ใบงานที่ 5 การเขียนโปรแกรม MFC #02

[การเพิ่ม control ในโปรแกรม MFC อย่างง่าย]

วัตถุประสงค์

1. เพื่อให้รู้วิธีการเพิ่ม controls ลงในโปรแกรม MFC อย่างง่าย
2. เพื่อให้มีความคุ้นเคยกับการใช้เครื่องมือต่างๆ ในการสร้าง Application ด้วย MFC Framework

เนื้อหา

เพื่อให้นักศึกษา มีความคุ้นเคยกับการสร้างและใช้งาน control ต่างๆ ในใบงานนี้จะใช้วิธีการเพิ่ม controls ต่างๆ ลงในโปรแกรมแทนการใช้ wizard

ขั้นตอนการสร้างและใช้งาน controls โดยทั่วไป มีดังนี้

1. กำหนด ID ให้กับ control
2. ประกาศชนิดของ control ที่จะสร้างไว้ใน class declaration
3. ขอเนื้อที่ในหน่วยความจำสำหรับ control จากระบบปฏิบัติการ
4. สร้างและลงทะเบียน control กับระบบ Windows
5. ใช้งาน control
6. ลบ control ออกจากระบบและคืนหน่วยความจำให้ระบบปฏิบัติการ

ลำดับการทดลอง

1. ขั้นเตรียมการ โดยการนำเข้าไฟล์จากงานเดิม
   1.1. สร้าง project ใหม่ เป็นชนิด Win32 project แบบ empty project
   1.2. Copy ไฟล์ .cpp และ .h จากใบงานที่ 4 มาไว้ใน folder ของ project นี้
   1.3. เรียกเมนู View -> Solution Explorer
   1.4. คลิกขวา ที่ source file เลือก Add -> Existing Item..
   1.5. เลือกไฟล์ easyMFC01.cpp และ easyMFC01.h กดปุ่ม Add
   1.6. คลิกขวาที่ easyMFC01.cpp เลือก rename ตั้งชื่อใหม่เป็น easyMFC02.cpp และ rename ไฟล์ easyMFC01.h เป็น easyMFC02.h
   1.7. Double click ที่ไฟล์ easyMFC02.cpp เข้าไปแก้ #include “easyMFC01.h” เป็น #include “easyMFC02.h”
   1.8. ทดลอง RUN โปรแกรม จะได้หน้าต่างเหมือนโปรแกรมใน Lab4

2. รายละเอียดเกี่ยวกับ class และวิธีการสร้างปุ่มกด

ชื่อคลาส	CButton
วิธีการจองหน่วยความจำ	CButton *pButton; pButton = new CButton ;
การสร้าง Object	virtual BOOL Create(LPCTSTR lpszCaption, // ข้อความที่ปรากฏบน control DWORD dwStyle, // รูปแบบของ control const RECT& rect, // พื้นที่ที่จะสร้าง control CWnd* pParentWnd, // window แม่ของ control UINT nID // ID ของ control);
dwStyle (ที่ใช้บ่อย)	**BS_PUSHBUTTON** Creates a pushbutton that posts a **WM_COMMAND** message to the owner window when the user selects the button.

หมายเหตุ dwStyle จะต้อง combind โดยการ or (ด้วยเครื่องหมาย |) กับ windows style WS_CHILD และ WS_VISIBLE ด้วยเสมอ

3. การเพิ่มปุ่มกดให้กับ Windows

- เปิดไฟล์ easyMFC02.h

3.1. กำหนด ID ให้กับ control

แทรก #define IDC_MYBUTTON1 1000 ลงในบรรทัดที่ 3 ของไฟล์ easyMFC02.h

3.2. ประกาศชนิดของ control ที่จะสร้างไว้ใน class declaration

แทรกบรรทัด Cbutton *myButton; ลงในบรรทัดส่วน public ของคลาส CmyFrame

คอมไพเลอร์จะรู้จักตัวแปร *myButton ว่าเป็นชนิดปุ่มกดมีขอบเขตอยู่ในคลาส CMyFrame ซึ่งจะมีเฉพาะชื่อแต่ยังไม่มีตัวตนของปุ่มกด

```
// EasyMFc01.h
// application class
#define IDC_MYBUTTON1 1000
class CMyApp : public CWinApp
{
    public:
    virtual BOOL InitInstance();
};

// frame window class
class CMyFrame : public CFrameWnd
{
    public:
    CMyFrame();
    ~CMyFrame();
    CButton *myButton;

};
```

