

공용 컨트롤 II

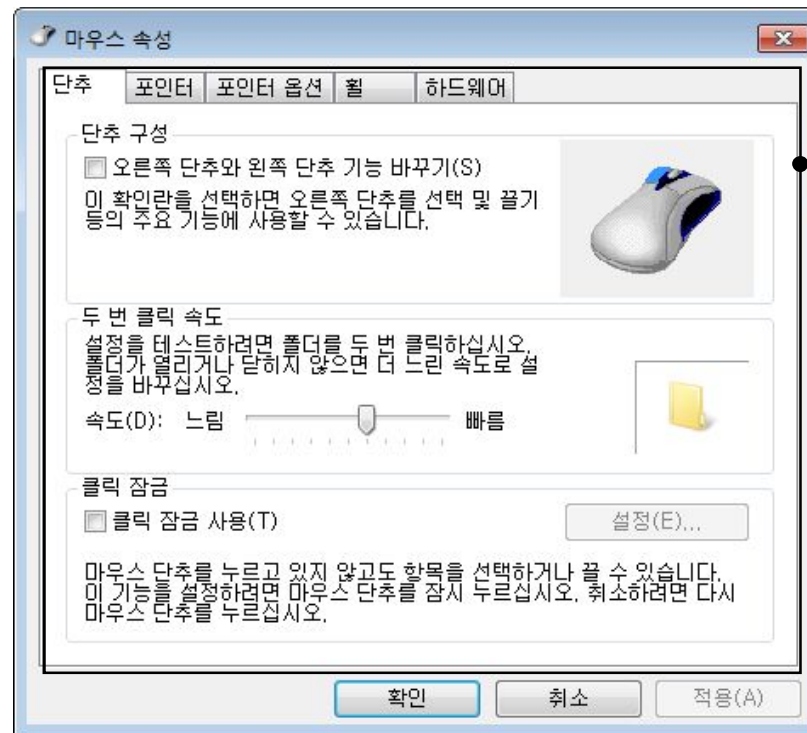
(5주차)

학습개요

- 학습 목표
 - 공용 컨트롤의 종류와 특징을 이해한다.
 - 공용 컨트롤의 프로그래밍 방법을 익힌다.
- 학습 내용
 - 탭 컨트롤
 - 스피ن 버튼 컨트롤
 - 프로그레스 컨트롤
 - 실습

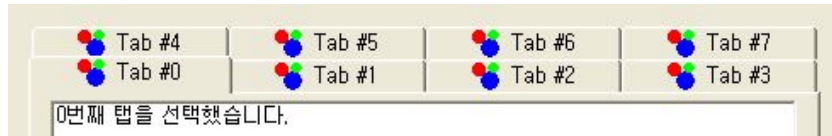
탭 컨트롤

- 동일한 화면 영역에 서로 다른 내용을 표시

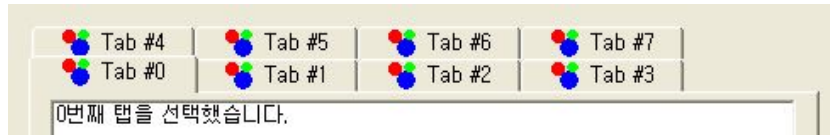


탭 컨트롤

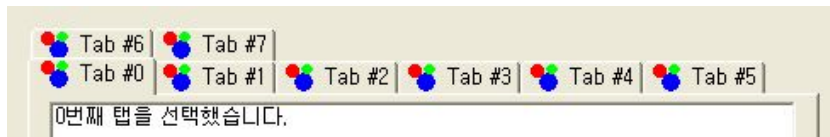
탭 정렬 방식



Right Justify : 각 탭의 폭이 자동 증가하여 한 줄을 가득 채운다.



Fixed Width : 모든 탭의 폭이 가장 긴 레이블에 맞춰 일정 크기가 된다.



Ragged Right : 각 탭의 폭이 자동 증가하지 않으므로 한 줄을 가득 채우지 않는다.

탭 컨트롤 통지 코드

- 상태 변화가 생기면 WM_NOTIFY 통지 메시지를 부모 윈도우에 보냄
- 통지 메시지와 더불어 다음과 같은 통지 코드 전달

통지 코드	의미
TCN_FOCUSCHANGE	버튼 포커스가 변경되었다.
TCN_GETOBJECT	오브젝트를 탭 위에 끌어다 놓았다.
TCN_KEYDOWN	키를 눌렀다.
TCN_SELCHANGE	탭 선택이 변경되었다.
TCN_SELCHANGING	탭 선택이 변경되기 직전이다.

탭 컨트롤 클래스 (1)

- 탭에 사용할 이미지 리스트 설정하기

```
CImageList * CTabCtrl::SetImageList(CImageList * pImageList);
```


- 탭 추가하기

```
LONG CTabCtrl::InsertItem(int nItem, TCITEM* pTabCtrlItem);  
LONG CTabCtrl::InsertItem(int nItem, LPCTSTR lpszItem);  
LONG CTabCtrl::InsertItem(int nItem, LPCTSTR lpszItem, int nImage);
```

- nItem: 탭의 인덱스(0, 1, ...)
- TCITEM: 추가할 탭에 대한 모든 정보를 가진 구조체
- lpszItem: 새로운 탭에 표시할 텍스트(레이블)
- nImage: 탭에 표시할 이미지(이미지 리스트의 인덱스 값)

탭 컨트롤 클래스 (2)

