الباب الرابع عشر

تطبيقات صناديق الحوار والنماذج

Dialog-based Applications and Forms

مفتتح

فى هذا الباب ، سوف نتعلم كيفية تصميم البرامج التى تعتمد على صناديق الحوار وكذلك برامج النماذج (Forms). أما البرنامج الذى يعتمد على صندوق الحوار فهو أحد إمكانات الساحر التى يقدمها لنا عندما نبدأ فى إنشاء مشروع جديد. وهو برنامج لا يحتوى على شىء إلاّ صندوق حوار ، ولذلك فإن هذه النوعية من البرامج تصلح فقط لبرامج المنافع أو البرامج التى لا تتطلب القوائم وسطور الأدوات وقضبان الانزلاق ، وسائر خصائص البرامج العامة. أما برامج النماذج ، فهى برامج عامة تتميز بما تتميز به سائر التطبيقات ، فيما عدا أنها تعتمد على الوراثة من فصيلة النماذج CFormView والتى تمنح المشهد الشكل العام للنموذج الذى يحتوى على الأزرار بأنواعها وصناديق الاختبار وسائر أدوات التحكم.

تذكر:

١. للاطلاع على الصيغة التفصيلية لإحدى الدوال أو الفصائل ، ضع مؤشر الفأر فوقها ثم اضغط زر اللوحة F1.

٢. لعرض الخريطة الكاملة لشجرة فصائل المؤسسة MFC ابحث فى شاشة النجدة (باستخدام البطاقة Search) عن "Hierarchy Chart".

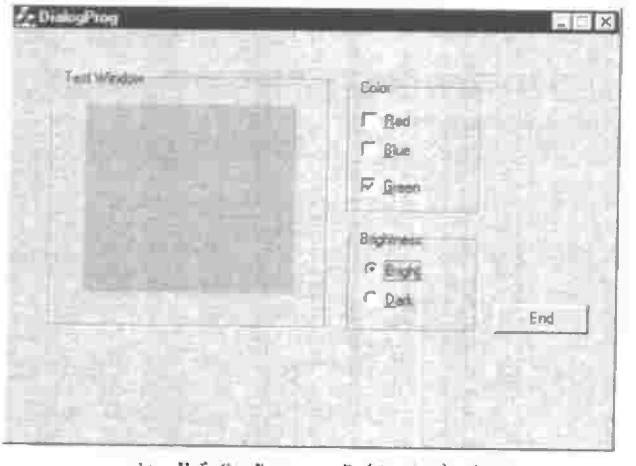
٣. للاطلاع على الكود الكامل لأحد المشروعات ، افتح المشروع من القرص المصاحب للكتاب.

(١٤-١) مشروع برنامج صندوق حوار

فى هذا المشروع سوف ننشئ تطبيقاً مبنياً على صندوق حوار (Dialog Box) ، وسوف تلاحظ أنه يحتوى على الكثير من الإمكانات الجاهزة التى لا تحتاج إلى برمجة أو إعداد. وفى الشكل التالى نرى الشكل النهائى للتطبيق المطلوب إعداده ، فهو عبارة عن صندوق حوار يحتوى على أقسام ثلاثة:

- قسم لعرض واختبار اللون (Test Window)
- قسم للتحكم فى الألوان الأساسية(Color): الأحمر والأزرق والأخضر ، من خلال صناديق اختبار. وفى إمكانك أن تختار لوناً أو أكثر. وعندما تختار جميع الألوان تحصل على اللون الأبيض. أما إذا كانت صناديق الاختيار كلها فارغة تحصل على اللون الأسود.
- قسم للتحكم فى درجة الإضاءة (Brightness).

ويوجد المشروع على القرص المصاحب للكتاب تحت الاسم DialogProg.

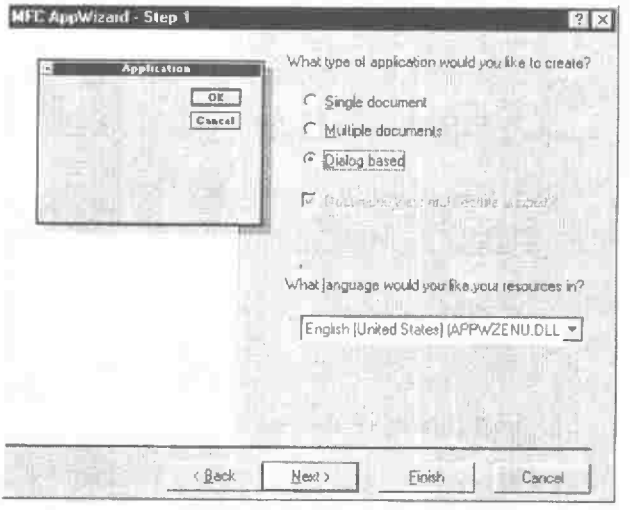


شكل (١٤-١) الصورة النهائية للبرنامج

الخطوة الأولى: إنشاء هيكل البرنامج

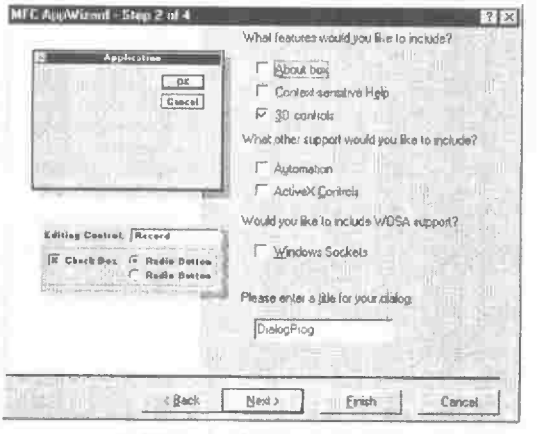
ابدأ مشروعا جديداً بالاسم DialogProg واتبع الخطوات التالية من خطوات الساحر:

- فى الخطوة الأولى ، اختر زر الراديو Dialog based.



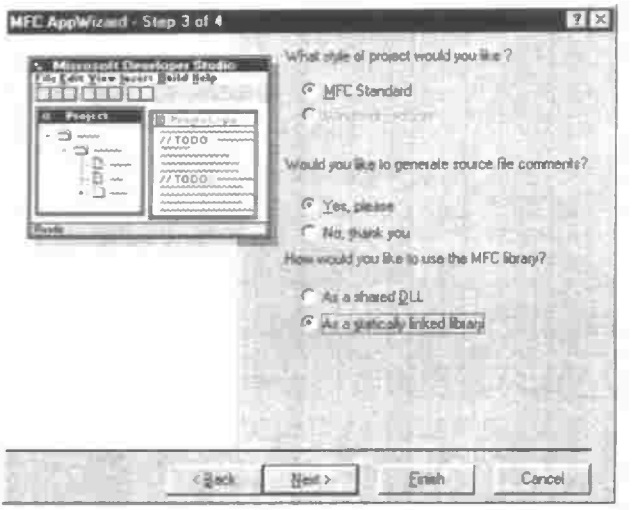
شكل (١٤-٢) الخطوة الأولى

- فى الخطوة الثانية ، احذف الاختيارات About box و ActiveX Controls من صناديق الاختيار.



