فمثلاً لإخفاء بند قائمة يدعى `mnuMyItem` خلال مرحلة التنفيذ، استخدم العبارة التالية:

```
mnuMyItem.Visible = False
```

سوف يختفي البند `mnuMyItem` بعد تنفيذ هذه العبارة، وتزاح كل بنود القائمة الواقعة تحته إلى الأعلى، لملء الفراغ الذي تركه. (تظهر القائمة للمستخدم وكأن هذا البند غير موجود أصلاً).

يمكن جعل كل بنود القائمة غير مرئية مرة واحدة، وذلك بإسناد القيمة `False` للخاصية `Visible` التابعة لعنوان هذه القائمة. فتختفي عندها هذه القائمة، وتزاح كل عناوين القائمة الواقعة على يسارها إلى اليمين لملء الفراغ الذي تركه هذا العنوان.

جرب كتابة التعليمة التالية في الإجراء `:Form_Load()`

```
mnuColors.Visible = False
```

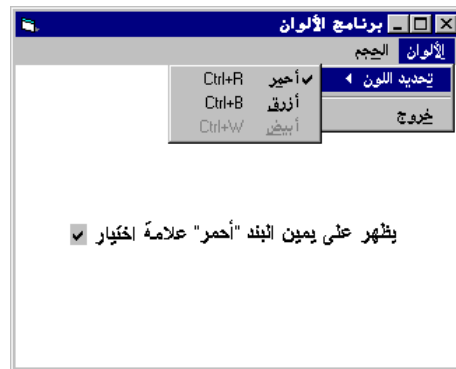

سوف يختفي عنوان قائمة الألوان بالإضافة لجميع بنوده، وكأن قائمة الألوان غير موجودة أصلاً.

استخدام علامات الاختيار Check Marks

يلزمنا في بعض البرامج وضع إشارة على أحد بنود القائمة. يمكنك استخدام علامات الاختيار لإنجاز هذه الغاية. يوضح الشكل ١٦-٥ أحد بنود قائمة تحديد اللون مع علامة اختيار بجانبه.

الشكل ١٦-٥

علامة اختيار على
يسار البند أحمر.



تستخدم الخاصية Checked لوضع أو رفع علامة الاختيار. فعند إسناد القيمة True للخاصية Checked التابعة لأحد عناصر تحكم القائمة، توضع علامة اختيار بجانب ذلك البند، أما إسناد القيمة False للخاصية Checked فيؤدي إلى إزالة علامة الاختيار من جانب هذا البند. فمثلاً استخدم العبارة التالية لوضع علامة اختيار بجانب البند أحمر.

```
mnuRed.Checked = True
```

أما لإزالة علامة الاختيار من جانب البند أحمر، فاستخدم العبارة التالية:

```
mnuRed.Checked = False
```

(لا يستخدم برنامج الألوان علامات الاختيار لبنود القائمة الموجودة فيه).

لاحظ أن أحد مربعات الاختيار التي تظهر في الجزء العلوي من الإطار Menu Editor، يدعى Checked. ووضع علامة اختيار في مربع الاختيار هذا، يعني أن بند القائمة المضاء (المنتقى) حالياً في الجزء السفلي من الإطار Menu Editor، سوف يظهر وعلى يمينه علامة اختيار (يمكنك إزالة هذه العلامة من خلال التعليمة التي تسند القيمة False للخاصية Checked التابعة لنفس البند).

برنامج القائمة المتغيرة

سنكتب الآن برنامجاً يدعى برنامج القائمة المتغيرة، حيث يوضح كيفية إضافة البنود أو إزالتها من القائمة خلال مرحلة التنفيذ. دعنا نحدد أولاً ما يفترض أن يقوم به برنامج القائمة المتغيرة، ثم نتابع بعد ذلك كتابة البرنامج.

يظهر عند بدء التشغيل شريط قائمة يتضمن العنوان **تغيير**، حسب ما يبينه الشكل ٥-١٧.

الشكل ٥-١٧

برنامج القائمة المتغيرة.



