

# ATLに見る魔術

H.24/09/15

Egtra

第2回閥鍋プログラミング勉強会発表資料



# ATL

- **Active Template Library**
  - Visual C++ 付属
  - COMまわりの面倒を見てくれる
  - テンプレートが沢山

# COM

- **IUnknown**
  - **QueryInterface**
  - **AddRef**
  - **Release**
- **参照カウント・キャスト**

# モジュール

- **DLLやEXE（プロセス）も参照カウントで管理**

# 典型的な実装

```
class Hoge
```

```
: public IUnknown {
```

```
Hoge() { Module Lock; }
```

```
AddRef() { ++count; }
```

## 典型的な実装 (2)

```
Release() {  
    auto old = --count;  
    if (old == 0) {  
        Module Unlock;  
        delete this;  
    }  
}
```

# ATLの基本

```
class Hoge  
  : public CComObjectRoot  
  , ..... (その他インタフェース)  
  
  {  
    // 実装  
  
  };
```

## ATLの基本 (2)

- Hogeクラスではなく、  
CComObjectでオブジェクト  
を作る

```
CComObject<Hoge>* obj;
```

```
CComObject<Hoge>
```

```
    ::CreateInstance(&obj);
```



# ATLの基本 (3)

```
class CComObjectRoot {  
    InternalAddRef() {}  
    InternalRelease() {}  
    long count;  
};
```

# ATLの基本 (4)

```
template<class T>
class CComObject : T {
    virtual ULONG AddRef() override {
        return InternalAddRef();
    }
    // QueryInterface/Releaseも同様
};
```

# ATLの基本 (5)

- **実装(Hoge)は多数のインタフェースから派生**
- **各インタフェースすべて IUnknownから派生**

# ATLの基本 (6)

**Hoge**

- **IStream**
  - **ISequentialStream**
    - **IUnknown**
  - **IDispatch**
    - **IUnknown**

## ATLの基本 (7)

- **継承のツリーにある  
IUnknownすべてをoverride  
するには  
最派生クラスしかない**

# ATLでの参照カウント実装

- **CComObject**
- **CComObjectNoLock**
- **CComObjectGlobal**
- **CComObjectStack(Ex)**

# CComObjectGlobal

- グローバル・静的変数用

```
AddRef() {  
    _Module.Lock();  
}
```

# CComObjectStack

- **自動変数（ローカル変数）**

```
AddRef() {  
    // 何もしない!  
}
```



# 2つの顔を持つオブジェクト

- おさらい: IUnknown
  - virtual QueryInterface();
  - virtual AddRef();
  - virtual Release();

# 2つの顔を持つオブジェクト (2)

- **\_ICPLocator**
  - **virtual \_LocCPQueryInterface();**
  - **virtual AddRef();**
  - **virtual Release();**

# 2つの顔を持つオブジェクト (3)

```
Hoge* hoge;
```

```
IUnknown* obj = hoge;
```

```
IUnknown* obj2 =
```

```
    reinterpret_cast<IUnknown*>(
        (_ICPLocator*)hoge);
```

# 2つの顔を持つオブジェクト (4)

- **obj: interface A, B, C**
- **obj2: interface D, E, F**
- **AddRef/Releaseは共有**

**This work is licensed under a Creative Commons Attribution-ShareAlike 2.1 Japan License.**