شكل (١٤-٣) الخطوة الثانية

- فى الخطوة الثالثة ، احذف الاختيار As Shared DLL ، واختر بدلاً منه As statistically linked library.



شكل (١٤-٤) الخطوة الثالثة

- فى الخطوة الرابعة والأخيرة ، سوف ترى أسماء الفصائل التى ولدها الساحر: فصيلة التطبيق CDialogProgApp وفصيلة الحوار CDialogProgDlg.

شكل (١٤-٥) الخطوة الرابعة والأخيرة

• ويوضح الشكل التالى مشهد الفصائل فى بيئة الأستوديو حيث نرى الفصائل التى ولدها الساحر.

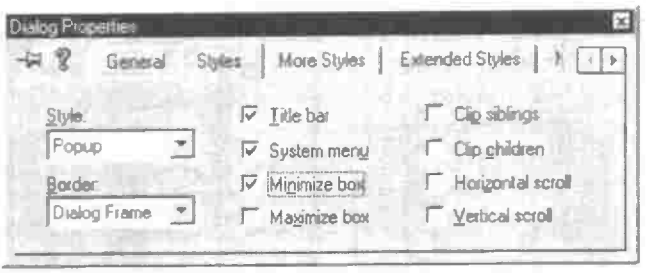
شكل (١٤-٦) مشهد الفصائل

أما فى القسم الأيمن من الشاشة فإنك عادة ترى صندوق الحوار الفارغ جاهزاً على التعديل. وإذا كنت قد غيرت المشهد بالتجول فى أرجاء الأستوديو ، فيمكنك التوصل إلى صندوق الحوار من مشهد الموارد كالعادة.

الخطوة الثانية: إنشاء أدوات التحكم

١. اضغط بالزر الأيمن على أى نقطة فى صندوق الحوار (فيما عدا أدوات التحكم الموجودة به) ثم اعرض صندوق الخصائص للصندوق (Dialog Properties).

٢. اضغط على بطاقة الطراز (Styles) لتعديل طرازات أدوات التحكم كما هى موضحة بالشكل التالى. لاحظ أن التغيير الوحيد الذى أضفناه هو اختيار الخاصية Minimized التى تمكنك من تصغير النافذة.

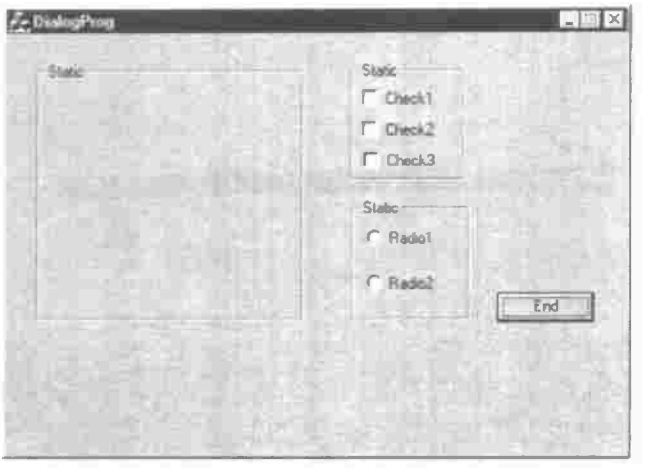


شكل (١٤-٧) طرازات أدوات التحكم بالصندوق

٣. احذف الزر OK من صندوق الحوار باختياره ثم الضغط على زر اللوحة Delete. احذف أيضاً الزر "To Do..".

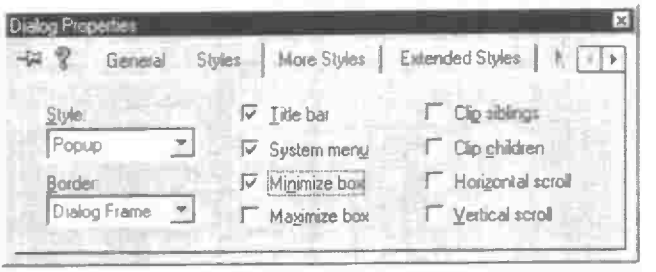
٤. غير اسم الزر Cancel إلى End وذلك لاستخدامه فى إنهاء البرنامج. يتم ذلك من خلال صندوق الخصائص للزر (البطاقة General).

٥. أضف أدوات التحكم الموضحة بالشكل التالى.



شكل (١٤-٨) صندوق الحوار بعد إضافة أدوات التحكم

٦. قم بتغيير أسماء أدوات التحكم وفقاً للمعلومات الموضحة بالجدول التالى للشكل. لاحظ أن العنصر الأول من كل مجموعة يأخذ الخاصية "group".

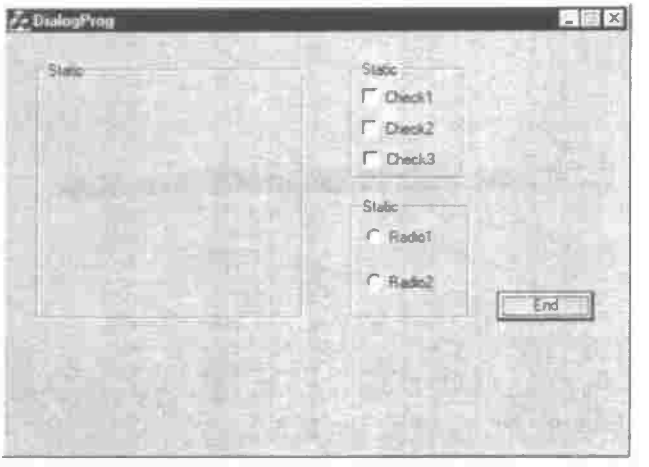


شكل (١٤-٧) طرازات أدوات التحكم بالصندوق

٣. احذف الزر OK من صندوق الحوار باختياره ثم الضغط على زر اللوحة Delete. احذف أيضاً الزر "To Do..".

٤. غير اسم الزر Cancel إلى End وذلك لاستخدامه فى إنهاء البرنامج. يتم ذلك من خلال صندوق الخصائص للزر (البطاقة General).

٥. أضف أدوات التحكم الموضحة بالشكل التالى.



شكل (١٤-٨) صندوق الحوار بعد إضافة أدوات التحكم

٦. قم بتغيير أسماء أدوات التحكم وفقاً للمعلومات الموضحة بالجدول التالى للشكل. لاحظ أن العنصر الأول من كل مجموعة يأخذ الخاصية "group".

ملاحظة:

يمكنك ، بعد وضع الأدوات فوق صندوق الحوار ، أن تقوم بتنظيمها بالاستعانة باختيار القائمة Layout - Align.

:ملاحظة

الخطوة الثالثة: تعريف المتغيرات الأعضاء

١. افتح شاشة ساحر الفصائل ، واختر منها بطاقة البيانات الأعضاء (Member Variables) ، ثم اختر الفصيلة CDialogProgDlg من صندوق الفصائل (Class name). أضف المتغيرات الأعضاء المناظرة لأرقام التعارف الموضحة بالجدول التالى ، وذلك بالاستعانة بالزر Add Variable مع قبول الأوضاع سابقة التعريف.

