MFC
Microsoft Foundation Classes

Paolo Cignoni
cignoni@iei.pi.cnr.it
24/3/98

Introduzione

- Microsoft Foundation Class Library (MFC) è un "application framework" per scrivere applicazioni per Windows.
- Le MFC sono implementate come un insieme di classi C++, molte delle quali rappresentano oggetti comuni come documenti, finestre, dialog box ecc.

MFC

- Le MFC mettono a disposizione varie classi per rappresentare la struttura di una applicazione:
- Concetti fondamentali rappresentati tramite oggetti
  - applicazione
  - finestra
  - documento
  - vista
- Un'applicazione è un'istanza di un oggetto.

CWinApp

- La CWinApp class è la classe base da cui si deriva la classe per rappresentare un applicazione.
- Si definiscono funzioni membro per inizializzare la propria applicazione (ed ogni sua istanza) e per i vari compiti propri dell'applicazione.

CWinApp

- La classe **CWinApp** incapsula, l'inizializzazione, l'esecuzione e la terminazione di un'applicazione in ambiente Windows
- Un'applicazione deve avere uno (ed un solo) oggetto di una classe derivata da **CWinApp**
- Uno dei membri di tale classe è

```
CWnd* m_pMainWnd;
```


CWnd

- Classe di base per la rappresentazione di generiche finestre
- Da questa classe sono derivate sottoclassi per i più comuni tipi di finestre:
  - Frame Window, Dialog, Views, Buttons

CFrameWnd

- Classe Derivata da CWnd
- Fornisce le funzionalità e i membri per la gestione di una frame window (finestra con barra per il titolo, bottoni di chiusura, minimiz., maxim., bordo per il resize ecc.)
- Di solito si deriva una classe da **CFrameWnd** e vi si aggiunge variabili membro per mantenere dati specifici alla propria applicazione

7

Hello World con le MFC

- Due classi in gioco, finestra e applicazione
  - Derivare la classe CHelloWnd da CFrameWnd
  - Derivare la classe CHelloApp da CWinApp
- Definire una nuova funzione di inizializzazione per l'applicazione che crei, mostri e faccia aggiornare la finestra.
- Dichiarare un'istanza di un CHelloApp

8

La minima applicazione

```
#include <afxwin.h>
// Define a window class derived from CFrameWnd
class CHelloWindow : public CFrameWnd {
public:
   CHelloWindow()
   {Create(NULL,"Hello World!",WS_OVERLAPPEDWINDOW,
   rectDefault); }
};

// Define an application class derived from CWinApp
class CHelloApp : public CWinApp{
public:
   virtual BOOL InitInstance();
};

// Construct the CHelloApp's m_pMainWnd data member
BOOL CHelloApp::InitInstance(){
   m_pMainWnd = new CHelloWindow();
   m_pMainWnd->ShowWindow(m_nCmdShow);
   m_pMainWnd->UpdateWindow();
   return TRUE;
}
// HelloApp's constructor initializes and runs the app
CHelloApp HelloApp;
```

9

Hello World 2

- L'esempio precedente era minimo
- Aggiungiamo:
  - La stringa "Hello World" nel centro della finestra
  - Una finestra di dialogo di about

10

Hello World 2

- Ancora una volta deriviamo due classi
  - Una per l'applicazione CHelloApp
  - una per la finestra CHelloWindows
- La scritta nel mezzo della finestra deve essere disegnata dall'applicazione:
  - Gestione del messaggio ON_WM_PAINT

11

Gestione Messaggi

- Filosofia di fondo:
- Il sistema avvisa l'applicazione che è successo qualcosa mandandogli un messaggio
- l'applicazione deve gestire i messaggi
- Le MFC incapsulano il meccanismo di base della gestione dei messaggi di Windows con una serie di *MACRO*

12

MFC (Visual C++) e Messaggi

- Nella dichiarazione della classe

```
//{{AFX_MSG( CHelloWindow )
afx_msg void OnPaint();
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
```