تحتوي القائمة **تغيير** في بداية التشغيل على ثلاثة بنود هي على التوالي: **إضافة بند**، و **إزالة بند**، و **خروج**، وخط فاصل. (انظر الشكل ٥-١٨). لاحظ أن بند القائمة **إزالة**، يظهر بلون باهت.

الشكل ٥-١٨

القائمة تغيير بدون بنود مضافة.



يتم إضافة بند جديد إلى القائمة **تغيير**، كلما اخترت البند **إضافة بند**. فمثلاً يوضح الشكل ٥-١٩ كيف تبدو القائمة **تغيير** بعد اختيار **إضافة بند** ثلاث مرات.

الشكل ٥-١٩

القائمة تغيير مع

ثلاثة عناصر مضافة.



يتم حذف البند الأخير من القائمة تغيير كلما اخترت البند إزالة بند.

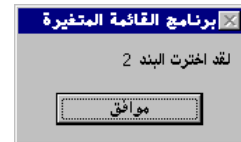
يظهر عند اختيار أي من العناصر التي أضفتها، رسالة تحمل اسم البند الذي اخترته.

فمثلاً عند اختيار البند 2، تظهر الرسالة المبينة بالشكل ٥-٢٠.

الشكل ٥-٢٠

الرسالة التي تظهر بعد اختيار

البند 2 من القائمة تغيير.



التصميم المرئي لبرنامج القائمة المتغيرة

نبدأ كعادتنا، بتصميم نموذج البرنامج:

□ ابدأ مشروعاً من النوع التنفيذي القياسي Standard EXE.

□ احفظ نموذج المشروع باسم Grow.Frm في الدليل C:\VB5Prg\Ch05، واحفظ

ملف المشروع باسم Grow.Vbp في ذات الدليل.

□ أنشئ النموذج وفق الجدول ٥-٣.

الجدول ٥-٣. جدول خصائص النموذج frmGrow.

القيمة	الخاصية	الكائن
FrmGrow	Name	Form
برنامج القائمة المتغيرة	Caption	
True	RightToLeft	
White	BackColor	

إنشاء قائمة برنامج القائمة المتغيرة

سنبدأ الآن بربط القائمة مع النموذج frmGrow:

□ اختر النموذج frmGrow.

□ اختر البند Menu Editor من قائمة Tools.

يستجيب فيجول بيسك بإظهار الإطار Menu Editor.

□ صمم القائمة وفق الجدول ٥-٤.

لاحظ أن البند إضافة بند يمتلك ثلاث نقاط قبله، مما يعني أنه مزاح يمينياً داخل الإطار Menu Editor، لاحظ أيضاً أن البنود إزالة بند و خروج والخط الفاصل (-) مزاحين أيضاً إلى اليمين في الإطار Menu Editor، مثلما كانت الحال عليه مع البنود تحديد اللون و خروج و صغير و كبير، المبينين في الشكل ٥-٩.

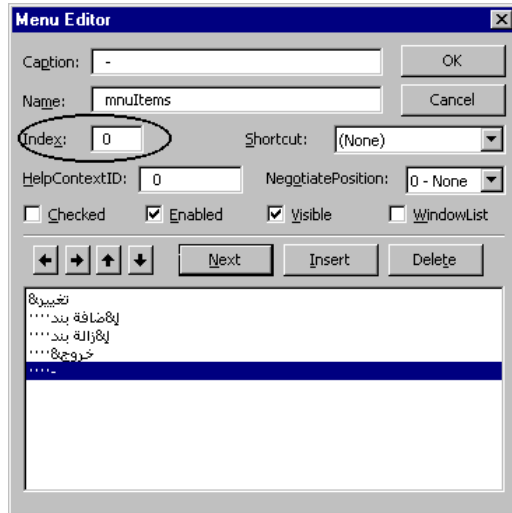
الجدول ٥-٤. جدول قائمة النموذج frmGrow.

العنوان	الاسم
&تغيير	MnuGrow
...إضافة بند	MnuAdd
...إزالة بند	MnuRemove
...خروج	MnuExit
...	mnuSep1

- ❑ ضع الإضاءة فوق آخر بند من القائمة (الخط الفاصل)، ثم اكتب القيمة صفر في مربع النص Index (انظر الشكل ٢١-٥).
 - ❑ انقر الزر **Ok** في الإطار Menu Editor.
- انتهت الآن مرحلة التصميم المرئي لبرنامج القائمة المتغيرة.