النوع	المتغير	رقم التعارف
BOOL	m_Blue	IDC_BLUE
BOOL	m_Red	IDC_RED
BOOL	m_Green	IDC_GREEN
int	m_Bright	IDC_BRIGHT

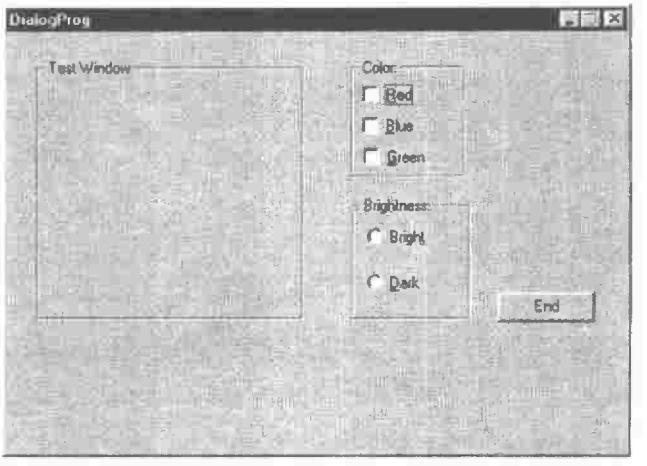
جدول(١٤-٢) المتغيرات الأعضاء في صندوق الحوار

لاحظ أن الأدوات الإستاتيكية للمجموعات لا تناظر أية متغيرات ، كما أن هناك متغيراً واحداً فقط يمثل أزرار الراديو ، لأنها لا تعمل جميعاً فى نفس الوقت ، هذا هو المتغير Bright من النمط int. وبحسب قيمة المتغير يتحدد اختيار زر معين من أزرار الراديو. والشكل التالى يوضح نافذة المتغيرات.

رقم التعارف ID	نوع أداة التحكم	الاسم	الخاصية
IDC_COLOR	مجموعة	Color:	
IDC_RED	صندوق اختبار	&Red	Group
IDC_BLUE	صندوق اختبار	&Blue	
IDC_GREEN	صندوق اختبار	&Green	
IDC_INTENSITY	مجموعة	Brightness:	
IDC_BRIGHT	زر راديو	Brigh&t	Group
IDC_DARK	زر راديو	&Dark	
IDC_SAMPLE		Test Window	

جدول (١٤-١) أرقام التعارف وخصائص صندوق الحوار

٧. غير ترتيب الوقفات باستخدام أمر القائمة Layout-Tab order بحسب رغبتك. فى النهاية سوف يبدو الصندوق كما بالشـــكل التالى.



شكل (١٤-٩) صندوق الحوار فى الصورة النهائية

٣. جرب البرنامج عند هذا الحد وتأكد أنه يعـرض الصنـدوق المطلوب.

الخطوة الرابعة: تعديل الكود

١. **ملف العناوين DialogProgDlg.h:** افتح الملف ثم أضف إليه الكود الموضح بالبنط الثقيل.

```
class CDialogProgDlg : public CDialog
{
// Construction
public:
	CDialogProgDlg(CWnd* pParent = NULL);   // standard
constructor

public:
	CRect m_Sample;
	enum{INT_DARK, INT_BRIGHT};
```

فى هذا الكود قمنا بإعلان المستطيل m_Sample كهدف من النمـط **CRect** وهو يمثل المستطيل الذى سوف يظهر بـه اللـون والـذى يحمل الاسم Test Window (وهو يماثل النمط النوافذى) **RECT**.

يستخدم النمط RECT لتمثيل المستطيلات وهو عبارة عن منشـأ صيغته كالتالى:

```
typedef struct tagRECT {
  LONG left;
```

```
  LONG top;
  LONG right;
  LONG bottom;
} RECT;
```

أما المنشأ المتعدد **enum** فهو يحتوى على قيمتين تمثـــلان اللـــون الفاتح و اللون الداكن.

٢. **ملف الكود DialogProgDlg.cpp:** افتح الملف ثم أضف كود المقابض المختلفة ، كما هو موضح بالبنط الثقيل فـــى الشـــكل التالى.

```
void CDialogProgDlg::OnRed()
{                                                    حالة اللون الأحمر
 // TODO: Add your control notification handler code here
   m_Red = IsDlgButtonChecked (IDC_RED);
   InvalidateRect (&m_Sample);
   UpdateWindow ();
}
//                                                   حالة اللون الأخضر
void CDialogProgDlg::OnGreen()
{
  // TODO: Add your control notification handler code here
   m_Green = IsDlgButtonChecked (IDC_GREEN);
   InvalidateRect (&m_Sample);
   UpdateWindow ();
}
//                                                   حالة اللون الأزرق
void CDialogProgDlg::OnBlue()
{
   // TODO: Add your control notification handler code here
   m_Blue = IsDlgButtonChecked (IDC_BLUE);
   InvalidateRect (&m_Sample);
   UpdateWindow ();
}
//                                                   حالة اللون الداكن
void CDialogProgDlg::OnDark()
{
```

```
  // TODO: Add your control notification handler code here
  if (IsDlgButtonChecked (IDC_DARK))
    {
    m_Bright = INT_DARK;
    InvalidateRect (&m_Sample);
    UpdateWindow ();
    }
}
//                                                    حالة اللون الفاتح
void CDialogProgDlg::OnBright()
{
  // TODO: Add your control notification handler code here
  if (IsDlgButtonChecked (IDC_BRIGHT))
    {
    m_Bright = INT_BRIGHT;
    InvalidateRect (&m_Sample);
    UpdateWindow ();
    }
}
```

فى شريحة الكود السابقة يختص كل مقبض من المقابض الثلاثة الأولى بلون ما. وتتشابه الدوال الثلاثة فى منطقها. ففى البداية يتم اختبار وجود علامة الاختبار بالصندوق المعين ، وذلك باستخدام الدالة **CWnd::IsDlgButtonChecked()** التى تستخدم رقم تعارف الصندوق كبارامتر. وترجع هذه الدالة قيمة صحيحة إذا كانت هناك علامة اختبار فى الصندوق وإلاّ فإنها ترجع القيمة صفراً. ويتم تخصيص القيمة المرتجعة إلى العضو المنظر للصندوق فيصبح عاملاً TRUE أو غير عامل FALSE.

يلى ذلك استخدام الدالة **CWnd::InvalidateRect()** التى تؤدى إلى إتلاف محتويات صندوق اللون تمهيداً لإعادة طلائه بالدالة **CWnd::UpdateWindow()** وذلك بإرسال الرسالة WM_PAINT.

أما الدوال OnBright() و OnDark() فهى تختبر أزرار الراديو ، وبحسب الزر المختار حالياً يتم تخصيص أحد القيمتين INT_BRIGHT أو INT_DARK إلى المتغير الوحيد الممثل لأزرار الراديو وهو m_Bright الممثل لدرجة الإضاءة. ثم يتم إتلاف محتويات النافذة وإعادة طلائها باللون ذى الإضاءة الجديدة.

