

Wstęp do programowania składnikowego

CZĘŚĆ 1: składniki i interfejsy

Jarosław Francik

Programowanie składnikowe

- Złota zasada:
Interfejsem jest wszystko
- Interfejsy lepiej oddają funkcjonalność obiektów niż dziedziczenie

Założenia:

1. Programowanie składnikowe to **sposób programowania** (filozofia)
2. COM to tylko jedna ze specyfikacji **wspierających** programowanie składnikowe
3. W ramach wykładu korzystamy ze zwykłego C++ by tworzyć oprogramowanie składnikowe
4. W miarę potrzeb będziemy wprowadzać elementy COM

Pierwsze podejście...

- Interfejs = klasa abstrakcyjna
- Składnik = klasa implementująca abstrakcyjny interfejs (dziedzicząca po klasie abstrakcyjnej)

Wiele interfejsów = dziedz. wielobazowe

```
#include <iostream.h>

// interfejs A
class IA
{
public:
    virtual void fa() = 0;
};

// interfejs B
class IB
{
public:
    virtual void fb() = 0;
};

// Składnik
class CMyComp : public IA, public IB
{
// Implementacja interfejsu A
virtual void fa();

// Implementacja interfejsu B
virtual void fb();
};

void CMyComp::fa()
{
    cout << "Jestem fa" << endl;
}

void CMyComp::fb()
{
    cout << "Jestem fb" << endl;
}

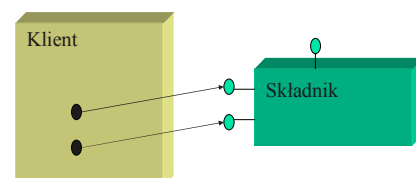
void main()
{
    CMyComp *pComp = new CMyComp;

    // wskaźnik do interfejsu A
    IA *pIA = pComp;
    pIA->fa();

    // wskaźnik do interfejsu B
    IB *pIB = pComp;
    pIB->fb();

    delete pComp;
}
```

Model komunikacji klient-składnik



Pierwsze podejście...

Specyfikacja COM nie wymusza stosowania klas C++,

jednak:

binarny format tablicy funkcji wirtualnych C++ jest zgodny ze specyfikacją COM

Pierwsze podejście...

- Wprowadzamy ułatwienia i wymogi COM:
 - include <objbase.h>
 - interface (słowo kluczowe C++)
 - __stdcall – konwencja wywołania

[listing #1]

Pierwsze podejście...

- Słaby punkt:
 - wykorzystujemy bezpośrednio wskaźnik na klasę komponentu
 - w ten sposób zbyt mocno wiążemy klienta z implementacją składnika
- Rozwiązanie:
 - stworzyć odrębną funkcję *CreateInstance*, której implementacja jest wewnętrznym szczegółem modułu składnika

Lepsze podejście...

- Publicznie widzimy:
 - interfejsy
 - deklarację funkcji *CreateInstance*
 - narzędzie do uzyskiwania wskazanych interfejsów



oczywiście też interfejs:
IUnknown