الشكل ٢١-٥

إسناد القيمة صفر للخاصية Index
التابعة للخط الفاصل.



إنشاء مصفوفة عناصر تحكم القائمة

لقد أدى إسناد القيمة صفر للخاصية Index التابعة للخط الفاصل، إلى ما يدعى بمصفوفة عناصر تحكم القائمة. سوف تدعى هذه المصفوفة بالاسم mnuItems، لأنك كتبت الاسم mnuItems في الخاصية Name.

تتألف المصفوفة mnuItems حتى هذه اللحظة، من بند واحد هو الخط الفاصل Separator Bar (أي بمعنى أن البند mnuItems(0) هو بند الخط الفاصل).

سيضيف برنامجنا مع كل اختيار للبند إضافة بند من القائمة تغيير، أثناء تنفيذ البرنامج بنداً جديداً إلى المصفوفة mnuItems. ونتيجة لذلك، سيظهر المزيد من بنود القائمة تحت بند الخط الفاصل.

إدخال نص برنامج القائمة المتغيرة

سندخل الآن نص برنامج القائمة المتغيرة كالتالي:

□ اكتب النص التالي في قسم التصاريح العامة للنموذج frmGrow:

```
' يجب التصريح عن كل المتحولات في البرنامج'
Option Explicit

'التصريح عن المتحول gLastElement
Dim gLastElement As Integer
```

ملاحظة

إذا واجهت صعوبة في إضافة نص في قسم التصاريح العامة، جرب ما يلي: ضع مؤشر الفأرة عند بداية التعليمة Option Explicit ثم اضغط المفتاح Enter لعدة مرات. سيضاف نتيجة لذلك عدة أسطر فارغة إلى قسم التصاريح العامة، تستطيع الآن إدخال النص الإضافي في قسم التصاريح العامة هذا.

□ اكتب النص التالي في الإجراء Form_Load():

```
Private Sub Form_Load()
    'إعطاء قيمة ابتدائية للمتحول gLastElement
    gLastElement = 0

    'عند تشغيل البرنامج لأول مرة لا يوجد أية بنود
    'يمكن إزالتها لذلك يتم تعطيل البند إزالة
    mnuRemove.Enabled = False
End Sub
```

□ اكتب النص التالي في الإجراء mnuAdd_Click():

```
Private Sub mnuAdd_Click()
    'زيادة المتحول gLastElement
    gLastElement = gLastElement + 1

    'إضافة عنصر جديد للمصفوفة mnuItems[]
    Load mnuItems(gLastElement)
    'تحديد العنوان للبند الجديد
    mnuItems(gLastElement).Caption = _
        "البند " + Str(gLastElement)
```

```

' بما أنه تم إضافة بند جديد للقائمة المتغيرة
' لذلك يجب تفعيل البند إزالة بند
mnuRemove.Enabled = True

```

End Sub

□ اكتب النص التالي في الإجراء :mnuRemove_Click()

```

Private Sub mnuRemove_Click()
'إزالة آخر بند من المصفوفة mnuItems
Unload mnuItems(gLastElement)

'إنقاص قيمة المتحول gLastElement
gLastElement = gLastElement - 1

'عندما لا يتبقى في المصفوفة أية بنود
'يجب عندها تعطيل البند إزالة
If gLastElement = 0 Then
    mnuRemove.Enabled = False
End If

```

End Sub

□ اكتب النص التالي في الإجراء :mnuItems_Click()

```

Private Sub mnuItems_Click(Index As Integer)
'التصريح عن المتحول الذي سيحوي الرسالة
Dim myMsg As String

'التصريح عن المتحول الذي سيحوي عنوان الرسالة
Dim myTitle As String

'عرض الرسالة التي تبين البند الذي اختاره المستخدم
myTitle = "برنامج القائمة المتغيرة"
myMsg = "لقد اخترت البند" + Str(Index)
MsgBox myMsg, vbOKOnly + vbMsgBoxRtlReading + _
vbMsgBoxRight, myTitle

```

End Sub

إذا أردت الوصول إلى الإجراء `mnuItems_Click()`، أظهر إطار نص البرنامج، ثم اختر من مربع السرد اليساري العلوي البند `mnuItems`، واختر من مربع السرد اليميني العلوي البند `Click`.