٣. **ملف الكود DialogProgDlg.cpp:** غير محتويات الدالة **OnInitDialog()** بإضافة الكود الموضح بالبنط الثقيل فى الشكل التالى:

```
BOOL CDialogProgDlg::OnInitDialog()
{
  CDialog::OnInitDialog();
// Set the icon for this dialog.  The framework does this automatically
// when the application's main window is not a dialog
  SetIcon(m_hIcon, TRUE);                    // Set big icon
  SetIcon(m_hIcon, FALSE);                   // Set small icon
// TODO: Add extra initialization here
  GetDlgItem (IDC_SAMPLE)->GetWindowRect
(&m_Sample);
  ScreenToClient (&m_Sample);
  int Thickness = (m_Sample.right - m_Sample.left) / 12;
  m_Sample.InflateRect (-Thickness, - Thickness);
  return TRUE;  // return TRUE  unless you set the focus to a control
}
```

تؤدى الدالة **OnInitDialog()** إلى تحويل دفة التحكم إلى صندوق الحوار قبل أن يظهر على الشاشة. ولذلك فإنها هى الدالة المناسبة التى نضمنها مواصفات قسم العينة الذى يظهر فيه اللون. ونرى هنا الدوال الآتية:

- الدالة **CWnd::GetDlgItem()** التى تستخدم كمؤشر إلى النافذة الإبنة. وفى هذه الحالة فإنها تشير إلى صندوق العينة (IDC_SAMPLE).
- أما الدالة **CWnd::GetWindowRect()** فإنها تختزن إحداثيات مستطيل صندوق العينة فى العضو m_Sample.
- تستخدم الدالة **CWnd::ScreenToClient()** فى تحويل إحداثيات الشاشة إلى إحداثيات العميل (Client Coordinates) أى إحداثيات منسوبة إلى الركن الأيسر العلوى لصندوق الحوار. وتستخدم الدالة مؤشراً إلى العضو m_Sample كبارامتر لها.
- تم إعلان المتغير Thickness ليعبر عن سمك الإطار الذى يحتوى بداخله على العينة. وقد اخترنا أن يكون السمك $^{1}/_{12}$ من عرض المستطيل ، أى:

(m_Sample.right - m_Sample.left) / 12;

- أما الدالة **Crect::InflateRect()** فهى تستخدم فى إعتام إطار المستطيل بمقدار المتغير Thickness من جميع الجوانب. فالبارامتر الأول يدل على سمك الإطار الأيمن والأيسر ، والبارامتر الثانى يدل على سمك الإطار العلوى والسفلى.

٤. **ملف الكود DialogProgDlg.cpp:** غير محتويات الدالة **OnPaint()** باستخدام الكود الموضح بعد بالبنط الثقيل:

```
void CDialogProgDlg::OnPaint()
{
   if (IsIconic())
  {
```

```
    CPaintDC dc(this); // device context for painting
    SendMessage(WM_ICONERASEBKGND, (WPARAM)
dc.GetSafeHdc(), 0);
    // Center icon in client rectangle
    int cxIcon = GetSystemMetrics(SM_CXICON);
    int cyIcon = GetSystemMetrics(SM_CYICON);
    CRect rect;
    GetClientRect(&rect);
    int x = (rect.Width() - cxIcon + 1) / 2;
    int y = (rect.Height() - cyIcon + 1) / 2;
    // Draw the icon
    dc.DrawIcon(x, y, m_hIcon);
  }
else
  {
// New Code:
    COLORREF MyColor = RGB
        (m_Red ? (m_Bright==INT_DARK ? 200 : 255) : 0,
        m_Green ? (m_Bright==INT_DARK ? 200 : 255) : 0,
        m_Blue ? (m_Bright==INT_DARK ? 200 : 255) : 0);
      CBrush MyBush (MyColor);
      CPaintDC dc(this);
      dc.FillRect (&m_Sample, &MyBush);
// removed: CDialog::OnPaint();
}
```

أول ما نلاحظه أننا قد حذفنا الدالة الموجودة بالسطر الأخير بوضع علامة تعليق أمامها. هذه هى الدالة سابقة التعريف. أما الجزء الجديد الذى أضفناه فهو الجزء **else** من العبارة الشرطية ، والذى يختص بتحديد الألوان المعروضة فى صندوق العينة باستخدام الماكرو **RGB** الذى يأخذ الصورة:

```
COLORREF RGB(
  BYTE bRed,
  BYTE bGreen,
  BYTE bBlue
);
```

كما استخدمنا تعبيراً شرطياً بالصورة "x ? y: z" ، فإذا تحقق الشرط x يأخذ المتغير القيمة y وإلاّ فيأخذ القيمة z. فمثلاً يتم تحديد اللـــون الأحمر كالتالى:

```
m_Red ? (m_Bright==INT_DARK ? 200 : 255) : 0
```

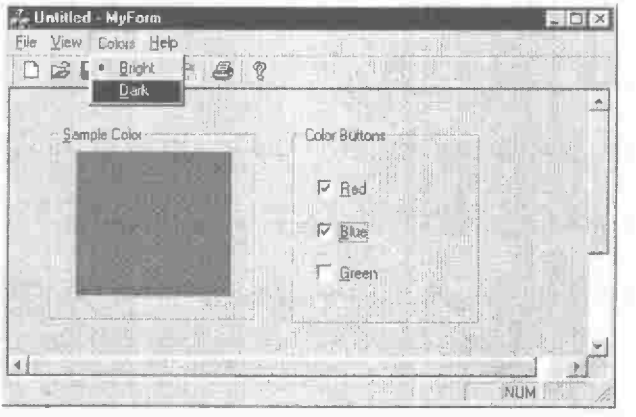
يقول هذا التعبير: إذا تحقق شرط اللون الأحمر (أى كان مختـلراً) ، قيم التعبير الشرطى (الفرعى) الوارد بين القوسين ، وإلاّ فامنحـــــه القيمة صفراً.

أما التعبير الشرطى الفرعى فيقـول: إذا كـــان اللـــون داكنـــا (أى m_Bright==INT_DARK) فامنح المتغير m_Bright القيمـــة ٢٠٠ وإلاّ فلتكن ٢٥٠. والقيمة ٢٥٠ معناها تشبع اللون ، أمـــا القيمـــة ٢٠٠ فمعناها أن يكون اللون غير مشبع ، أى أنه يحتوى على قليل مـــن اللون الأسود الذى يجعله داكنا. وبطبيعة الحال فإن جميع الألـــوان الأخرى سوف تكون أصفاراً حتى نحصل على اللون الأحمر. أمـــا لو كانت هناك ألوان أخرى مختارة فإن اللون النهائى يصبح خليطـاً من الألوان. أما الدالة **CDC::FillRect()** فهى تستخدم لملأ المســـاحة باللون المختار. وهى تستخدم بارامترين الأول عبارة عن مؤشـــــر إلى المستطيل الممثل للمساحة المطلوب تلوينها ، والثانى مؤشر إلى هدف فرشاة (من الفصيلة **CBrush**).