1 define

2 declare

3.3. สร้างปุ่มกดโดยใช้คำสั่ง Create ดังต่อไปนี้

```
CMyFrame::CMyFrame()
{
    myButton = new CButton;
    Create(NULL,"Hello Visual C++, Easy MFC 01",
        WS_OVERLAPPEDWINDOW, CRect(100,100,500,300));
    myButton->Create("Click me",
        WS_CHILD | WS_VISIBLE |BS_PUSHBUTTON,    // styl
        CRect(10,10,100,50),                     // posi
        this,                                    // para
        IDC_MYBUTTON1                            // ID c
        );
}
```

3 memory allocate for control

4 Create control

3.4. เมื่อใช้งานเสร็จ ก่อนจบโปรแกรมจะต้องคืนหน่วยความจำที่ขอไว้กลับคืนสู่ระบบ โดยเพิ่มคำสั่งต่อไปนี้ลงใน destructor

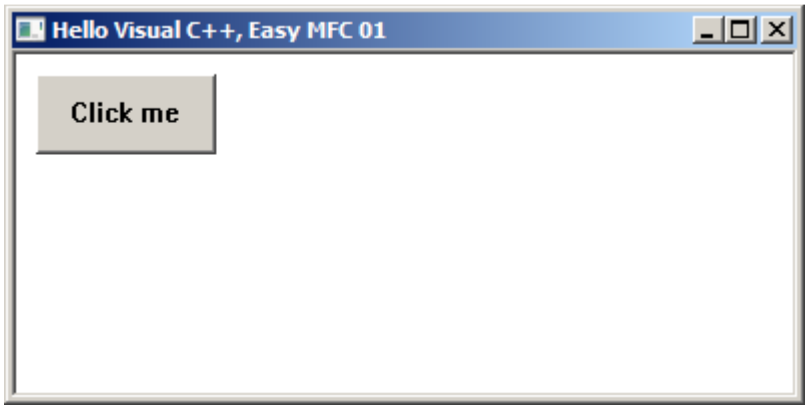
Destructor คือฟังก์ชันที่จะทำงานทุกครั้งที่ object หมดอายุ มีชื่อเดียวกับคลาสแต่มีเครื่องหมาย ~ นำหน้าชื่อฟังก์ชัน ในที่นี้คือ CMyFrame::~CMyFrame()

```
CMyFrame::~CMyFrame()
{
    delete myButton;
}
```

6 destroy control

หมายเหตุ เราจะข้ามขั้นตอนที่ 5 คือ การใช้งาน control ไปก่อน เนื่องจากเป็นเรื่องยาว ซึ่งจะอยู่ในใบงานที่ 7

4. ทดลอง compile และ Run โปรแกรม



Reference

- **ไฟล์ easyMFC02.h**

```
// EasyMFc01.h
// application class
#define IDC_MYBUTTON1 1000
class CMyApp : public CWinApp
{
	public:
	virtual BOOL InitInstance();
};

// frame window class
class CMyFrame : public CFrameWnd
{
	public:
	CMyFrame();
	~CMyFrame();
	CButton *myButton;
};
```

- **ไฟล์ easyMFC02.cpp**

```
// EastMFC01.cpp
#include <afxwin.h> // MFC library header file declares base classes
#include "easyMFC02.h"

CMyApp theApp; // the one and only CMyApp object

BOOL CMyApp::InitInstance()
{
	m_pMainWnd = new CMyFrame();
	m_pMainWnd->ShowWindow(m_nCmdShow);

	m_pMainWnd->UpdateWindow();
	return TRUE;
}
```

```
CMyFrame::CMyFrame()
{
	myButton = new CButton;
	Create(NULL,"Hello Visual C++, Easy MFC 01",
		WS_OVERLAPPEDWINDOW, CRect(100,100,500,300));
	myButton->Create("Click me",
		WS_CHILD | WS_VISIBLE |BS_PUSHBUTTON, // style for button control
		CRect(10,10,100,50), // position and size of control
		this,  // parant windows, this= name of this class
		IDC_MYBUTTON1 // ID of control
		);
}

CMyFrame::~CMyFrame()
{
	delete myButton;
}
```