□ اكتب النص التالي في الإجراء `mnuExit_Click()`:

```
Private Sub mnuExit_Click()  
    End  
End Sub
```

□ احفظ المشروع الآن باختيار البند **Save Project** من قائمة **File**.

تنفيذ برنامج القائمة المتغيرة

نفذ البرنامج ثم اختر البنود المختلفة من القائمة تغيير، لاحظ الظواهر التالية أثناء تنفيذ البرنامج:

يظهر البند إزالة بند بلون باهت عند بدء تشغيل البرنامج (لا يكون هذا البند متاحاً في البداية، لأنه لا توجد بنود مضافة حتى هذه اللحظة، يمكنك حذفها).
يضاف لدى كل اختيار للبند إضافة بند، بند جديد إلى القائمة تغيير، باسم (بند ×).
حيث × هو رقم تسلسلي يعطى لاسم البند الجديد.
يصبح البند إزالة بند فعالاً حالما يضاف بند واحد على الأقل إلى القائمة تغيير.
يؤدي اختيار البند إزالة بند في كل مرة، إلى إزالة آخر بند مضاف إلى القائمة،
ويصبح هذا البند (أي إزالة بند) معطلاً عندما تخلو القائمة من جميع البنود المضافة. (تم حذفها جميعاً).

□ اختر البند خروج من القائمة، لإنهاء عمل البرنامج.

كيف يعمل برنامج القائمة المتغيرة

يستخدم برنامج القائمة المتغيرة المتحول `gLastElement` للاحتفاظ بعدد البنود في المصفوفة `mnuItems`. ولأن هذا المتحول يجب أن يكون مرئياً (متاحاً) لكل الإجراءات في النموذج، فقد صرحنا عنه في قسم التصاريح العامة للنموذج `frmGrow`:


```

يجب التصريح عن كل المتحولات في البرنامج'
Option Explicit

'tصريح عن المتحول gLastElement
Dim gLastElement As Integer

```

نص الإجراء Form_Load()

يُنَفَّذُ الإجراء Form_Load() آلياً، عند بداية تشغيل البرنامج. لاحظ أن المتحول gLastElement في هذا الإجراء، قد تم إسناد القيمة الابتدائية صفر له، كما أن البند إزالة بند قد أُبْطِلَ عمله:

```

Private Sub Form_Load()
    إعطاء قيمة ابتدائية للمتحول gLastElement
    gLastElement = 0

    عند تشغيل البرنامج لأول مرة لا يوجد أية بنود
    يمكن إزالتها لذلك يتم تعطيل البند إزالة
    mnuRemove.Enabled = False
End Sub

```

تعمل التعليمة الأولى في هذا الإجراء على إسناد القيمة الابتدائية صفر، للمتحول gLastElement. لأن المصفوفة mnuItems تمتلك بنداً واحداً فقط، وهو البند ذي الرقم صفر الذي تم تشكيله أثناء مرحلة التصميم (الخط الفاصل). أما التعليمة الثانية في هذا الإجراء، فإنها تعمل على تعطيل البند إزالة بند، لأنه لا يوجد بند يمكن إزالته عند بداية تشغيل البرنامج.

نص الإجراء mnuAdd_Click()

ينفذ الإجراء mnuAdd_Click() عند اختيار البند إضافة بند من قائمة تغيير. وهذا الإجراء مسؤول عن إضافة بند جديد إلى المصفوفة mnuItems وإسناد عنوان مناسب للخاصية Caption التابعة لهذا البند:

```

Private Sub mnuAdd_Click()
    'زيادة المتحول gLastElement
    gLastElement = gLastElement + 1

    'إضافة عنصر جديد للمصفوفة mnuItems[]
    Load mnuItems(gLastElement)

    'تحديد العنوان للبند الجديد
    mnuItems(gLastElement).Caption = _
        "البند " + Str(gLastElement)

    'بما أنه تم إضافة بند جديد للقائمة المتغيرة
    'لذلك يجب تفعيل البند إزالة بند
    mnuRemove.Enabled = True
End Sub