قم بتشغيل البرنامج ، وشاهد النتائج.

(١٤-٢) مشروع نموذج يحتوى على صندوق حوار
Form Project

فى هذا المشروع سوف ننشئ تطبيقاً مماثلاً للتطبيق السابق فيما عدا أن المشروع يأخذ صورة نموذج (Form) ، كما أننا سوف نستخدم اختيارات القائمة لتغيير درجة إضاءة اللون. والمشروع يوجد على القرص المصاحب تحت الاسم MyForm.dsw. وعندما تنفذ هذا المشروع سوف تحصل على الشكل التالى.



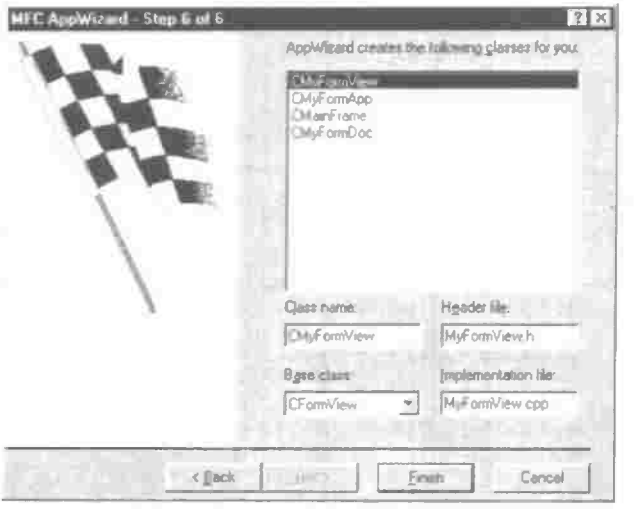
شكل (١٤-١٢) تنفيذ البرنامج MyForm

وكما نرى بالشكل أن النموذج عبارة عن نافذة ذات قائمة وسطر أدوات وقضبان انزلاق ، ولكنها جاهزة على استيعاب أدوات التحكم المختلفة. وفيما يلى نعرض خطوات إنشاء هذا المشروع.

الخطوة الأولى: إعداد مشروع النموذج

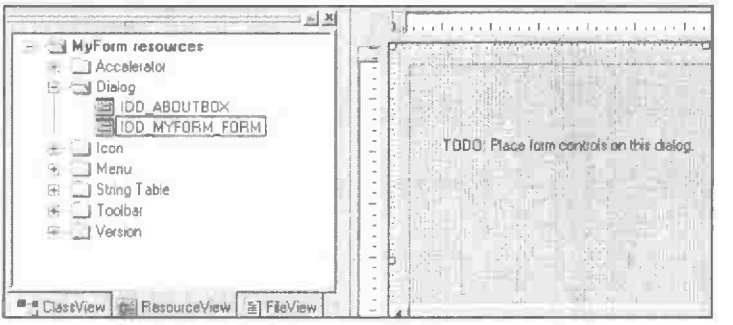
١. قم ببناء مشروع جديد بالطريقة المعتادة التى بنينا بها مشروعات الرسم. فإذا أتيت إلى الخطوة السادسة والأخيرة

اختر الفصيلة CMyFormView من صندوق الفصائل الأربعة الموضح بالشكل ، ثم اختر فصيلة الأساس (Base Class) لتكون CFormView. إن هذه الفصيلة تمنح مشروعك النموذج المطلوب.



شكل (١٤-١٣) الخطوة السادسة من المشروع

٢. قم بتنفيذ المشروع بالصورة الخام لاختباره ، فتشاهد النموذج الفارغ للمشروع. أغلق نافذة النموذج وابدأ فى تعديل النموذج الموضح بالشكل التالى. وعادة يبدأ المشروع بهذه الشاشة التى تحتوى على النموذج ، فإذا لم تجد هذه الشاشة ، افتح مشهد الموارد ، ثم عنصر صناديق الحوار (Dialog) ، ثم اضغط ضغطة مزدوجة على العنصر IDD_MYFORM_FORM.



شكل (١٤-١٤) نافذة المشروع الخام في بيئة الموارد

الخطوة الثانية: إنشاء أدوات التحكم

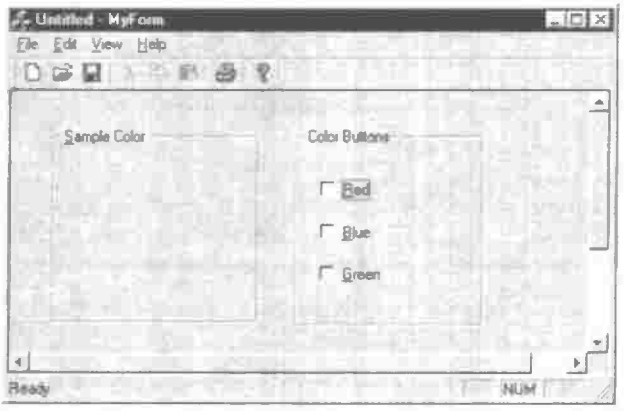
١. امسح العنصر الموجود بالنموذج "..TODO".

٢. باستخدام لوحة أدوات التحكم التى استخدمناها فى البرنامج السابق ، أضف عنصر مجموعة به ثلاثة صناديق اختبار ، وعنصر مجموعة آخر لاحتواء مستطيل العينة. امنح العناصر الأسماء الموضحة بالجدول التالى.

رقم التعارف ID	نوع أداة التحكم	الاسم	الخاصية
IDC_BUTTONS	مجموعة	Color Buttons	
IDC_SAMPLE	مجموعة	Sample Color	
IDC_RED	صندوق اختبار	&Red	Group
IDC_BLUE	صندوق اختبار	&Blue	
IDC_GREEN	صندوق اختبار	&Green	

جدول (١٤-٣) أرقام التعارف وأسماء الأدوات

٣. اضبط ترتيب الوقفات باستخدام أمر القائمة Layout - Tab Order. وعندما تنتهى من رسم الأزرار سوف تبدو شاشتك كما بالشكل التالى.



شكل (١٤-١٥) نافذة المشروع بعد إضافة الأدوات

الخطوة الثالثة: إضافة المتغيرات الأعضاء وعناصر القائمة

١. استخدم ساحر الفصائل (Class Wizard) لإضافة المتغيرات الأعضاء (Member Variables) المناظرة لصناديق الاختبار الثلاثة: IDC_RED و IDC_GREEN و IDC_BLUE ، بنفس الأسلوب المتبع فى البرنامج السابق. مع ملاحظة أن هذه الأعضاء تابعة للفصيلة CMyFormView.

٢. افتح عنصر القائمة (Menu) بمشهد الموارد ، ثم افتح العنصر IDR_MAINFRAME. احذف قائمة التحرير (Edit) حيث أنها بلا عمل فى هذا التطبيق. ثم أضف بدلاً منها قائمة جديدة بالاسم

Colors&. أضف عنصرين إلى القائمة الجديدة همـــا: Bright& و Dark& ، كما هو موضح بالجدول التلى.