- 탭 추가

사용 예	<pre>CTabCtrl m_tab; // 탭 컨트롤 객체 TCITEM tcltem; tcltem.mask = TCIF_TEXT; tcltem.pszText = _T("Tab #0"); m_tab.InsertItem(0, &tcltem); tcltem.pszText = _T("Tab #1"); m_tab.InsertItem(1, &tcltem); tcltem.pszText = _T("Tab #2"); m_tab.InsertItem(2, &tcltem);</pre>	<pre>m_tab.InsertItem(0, _T("Tab #0")); m_tab.InsertItem(1, _T("Tab #1")); m_tab.InsertItem(2, _T("Tab #2"));</pre>
결과		

탭 컨트롤 클래스 (3)

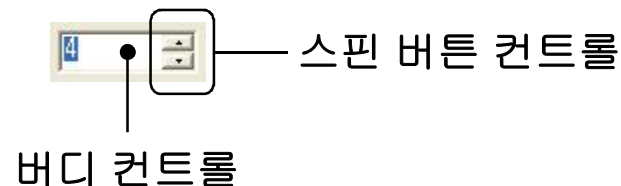
- 탭 선택 변경과 선택된 탭 알아내기

```
int CTabCtrl::SetCurSel(int nItem);  
int CTabCtrl::GetCurSel();
```

- nItem: 선택할 탭의 인덱스(0, 1, ...)

스핀 버튼 컨트롤(=업다운 컨트롤)

- 화살표가 위아래로 붙어 있어서 마우스 클릭으로 값을 증감시킬 수 있음
- 독립적으로 사용되기보다 다른 컨트롤(버디(Buddy) 컨트롤이라 부름)과 함께 사용됨
 - 편집 컨트롤이나 정적 텍스트 컨트롤이 주로 버디 컨트롤로 사용됨



스핀 버튼 컨트롤 클래스 (1)

- 범위 설정과 현재 범위 얻기

```
void CSpinButtonCtrl::SetRange(short nLower, short nUpper);  
void CSpinButtonCtrl::GetRange(int &lower, int &upper);
```

- 위치 설정과 현재 위치 얻기

```
int CSpinButtonCtrl::SetPos(int nPos);  
int CSpinButtonCtrl::GetPos();
```

- 버디 설정하기

```
CWnd* CSpinButtonCtrl::SetBuddy(CWnd* pWndBuddy);
```

스핀 버튼 컨트롤 클래스 (2)

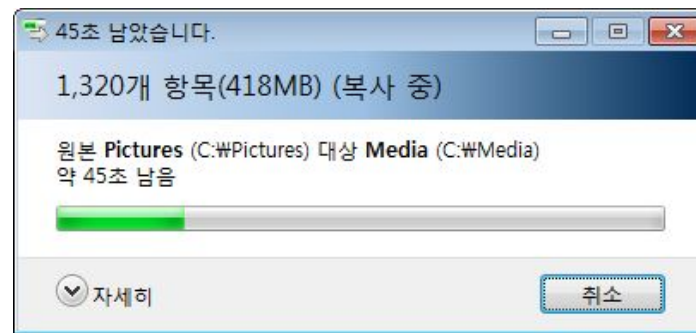
- 숫자의 진수 변경하기

```
int CSpinButtonCtrl::SetBase(int nBase);
```

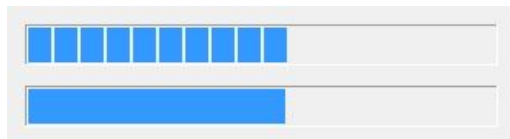
- 스펀 버튼 컨트롤 통지 메시지
 - 마우스로 조작하면 WM_VSCROLL 또는 WM_HSCROLL 메시지 발생 → 메시지와 더불어 현재의 위치값이 부모 윈도우에 전달됨

프로그레스 컨트롤

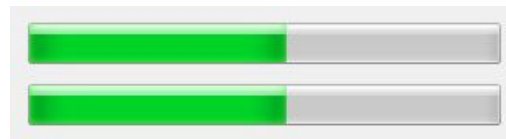
- 시간이 오래 걸리는 작업의 진행을 시각적으로 표시



- 프로그레스 컨트롤 스타일



(a) 클래식 테마



(b) 윈도우 7 테마

프로그레스 컨트롤 클래스

- 범위 설정과 현재 범위 얻기

```
void CProgressCtrl::SetRange(short nLower, short nUpper);  
void CProgressCtrl::GetRange(int& nLower, int& nUpper);
```

- 위치 설정과 현재 위치 얻기

```
int CProgressCtrl::SetPos(int nPos);  
int CProgressCtrl::GetPos();
```

- 현재 위치 증감하기

```
int CProgressCtrl::OffsetPos(int nPos);  
int CProgressCtrl::SetStep(int nStep);  
int CProgressCtrl::StepIt();
```

학습정리

- 탭 컨트롤은 동일한 화면 영역에 서로 다른 내용을 표시하고자 할 때 사용한다.
- 탭 컨트롤은 CTabCtrl 클래스의 객체를 생성해 사용한다.
- 스피너 버튼 컨트롤은 화살표가 위아래로 붙어 있어서 마우스 클릭으로 값을 증감시킬 수 있으며, 독립적으로 사용되기보다 다른 버디 컨트롤과 함께 사용된다.
- 스피너 버튼 컨트롤은 CSpinButtonCtrl 클래스의 객체를 생성해 사용한다.
- 스피너 버튼 컨트롤은 시간이 오래 걸리는 작업의 진행을 시각적으로 표시할 때 사용한다.
- 스피너 버튼 컨트롤은 CProgressCtrl 클래스의 객체를 생성해 사용한다.