Monikery

Programowanie składnikowe
w modelu COM

Jarosław Francik
czerwiec 2002

Do czego służą monikery?

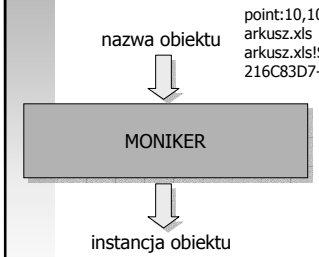
- Zastosowanie – dwa w jednym:
  - inteligentne nazwy obiektów
  - inicjalizowanie obiektów
- Co ma jedno do drugiego?

```
class CPoint
{
    CPoint (int x, int y);
    void Draw();
};

CPoint *p =
     new CPoint(10, 10);
```

```
interface IPoint
{
      HRESULT Draw();
};

IPoint *p = NULL;
CoCreateInstance(clsid,...
   IID_IPoint, (void**)&p);
```

CoCreateInstance nie jest odpowiednikiem konstruktora: nie ma parametrów wywołania

Monikery – istota działania



point:10,10
arkusz.xls
arkusz.xls!Sheet1!R1C1:R10C6
216C83D7-4621-4ffd-A87D-61DD9C3D2A66

Monikery standardowe

plikowy	*CreateFileMoniker*	ścieżka i nazwa pliku
elementowy	*CreateItemMoniker*	obiekt zawarty w innym obiekcie
klasowy	*CreateClassMoniker*	„opakowanie" CLSID: obiekt klasy
złożony	*CreateGenericComposite*	superpozycja kilku monikerów
URL	*CreateURLMoniker*	URL
wskaźnikowy	*CreatePointerMoniker*	identyfikuje obiekt w stanie aktywnym
OBJREF	*CreateObjrefMoniker*	uszeregowany wskaźnik do interfejsu IUnknown
antymoniker	*CreateAntiMoniker*	odwrotność monikera

Interfejs *IMoniker*

- Dziedziczy po *IPersistStream*
- Ciekawsze funkcje składowe:
  - GetDisplayName
    - zwraca nazwę symboliczną
  - ParseDisplayName
    - przekształca nazwę symboliczną w moniker
  - BindToObject
    - dowiązuje obiekt nazywany przez moniker
  - Reduce
    - redukuje moniker do najprostszej postaci
  - ComposeWith
    - łączy monikery tworząc moniker złożony

Stosowanie nazw symbolicznych

- Metoda IMoniker::ParseDisplayName
- Funkcja MkParseDisplayName
  - dla podanego łańcucha tworzy moniker
  - przyjmuje dwa typy nazw:
    - ścieżka do pliku, np:
      c:\moje dokumenty\arkusz.xls
    - standardowa nazwa obiektu:
      ProgID:Nazwa-obiektu

    gdzie:

      ProgID – zarejestrowany prog id monikera
      Nazwa-obiektu – nazwa rozpoznawalna dla monikera
    - przykład dla monikera klasowego:
      clsid:216C83D7-4621-4ffd-A87D-61DD9C3D2A66

Dowiązywanie obiektów

- Metoda IMoniker::BindToObject
- Funkcja BindMoniker (nieco prostsza w użyciu)

- Co robi IMoniker:: BindToObject?
  - tworzy instancję odpowiedniego obiektu
  - inicjalizuje ją
- Jak IMoniker::BindToObject inicjalizuje obiekty?
  - zależy od typu monikera
  - moniker plikowy: wciąga z pliku; tworzony obiekt musi implementować *IPersistFile*
  - moniker klasowy: wywołuje *CoGetClassObject*

Czas na kod…

monikerów raczej nie tworzymy przez CoCreateInstance!

```
// tworzy kontekst wiązania
IBindCtx *pBindCtx;
hr = CreateBindCtx(0, &pBindCtx);

// utworzenie monikera
ULONG chEaten;
IMoniker *pMoniker = NULL;
hr = MkParseDisplayName(pBindCtx, pszNazwa,
                        &chEaten, &pMoniker);

// dowiązuje obiekt
ICoś *p
pMoniker->BindToObject(pBindCtx, 0, IID_ICoś,(void**)&p);

pMoniker->Release();  // monikery nie żyją długo…
pBindCtx->Release();  // konteksty wiązania też nie
```


Kod: uproszczenia

- Dowiązanie obiektu bez jawnego kontekstu:

zamiast:

```
IBindCtx *pBindCtx;
CreateBindCtx(0, &pBindCtx);
pMoniker->BindToObject(pBindCtx, 0, IID_ICoś, (void**)&p);
pBindCtx->Release();
```

wystarczy:

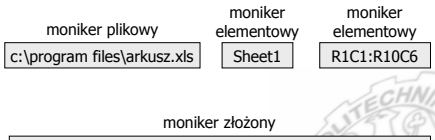
```
BindMoniker(pMoniker, 0, IID_ICoś, (void**)&p);
```

- Funkcja *CoGetObject*

robi wszystko: udostępnia obiekt na podstawie nazwy

```
HRESULT CoGetObject(LPCWSTR pszDisplayName,
   BIND_OPTS *pBindOptions, IID &riid, void **ppv);
```

Monikery złożone



moniker złożony

c:\program files\arkusz.xls!Sheet1! R1C1:R10C6

Przykład: OLE Linking

- Dokument złożony (kontener) zawiera utrwalone dane monikera plikowego dokumentu włączonego (linked)
- Korzystając z interfejsu *IPersistStream* (który implementują wszystkie monikery) można z dokumentu utrwalonego odtworzyć obiekt monikera plikowego
- Moniker plikowy identyfikuje swą nazwę (plik) i dowiązuje zapisany w nim dokument włączony
- Dokument włączony, aby współpracować z monikerem plikowym, implementuje IPersistFile

Podsumowanie

- Monikery tworzymy za pomocą specjalnych funkcji (*CreateXXXMoniker)* lub podając nazwę symboliczną (*MkParseDisplayName*)
- Monikery implementują IPersistStream, zatem możemy je przechowywać w strumieniach i magazynach
- Utworzony lub odtworzony moniker dowiązuje nazywany przez siebie obiekt (i inicjalizuje go)
- Po dowiązaniu obiektu moniker zazwyczaj jest usuwany