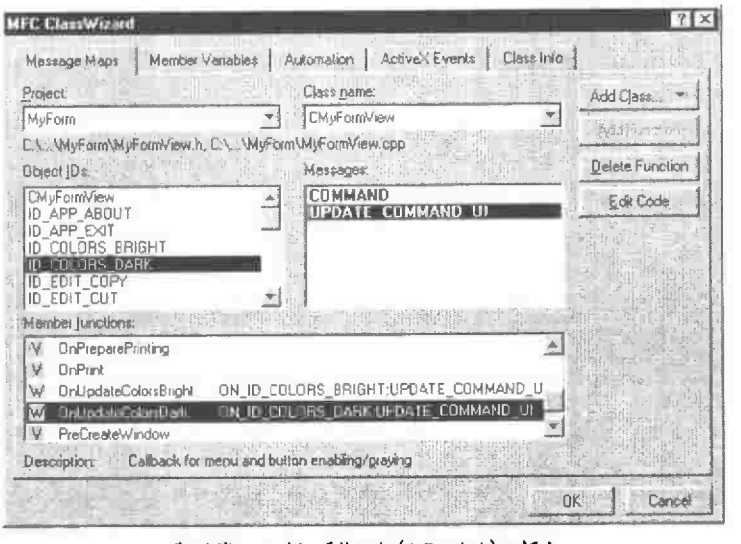
رقم التعارف	اسم عنصر القائمة
ID_COLORS_BRIGHT	&Bright
ID_COLORS_DARK	&Dark

جدول (١٤-٤) اختيارات قائمة الألوان Colors

الخطوة الرابعة: إضافة المقابض

١. افتـــح نـــافذة ســـاحر الفصـــائل ، واخـــتر منـــها الفصيلـــة CMyFormView من نـــافذة الفصيلـــة (Class name:). وأضـــف مقـــابض الرســـائل COMMAND و UPDATE_COMMAND_UI المناظرة لأرقام التعارف الآتية (أنظر الشكل التالى):

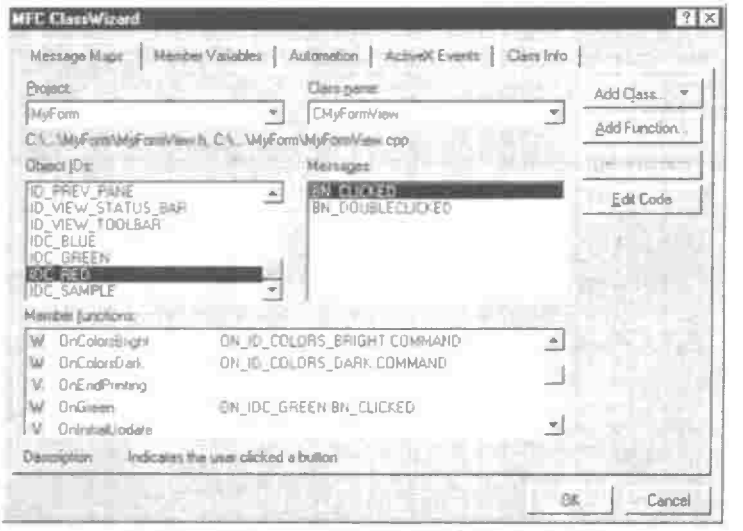
- ID_COLORS_DARK
- ID_COLORS_BRIGHT



شكل (١٤-١٦) إضافة مقابض القائمة

٢. أضف أيضاً مقابض الرسالة **BN_CLICKED** المناظرة لصناديق الاختبار الآتية (أنظر الشكل التالى):

- IDC_RED
- IDC_GREEN
- IDC_BLUE



شكل (١٤-١٧) إضافة مقابض صناديق الاختبار

الخطوة الخامسة: تعديل الكود

١. **ملف العناوين MyFormView.h:** افتح الملف بضغطة مزدوجة على اسم الفصيلة CMyFormView ثم أضف الأعضاء الموضحة بالبنط الثقيل بالشكل التالى:

```
public:
// New Code:                                  أضف هذه الأعضاء:
  int m_Brightness;                  متغير شدة الإستضاءة
  CRect m_MySample;                  متغير مستطيل العينة
  enum {INT_DARK, INT_BRIGHT};       منشأ الصفات : داكن و فاتح
```

٢. **ملف الكود MyFormView.cpp**: افتح ملف دالة البناء CMyFormView() ثم أضف إلي نهايتها العبارة التالية لشحن متغير شدة الاستضاءة (Brightness) بقيمة ابتدائية.

```
// New Code:
m_Brightness = INT_BRIGHT;
}
```

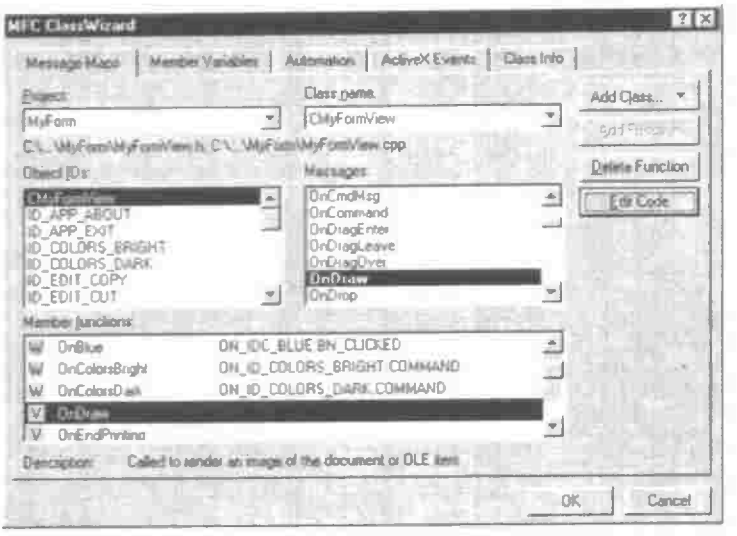
٣. **ملف الكود MyFormView.cpp**: افتح ملف الدالة **OnInitialUpdate()** ثم أضف إليها الكود الموضح بالبنط الثقيل بالشكل التالى.

```
void CMyFormView::OnInitialUpdate()
{
  CFormView::OnInitialUpdate();
  GetParentFrame()->RecalcLayout();
  ResizeParentToFit();
// New Code:
  GetDlgItem (IDC_SAMPLE)->GetWindowRect
(&m_MySample);
  ScreenToClient (&m_MySample);
  int Thickness = (m_MySample.right - m_MySample.left) /
10;
  m_MySample.InflateRect (-Thickness, -Thickness);
}
```

والكود المستخدم هنا ، هو نفسه الكود الذى أضفناه من قبل فى البرنامج السابق. وقد اخترنا هنا أن تكون قيمة سمك الإطار

(Thickness) عشر عرض المستطيل. وهى قيمة تقديرية علـــى أى حال.