```

أول شيء فعله هذا الإجراء كان زيادة قيمة المتحول gLastElement بمقدار ١:

```
gLastElement = gLastElement + 1
```

ثم أضاف بنداً جديداً إلى المصفوفة mnuItems بواسطة تعليمة Load:

```
Load mnuItems(gLastElement)
```

كما أسند عنواناً مناسباً للخاصية Caption التابعة للبند الجديد بواسطة التعليمة التالية:

```

mnuItems(gLastElement).Caption = _
    "البند " + Str(gLastElement)

```

وأخيراً تم تفعيل البند إزالة بند:

```
mnuRemove.Enabled = True
```

هذه الخطوة الأخيرة سببها أن الإجراء أضاف للتو بنداً إلى قائمة تغيير، ولهذا يفترض أن يكون المستخدم قادراً على إزالته.

نص الإجراء mnuRemove_Click()

ينفذ الإجراء mnuRemove_Click() عند اختيار البند إزالة بند من القائمة تغيير:

```

Private Sub mnuRemove_Click()
    'إزالة آخر بند من المصفوفة mnuItems

```

```

Unload mnuItems(gLastElement)

'إنقاص قيمة المتحول gLastElement
gLastElement = gLastElement - 1

'عندما لا يتبقى في المصفوفة أية بنود
'يجب عندها تعطيل البند إزالة
If gLastElement = 0 Then
    mnuRemove.Enabled = False
End If

```

End Sub

يحذف هذا الإجراء آخر عنصر في المصفوفة mnuItems بواسطة التعليمة: Unload

```
Unload mnuItems(gLastElement)
```

يلي ذلك إنقاص قيمة المتحول gLastElement بمقدار ١. ثم يتم التحقق إذا كان البند المتبقي هو البند ذي الرقم صفر، فإذا تحقق الشرط فهذا يعني وجوب تعطيل البند إزالة بند:

```

If gLastElement = 0 Then
    mnuRemove.Enabled = False
End If

```

نص الإجراء mnuItems_Click()

ينفذ الإجراء mnuItems_Click() عند اختيار أي من البنود المضافة الجديدة الموجودة في القائمة تغيير:

```
Private Sub mnuItems_Click(Index As Integer)
```

```

'التصريح عن المتحول الذي سيحوي الرسالة
Dim myMsg As String

'التصريح عن المتحول الذي سيحوي عنوان الرسالة
Dim myTitle As String

'عرض الرسالة التي تبين البند الذي اختاره المستخدم

```

```
myTitle = "برنامج القائمة المتغيرة"
myMsg = " لقد اخترت البند " + Str(Index)
MsgBox myMsg, vbOKOnly + vbMsgBoxRtlReading + _
vbMsgBoxRight, myTitle
```

End Sub

يستشعر الوسيط Index في هذا الإجراء، بالبند الذي تم اختياره. فمثلاً يساوي الوسيط Index إلى القيمة ١ عند اختيار البند المضاف الأول **بند 1**. باعتبار أن **البند 1** هو البند ذي القيمة 1 في المصفوفة .mnuItems. كما يساوي هذا الوسيط (أي Index) إلى القيمة 2 لدى اختيار **البند 2**، وهكذا تتغير قيمة الوسيط Index تلقائياً حسب البند المختار.

الخلاصة

تعلمت في هذا الفصل كيف تكتب برنامجاً يحتوي على قائمة، وكيف تربط هذه القائمة بالنموذج، باستخدام الإطار Menu Editor، وكيف تربط عناصر تحكم القائمة بنصوص البرنامج الخاصة بها.

كما تعلمت عن الخصائص المختلفة التي تناولها الفصل، مثلاً Checked, Visible التي تعطيك القدرة والمرونة على تبديل المظهر الذي تبدو به القائمة خلال مرحلة التنفيذ. ولا تنس أيضاً مصفوفة عناصر تحكم القائمة، حيث أوضح هذا الفصل آلية إنشائها وكيف تتمكن بواسطتها من إضافة بنود إلى القائمة أو حذفها وذلك خلال مرحلة التنفيذ.