





Interfejs IDispatch,
dispinterfejsy
i automatyzacja

Jarosław Francik

```

Interface IDispatch : IUnknown
{
    HRESULT GetTypeInfoCount([out] UINT __RPC_FAR *pctinfo);
    HRESULT GetTypeInfo(
        [in] UINT iTInfo, [in] LCID lcid,
        [out] ITypeInfo __RPC_FAR *ppTInfo);
    HRESULT GetIDsOfNames(
        [in] REFIID riid,
        [in] LPOLESTR __RPC_FAR *rgszNames,
        [in] UINT cNames,
        [in] LCID lcid,
        [out, size_is(cNames)] DISPID __RPC_FAR *rgDispId);
    HRESULT Invoke(
        [in] DISPID dispIdMember,
        [in] REFIID riid, [in] LCID lcid, [in] WORD wFlags,
        [in] DISPPARAMS __RPC_FAR *pDispParams,
        [out] VARIANT __RPC_FAR *pVarResult,
        [out] EXCEPINFO __RPC_FAR *pExcepInfo,
        [out] UINT __RPC_FAR *puArgErr);
};

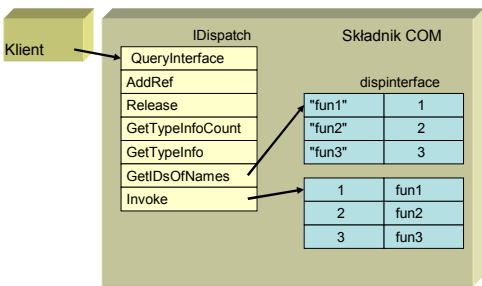
```

interfejs IDispatch (IDL)



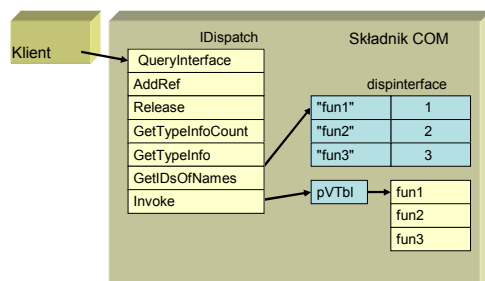


Dispinterface



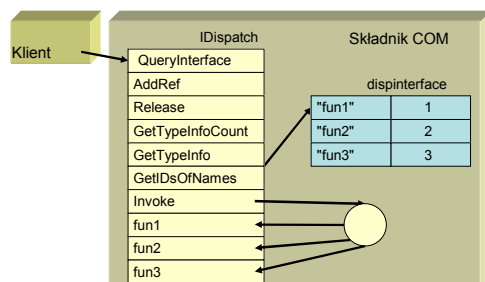


Dispinterface





dual interface





Parametry wywołania Invoke

- dispIdMember – dispid wywoływanej funkcji
- riid – zarezerwowany, ma być IID_NULL
- lcid – użyj GetUserDefaultLCID()
- wFlags – rodzaj funkcji:
DISPATCH_METHOD, DISPATCH_PROPERTYGET,
DISPATCH_PROPERTYPUT, DISPATCH_PROPERTYPUTREF
- pDispParams – tablica parametrów wywołania
tagDISPPARAMS
- pVarResult – wynik funkcji (wskaźnik na wariant)
- pExcepInfo, puArgErr – sytuacje wyjątkowe



DISPPARAMS i Warianty

```
struct tagDISPPARAMS { VARIANTARG *rgvarg; /* .... */ };
struct tagVARIANT { /* .... */
    VARTYPE vt; /* .... */ // stale VT_*
    union {
        LONG lVal; // VT_I4
        BYTE bVal; // VT_UI1
        SHORT iVal; // VT_I2
        /* ... */
        BSTR bStrVal; // VT_BYREF | VT_BSTR
    };
};
```

Ciekawe funkcje:

SysAllocString

VariantChangeType

```
wchar_t progid[] = L"COMLecture.Approach5.1";
CLSID clsid;
CLSIDFromProgID(progid, &clsid);

IDispatch *pIDispatch = NULL;
HRESULT hr = CoCreateInstance(clsid, NULL, CLSCTX_LOCAL_SERVER,
    IID_IDispatch, (void**)&pIDispatch);

DISPID dispid;
OLECHAR *name = L"fa";
hr = pIDispatch->GetIDsOfNames(IID_NULL, &name, 1,
    GetUserDefaultLCID(), &dispid);

VARIANTARG varg;
VariantInit(&varg); varg.vt = VT_EMPTY;

DISPPARAMS param;
param.cArgs = 0; param.rgvarg = &varg;
param.cNamedArgs = 0; param.cArgs = NULL;
hr = pIDispatch->Invoke(dispid, IID_NULL, GetUserDefaultLCID(),
    DISPATCH_METHOD, &param, NULL, NULL, NULL);
```

Klient dla dispinterface

Visual C++ i MFC:

```
IA ia;
ia.CreateDispatch("COMLecture.Approach5.1",)
ia.fa();
```

Visual Basic:

```
Dim x As Object
CreateObject("COMLecture.Approach5.1",)
x.fa
```

Klienci dla dispinterface