٤. **ملف الكود MyFormView.cpp:** سوف نضيف إلى هذا الملف الدالة **()OnDraw** باستخدام الساحر ، حيث أنها ليست موجـودة من الأصل. ولتحقيق ذلك اختر الفصيلـــة CMyFormView مـــن نافذة الساحر كما هو موضح بالشكل التالى. ثم اخـــتر الدالـــة **()OnDraw** من نافذة الرسائل (Messages) ، ثم اضغط الـــزر Add Function.



شكل (١٤-١٨) إضافة الدالة ()OnDraw

افتح ملف الدالة وأضف إليه الكود الموضح بالبنط الثقيل بالشـــكل التالى.

```
void CMyFormView::OnDraw(CDC* pDC)
{
// TODO: Add your specialized code here and/or call the base class
```

```
// New Code:                                    أضف هذه السطور:
  COLORREF MyColor = RGB
    (m_Red ? (m_Brightness==INT_DARK ? 200 : 255) : 0,
    m_Green ? (m_Brightness==INT_DARK ? 200 : 255) : 0,
    m_Blue ? (m_Brightness==INT_DARK ? 200 : 255) : 0);
  CBrush MyBrush (MyColor);
  pDC->FillRect (&m_MySample, &MyBrush);
}
```

ولا جديد فى هذا الكود عما استخدمناه فى البرنامج السابق. ومن الجدير بالذكر أنه يمكنك التحكم فى درجة استضاءة اللون بتغيير العدد ٢٠٠ إلى أى قيمة أخرى. جرب وشاهد.

٥. **ملف الكود MyFormView.cpp**: أما الكود الموضح بالشكل التالى فهو ما سنضيفه إلى الدوال الثلاثة الخاصة بالألوان:

- OnRed()
- OnGreen()
- OnBlue()

وهى جميعاً إضافات متشابهة.

```
void CMyFormView::OnBlue()
{
// TODO: Add your control notification handler code here
// New Code:                                    أضف هذه السطور
  m_Blue = IsDlgButtonChecked (IDC_BLUE);
  CClientDC ClientDC (this);
  OnPrepareDC (&ClientDC);
  CRect Rect = m_MySample;
  ClientDC.LPtoDP (&Rect);
  InvalidateRect (&Rect);
  UpdateWindow ();
}

void CMyFormView::OnGreen()
{
// TODO: Add your control notification handler code here
// New Code:                                    أضف هذه السطور
```

```
  m_Green = IsDlgButtonChecked (IDC_GREEN);
  CClientDC ClientDC (this);
  OnPrepareDC (&ClientDC);
  CRect Rect = m_MySample;
  ClientDC.LPtoDP (&Rect);
  InvalidateRect (&Rect);
  UpdateWindow ();
}

void CMyFormView::OnRed()
{
// TODO: Add your control notification handler code here
// New Code:                                    أضف هذه السطور
  m_Red = IsDlgButtonChecked (IDC_RED);
  CClientDC ClientDC (this);
  OnPrepareDC (&ClientDC);
  CRect Rect = m_MySample;
  ClientDC.LPtoDP (&Rect);
  InvalidateRect (&Rect);
  UpdateWindow ();
}
```

والكود السابق يماثل ما أضفناه إلى برنامج صندوق الحوار فيما عدا الكود الخاص بتحويل الإحداثيات. ففى هذا البرنامج يتم تحويل الإحداثيات من الإحداثيات المنطقية (Logical) إلى إحداثيات الجهاز (Device) كالآتى:

```
ClientDC.LPtoDP (&Rect);
```

والسبب فى ذلك أن فصيلة نافذة المشهد فى هذا البرنامج موروثة من الفصيلة **CScrollView**. وهذا يُمكن المستخدم من استخدام قضبان الانزلاق وتحريك المشهد بالنافذة إذا كانت بعض أدوات التحكم مختفية. وعملية الانزلاق ، كما نعلم من قبل ، تؤدى إلى اختلاف الإحداثيات المنطقية لمستطيل العينة (المخزن فى المتغير

m_MySample) عن إحداثيات الجهاز المستخدمة فى الدالة **InvalidateRect()**. والدالة الأخيرة تستخدم دائماً إحداثيات الجهاز.

٦. ملف الكود **MyFormView.cpp**: سوف نضيف إلى نفس الملف الكود اللازم للتحكم فى شدة الاستضاءة والذى يمثل مقابض اختيارات القائمة:

- OnColorsBright()
- OnColorsDark()
- OnUpdateColorsBright()
- OnUpdateColorsDark()

(لاحظ فى البرنامج السابق أن هذه العملية كانت تتم عن طريق أزرار الراديو). وهذا هو الكود المطلوب ، موضحاً بالبنط الثقيل.

```
void CMyFormView::OnColorsBright()                    حالة الألوان المضيئة:
    {
// TODO: Add your command handler code here
// New Code:
  m_Brightness = INT_BRIGHT;
  CClientDC ClientDC (this);
  OnPrepareDC (&ClientDC);
  CRect Rect = m_MySample;
  ClientDC.LPtoDP (&Rect);
  InvalidateRect (&Rect);
  UpdateWindow ();
}
//                                                    عند فتح القائمة:
void CMyFormView::OnUpdateColorsBright(CCmdUI* pCmdUI)
{
// TODO: Add your command update UI handler code here
// New Code:
  pCmdUI->SetRadio (m_Brightness == INT_BRIGHT);
}

void CMyFormView::OnColorsDark()                      حالة الألوان الداكنة
{
```

```
// TODO: Add your command handler code here
// New Code:
  m_Brightness = INT_DARK;
  CClientDC ClientDC (this);
  OnPrepareDC (&ClientDC);
  CRect Rect = m_MySample;
  ClientDC.LPtoDP (&Rect);
  InvalidateRect (&Rect);
  UpdateWindow ();
}
//                                                    عند فتح القائمة
void CMyFormView::OnUpdateColorsDark(CCmdUI* pCmdUI)
{
// TODO: Add your command update UI handler code here
// New Code:
  pCmdUI->SetRadio (m_Brightness == INT_DARK);
}
```

والجديد فى هذا الكود هو استخدام الدالة **SetRadio()** التابعة للفصيلة **CCmdUI** (لاحظ أننا قد استخدمنا من قبل الدوال **SetCheck()** والدالة **Enable()** فى الباب التاسع والحادى عشر للتحكم فى الأدوات وصناديق الاختبار). أما الدالة **SetRadio()** فهى تمنح اختيارات القائمة خاصية أزرار الراديو ، بمعنى أنها تمنع تشغيل الاختيارين فى نفس الوقت.

قم بتنفيذ البرنامج بعد هذه التعديلات وشاهد النتائج.

الموجز

١. تعلمنا فى هذا الباب إنشاء البرامج التى تعتمد على صناديق الحوار وكذلك برامج النماذج (Forms). وعرفنا كيفية استخدام أدوات التحكم مثل أزرار الراديو وصناديق الاختبار.

