

Wstęp do programowania składnikowego

CZĘŚĆ 3: Funkcja *CoCreateInstance*, fabryki klas i serwery COM

Jarosław Francik

Podstawy technologii COM

- HRESULT
 - standardowy typ wartości zwracanych przez funkcje
 - S_OK S_FALSE
 - E_UNEXPECTED E_NOTIMPL
 - E_NOINTERFACE E_OUTOFMEMORY
 - Bity:

0-15	return code	co się stało
16-30	facility code	id części systemu
31	sukces czy porażka	
 - Makro SUCCEEDED
 - Funkcja FormatMessage

Podstawy technologii COM

- Rejestr Windows
 - HKEY_CLASSES_ROOT
 - CLSID
 - {59F1DE0D-8843-4A57-82CE-D6DBA3D28D28}: Wykład COM, podejście 3
 - InProcServer32: c:\windows\system32\Component.dll
 - ProgID: COMLecture.Approach3.1
 - VersionIndependentProgID: COMLecture.Approach3
 - COMLecture.Approach3.1: Wykład COM, podejście 3
 - CLSID: {59F1DE0D-8843-4A57-82CE-D6DBA3D28D28}
 - COMLecture.Approach3: Wykład COM, podejście 3
 - CLSID: {59F1DE0D-8843-4A57-82CE-D6DBA3D28D28}
 - CurVer: COMLecture.Approach3.1



Podstawy technologii COM

- Rejestr Windows (c.d.)
 - Przydatne funkcje:
 - CLSIDFromProgID
 - Rejestrowanie serwerów COM
 - Serwer musi sam dostarczyć eksportowanych funkcji:
 - DllRegisterServer
 - DllUnregisterServer
 - Przydatne narzędzia:
 - RegOpenKeyEx, RegCreateKeyEx, RegSetValueEx, RegEnumKeyEx, RegDeleteKey, RegCloseKey
 - Pliki Registry.h i Registry.cpp można znaleźć wraz z programem przykładowym *approach3*



Podstawy technologii COM

- Przydatne funkcje:
 - CoInitialize(NULL); // dla każdego procesu osobno!
 - CoUninitialize
 - StringFromCLSID
 - StringFromIID
 - CLSIDFromString
 - IIDFromString
 - CoTaskMemAlloc
 - CoTaskMemFree



Podstawy technologii COM

- CoCreateInstance (
 const CLSID &clsid,
 IUnknown *pIUnknownOuter,
 DWORD dwClsContext,
 const IID &iid,
 void **ppv);



Trzecie podejście - klient

- Zastępujemy dotychczasowy CreateInstance wywołaniem CoCreateInstance
- Powiązanie między klientem a serwerem już tylko poprzez CLSID!



Trzecie podejście - klient

```
CoInitialize(NULL);
IA *pIA = NULL;
HRESULT hr = CoCreateInstance(
    CLSID_MyComponent,
    NULL,
    CLSCTX_INPROC_SERVER,
    IID_IA,
    (void**)&pIA);
if (SUCCEEDED(hr))
{
    pIA->fa();
    pIA->Release();
}
```



Fabryka klas

- IClassFactory – interfejs służący do tworzenia egzemplarzy klas (obiektów)
- ```
interface IClassFactory
{
 HRESULT __stdcall CreateInstance
 (IUnknown *pUnknownOuter,
 const IID& iid, void **ppv);
 HRESULT __stdcall LockServer(BOOL b);
}
```
- w wywołaniu CreateInstance brakuje CLSID  
Fabryka klas służy do tworzenia obiektów określonej klasy

---

---

---

---

---

---

---

## Fabryka klas

- Funkcja CoGetObject zwraca wskaźnik na fabrykę klas dla określonego CLSID

```
HRESULT __stdcall CoGetObject(
 const CLSID &clsid,
 DWORD dwClsContext,
 COSERVERINFO *pServerInfo,
 const IID &iid,
 void **ppv);
```

CoGetObject wywołuje DllGetClassObject:  
funkcję obowiązkowo eksportowaną przez serwery COM

---

---

---

---

---

---

---

---

## Fabryka klas

- CoCreateInstance vs CoGetObject

```
HRESULT __stdcall CoCreateInstance (...)
{
 *ppv = NULL;
 IClassFactory *pFactory = NULL;
 HRESULT hr = CoGetObject(clsid, ...,
 IID_IClassFactory, &pFactory);
 if (SUCCEEDED(hr))
 {
 hr = pFactory->CreateInstance(..., iid, ppv);
 pFactory->Release();
 }
 return hr;
}
```

---

---

---

---

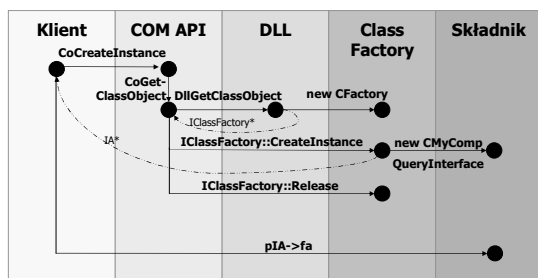
---

---

---

---

## Fabryka klas




---

---

---

---

---

---

---

---



## Trzecie podejście - serwer

- Implementacja klasy składnika:
  - prawie bez zmian
- Implementacja fabryki klas:
  - implementacja IUnknown – standardowa
  - implementacja IClassFactory::CreateInstance
  - implementacja IClassFactory::LockServer
- Implementacja eksportowanych funkcji:
  - DllCanUnloadNow – wymaga zliczania instancji
  - DllGetClassObject
  - DllRegisterServer / DllUnRegisterServer

---

---

---

---

---

---

---



[ listing #3 ]

---

---

---

---

---

---

---



## Trzecie podejście – co dalej?

- Serwer COM rejestrujemy za pomocą programu regsrv32
- Interfejsy mają stałe IID
- Aplikacje wcale nie muszą korzystać stale z tych samych CLSID

---

---

---

---

---

---

---