

## Основные классы MFC ( лекция 4 )

### 1. CObject

| Macro used                  | CObject::IsKindOf | CRuntimeClass::CreateObject | CArchive::operator>><br>CArchive::operator<< |
|-----------------------------|-------------------|-----------------------------|--|
| Basic CObject functionality | No                | No                          | No   |
| DECLARE_DYNAMIC             | Yes               | No                          | No   |
| DECLARE_DYNCREATE           | Yes               | Yes                         | No   |
| DECLARE_SERIAL              | Yes               | Yes                         | Yes  |

#### 1.1 DECLARE\_DYNAMIC()

```
class CPerson : public CObject
{
    DECLARE_DYNAMIC( CPerson )
public:
    CPerson() {} ;
};
```

(implementation file \*.CPP).

```
IMPLEMENT_DYNAMIC( CPerson, CObject )

void t()
{
    CObject* pMyObject = new CPerson;
    if(pMyObject->IsKindOf( RUNTIME_CLASS( CPerson ) ) )
        //...
    delete pMyObject;
}
```

#### 1.2 DECLARE\_SERIAL

```
class COtherObject : public CObject
{
public:
    COtherObject() { }
    virtual void Serialize( CArchive& ar ) { }
protected:
```

```
    DECLARE_SERIAL( COtherObject )
};

class CCompoundObject : public CObject
{
public:
    CCompoundObject();
    virtual void Serialize( CArchive& ar );
protected:
    COtherObject* m_pOther;    // Object allocated in constructor
    CObject* m_pObjDyn;    // Dynamically allocated object
    WORD m_ID;
    DECLARE_SERIAL( CCompoundObject )
};

IMPLEMENT_SERIAL(COtherObject,CObject,1)
IMPLEMENT_SERIAL(CCompoundObject,CObject,1) //3d-"version number"

CCompoundObject::CCompoundObject()
{
    m_pOther = new COtherObject; // Exact type known and object already
                                //allocated.

    m_ID = 0;
}

void CCompoundObject::Serialize( CArchive& ar )
{
    CObject::Serialize( ar );    // Always call base class Serialize.
    m_pOther->Serialize( ar );    // Call Serialize on objects of known
                                // exact type.

    // Serialize raw data
    if ( ar.IsStoring() )
        ar << m_ID;
    else
        ar >> m_ID;
}
```

### 1.3 AssertValid()

```
class CPerson : public CObject
{
protected:
```

```
        CString m_strName;
        float   m_salary; // оклад
public:
#ifdef _DEBUG
    virtual void AssertValid() const;    // Override
#endif
    // ...
};

#ifdef _DEBUG
void CPerson::AssertValid() const
{
    // call inherited AssertValid first
    CObject::AssertValid();

    // check CPerson members...
    ASSERT( !m_strName.IsEmpty()); // Must have a name
    ASSERT( m_salary > 0 ); // Must have an income
}
#endif
```