٢. استخدمنا فى هذا الباب مجموعة من الدوال الجديدة عـــلاوة على مهارات جديدة فى استخدام الفصائل والدوال التى مررنـــا بها من قبل ، وهى:

الفصائل:

CRect
Cwnd
CDC
CCmdUI
CBrush

الدوال:

CWnd::IsDlgButtonChecked()
CWnd::InvalidateRect()
CWnd::UpdateWindow()
CWnd::GetDlgItem()
CWnd::GetWindowRect()
CWnd::ScreenToClient()
Crect::InflateRect()
CDC::FillRect()
CDC.LPtoDP()
CCmdUI::SetRadio()

الثوابت والماكرو:

WM_PAINT
RECT
COLORREF
RGB
BN_CLICKED

معالم الطريق

إن تغطية لغة سى++ النوافذية ــ بعد أن وصلت إلى الطراز ٦ ــ تحتاج إلى أكثر من مجلد ضخم ، وذلك لأن شـــركة ميكروســـوفت تضيف إلى البناء عاما بعد عام حتى أصبح شامخاً. هذا علاوة على

أن البرمجة فى حد ذاتها قد تشعبت مجالاتها وفروعها. وقد أصبح من الشائع أن تجد كتاباً يتحدث عن برمجة قواعد البيانات بلغة سى++ وآخر يعرض موضوع برمجة المركبات COM أو ActiveX. وهذا الكتاب ــ بالرغم من محاولتنا أن نجعله وافياً بقدر الإمكان ــ لكنه فى الحقيقة يضع قدمك على الطريق ، ويفتح أمامك المجالات المختلفة للبرمجة. ولعلك فى نهاية هذه الرحلة سوف تتطلع إلى المزيد بالقراءة أو ممارسة البرمجة فى الموضوعات الآتية:

- الطباعة المتقدمة لصفحات الرسم وصناديق الحوار والنماذج.
- المزيد عن البرامج متعددة الوثائق.
- برمجة الخيوط المتعددة (Threads).
- الاتصالات ما بين العمليات (Process Communication)
- استخدام التكنولوجيا OLE
- استخدام مركبات ActiveX

مع أطيب التمنيات ، وإلى اللقاء دائماً.

الملحق (أ)

إجابات أسئلة "اختبر مستواك فى لغة سى++"

١. قيم التعبيرات بالترتيب هى:

```
1
0
0
```

٢. قيم n بالترتيب هى:

```
0
1
1
```

٣. قيم y بالترتيب هى:

```
12
13
14
```

٤. إذا كانت قيمة X محتوية على تعبير مثل 5+2 فإن الماكرو يعطى نتيجة خاطئة بسبب عدم وجود أقواس تحدد أولوية المؤثرات. الحل هو #define SQUARE (X) * (X).

٥. لابد أن يتم حجز مكان فى الذاكرة للمؤشر باستخدام الدالة malloc() أو المؤثر new. الكود الأتى يعمل بنجاح:

```
void main(void)
{
        int *i = new(int)
        *i = 10;
}
```

٦. (int)malloc(100 * sizeof(int))

٧. معنى الإعلانات بالترتيب:

- دالة ترجع مؤشراً إلى عدد صحيح
- مصفوفة تحتوى على خمسة أعداد صحيحة
- مصفوفة تحتوى على خمسة مؤشرات إلى أعداد صحيحة
- مؤشر إلى مصفوفة تختزن خمسة أعداد صحيحة
- مؤشر إلى مؤشر إلى لبنة (char)

٨. يتقدم المؤشر بمقدار ٤ بايت (باعتبار أن المتغير الصحيح يشغل ٤ بايت ــ أما فى بيئة ١٦ بت فيشغل ٢ بايت).

٩. استخدم هذا الكود لضرب المتغير فى ١٥:

```
y <<= 4;
```

(وهو يؤدى إلى إزاحة المتغير y بمقدار ٤ خانات إلى اليسار وتخصيص الناتج لنفس المتغير).

استخدم الكود الآتى لضرب المتغير a فى ١٦ وتخصيص الناتج للمتغير x:

```
x = (a << 4);
x -= a;
```

١٠. معنى الإعلانات بالترتيب:

- مؤشر إلى ثابت.

- مؤشر ثابت إلى متغير
- مؤشر ثابت إلى ثابت.

١١. ثمانية بايت. ٤ لكل عضو من البيانات الأعضاء.

١٢. `point (int p1=10, int const p2=20) : y(p2) { x=p1;}`

الملحق (ب)

أنواع البيانات بالمؤسسة MFC أو بالحزمة Windows SDK

BOOL	قيمة بوليانية
BSTR	مؤشر حرفى (Character Pointer) ــ ٣٢ بت
BYTE	عدد صحيح بدون إشارة ــ ٨ بت
COLORREF	قيمة عددية تستخدم للون ــ ٣٢ بت
DWORD	عدد صحيح بدون إشارة (أو عنوان كلمة (Word) فى الذاكرة ــ ٣٢ بت
LONG	عدد صحيح بإشارة ــ ٣٢ بت
LPARAM	قيمة عددية تمرر كبارامتر إلى روتين النوافذ (أو لدالة من النوع callback) ــ ٣٢ بت
LPCSTR	مؤشر إلى ثابت حرفى ــ ٣٢ بت
LPSTR	مؤشر إلى حرفى ــ ٣٢ بت
LPCTSTR	مؤشر إلى ثابت حرفى يستخدم النظام الموحد "Unicode" ــ ٣٢ بت
LPTSTR	مؤشر إلى حرفى يستخدم النظام الموحد "Unicode" ــ ٣٢ بت
LPVOID	مؤشر إلى نوع خالى من الرصيد (أو غير محدد) - ٣٢ بت
LRESULT	قيمة مرتجعة من روتين النوافذ أو دالة من النوع callback ــ ٣٢ بت
UINT	عدد صحيح ١٦ بت بدون إشارة (تحت نظام النوافذ 3.x) أو عدد صحيح ٣٢ بت بدون إشارة (تحت النظام Win32)

WNDPROC	مؤشر إلى روتين النوافذ ــ ٣٢ بت
WORD	عدد صحيح بدون إشارة ــ ١٦ بت
WPARAM	قيمة تمرر إلى روتين النوافذ أو دالة من النوع call back (١٦ بت تحت نظام النوافذ 3.x أو ٣٢ بت تحت النظام Win32)
POSITION	قيمة تستخدم لتحديد موقع عنصر فى مجموعة _ MFC فقط
LPCRECT	مؤشر ٣٢ بت إلى منشأ المستطيل RECT ــ MFC فقط

الملحق (ج)

المنشآت المستخدمة كأنماط للبيانات بالحزمة SDK

منشأ النقطة (POINT)

```
typedef struct tagPOINT {
   LONG x;
   LONG y;
} POINT;
```

منشأ المستطيل (RECT)

```
typedef struct tagRECT {
   LONG left;
   LONG top;
   LONG right;
   LONG bottom;
} RECT;
```

منشأ سعة المنظور (SIZE)

```
typedef struct tagSIZE {
   int cx;
   int cy;
} SIZE;
```

للاتصال بالمؤلف

اتصل بالناشر ، أو اكتب بريداً إليكترونيا (e-mail) إلى أحد العناوين التالية ، ونحن نرحب بنقدك ومقترحاتك.

Abolrous@netos.com
sabolrous@hotmail.com
sam@seaii.com