- Nel file dove sono definiti i membri

```
BEGIN_MESSAGE_MAP( CHelloWindow, CFrameWnd )
  //{{AFX_MSG_MAP( CHelloWindow )
  ON_WM_PAINT()
  //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

- Il ClassWizard del Visual C++ sfrutta i commenti per capire dove modificare il codice*!*

13

Device Context

- Un device context è una struttura di Windows che contiene informazioni sulle proprietà e attributi di un device (e.g. schermo, stampante)
- Tutte le operazioni di disegno sono fatte attraverso un device context. I Device context permettono di disegnare in maniera device-independent sullo schermo, sulla stampante o su di un metafile.

14

MFC e Device Context

- Esiste una classe generale CDC e varie sottoclassi tipo CPaintDC o CClientDC
  - L'oggetto **CDC** fornisce funzioni membro per lavorare con un device context come la stampante, lo schermo o il display context associato con la client area di una finestra
  - Tutte le operzioni di disegno devono essere fatte attraverso funzioni membro di un CDC

15

Scrivere in un CDC

- In risposta alla richiesta di ridisegnare la finestra si chiama la OnPaint()

```
void CHelloWindow::OnPaint()
{
  CPaintDC dc(this); // device context for painting
  CRect rect;
  GetClientRect(rect);

  dc.SetTextAlign(TA_BASELINE | TA_CENTER);
  dc.TextOut(rect.right/2,rect.bottom/2,
                              "Hello World");
}
```

16

Hello World 3

- L'esempio precedente era minimo
- Aggiungiamo:
  - Una finestra di dialogo di about
- Entrano in gioco le ***RISORSE***

17

Resource Files

- Le *risorse (resource files)* sono i dati (binari) che un compilatore di risorse o lo sviluppatore aggiungono al file eseguibile di una applicazione
  - Risorse standard descrivono icone, cursori, bitmap, dialog box, fonts, testo di messaggi, ecc.
  - Risorse definite dall'applicazione possono contenere qualsiasi dato richiesto da una specifica applicazione.

18

Risorse e Visual C++

- Il visual c++ mette a disposizione tool per la creazione e l'integrazione di risorse in un'applicazione.
- Oltre alla dialog box aggiungiamo le seguenti risorse:
  - icona
  - menu
  - String Table
  - accellerator

19

Architettura di Applicazioni Windows

- La struttura delle applicazioni *vere* è basata sui concetti di *documento* e *vista*:
- Un'applicazione serve all'utente per creare,modificare e gestire un documento.
- Due tipi principali di interfaccia
  - SDI *Single Document Interface* (notepad, I.E.)
  - MDI *Multiple Document Interface* (Word, Excel)

20

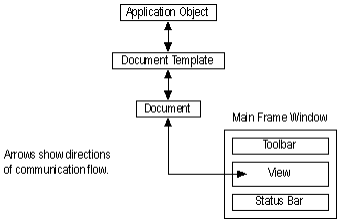
Documenti e viste

- Un documento è un data object con cui l'utente interagisce durante una sessione di editing. E' creato con i comandi New o Open dell'applicazione nel menu File ed è tipicamente salvato in un file.
- Una vista è un window object attraverso cui l'utente interagisce con il documento

21

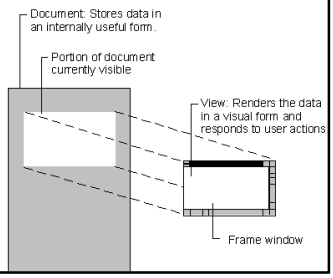
Document Template

- Gli oggetti Document Template coordinano la creazione di documenti viste e frame window quando un nuovo documento e/o vista è creato
- SDI
- MDI



Document Template

- Un oggetto document template definisce le relazioni tra tre classi
  - Documenti
  - Viste
  - Frame Windows



Scribble

- Tutorial delle MFC fornito con il Visual C++
  - MDI application
  - Document e view
  - Application Wizard

24