

Môn Lập trình hướng đối tượng

Bài thực hành số 11 : Xây dựng class "serializable" và quan sát việc đọc/ghi đối tượng

Mục tiêu :

- Giúp SV làm quen với việc đặc tả 1 class "serializable" và kỹ thuật viết tác vụ Serialize().
- Giúp SV làm quen với việc viết code đọc/ghi đối tượng.

Nội dung :

- Tạo đặc tả 2 class B và A "serializable" có mối quan hệ tham khảo chéo nhau.
- Hiện thực tác vụ Serialize() a từng class A và B.
- Viết ứng dụng tạo đối tượng B, khởi tạo các thuộc tính cổ điển, các thuộc tính tham khảo đến các đối tượng A, ghi nội dung hiện hành của đối tượng B lên file.
- Viết ứng dụng đọc lại đối tượng B từ file vào bộ nhớ để xử lý tiếp.
- Dịch và chạy chương trình.
- Debug chương trình để tìm lỗi nếu có.
- Chạy ứng dụng ghi đối tượng rồi chạy ứng dụng đọc đối tượng, kiểm tra nội dung đối tượng đọc được có giống với nội dung của đối tượng ghi lên không.

Qui trình :

- Chạy VC++, tạo Project dạng "Win32 Console Application" quản lý ứng dụng gồm 1 file Persistent.cpp với nội dung như sau :

```
//Persistent.cpp : Defines the entry point for the console
application.
//...
CWinApp theApp;
using namespace std;

//khai báo B là class
class B;

//định nghĩa class A "serializable"
class A : public CObject {
public :
    int intA1;
    double dblA2;
    B* pab;
    A();
    virtual void Serialize (CArchive& ar);
    DECLARE_SERIAL(A)
};

//định nghĩa class B "serializable"
class B : public CObject {
public :
    int intB1;
    double dblB2;
    A ba;
    A* pba;
```

```

    A* pba1;
    B();
    virtual void Serialize (CArchive& ar);
    DECLARE_SERIAL(B)
};

IMPLEMENT_SERIAL(A,CObject,1)
IMPLEMENT_SERIAL(B,CObject,1)

B::B() { pba = new A; }
void B::Serialize (CArchive& ar) {
    CObject::Serialize( ar ); //Always call base class Serialize.
    ba.Serialize( ar );
    // Serialize dynamic members and other raw data
    if ( ar.IsStoring() ) {
        ar << pba1; ar << pba;
        // Store other members
        ar << b1; ar << b2;
    } else {
        ar >> pba1; ar >> pba; // Polymorphic reconstruction of
persistent object
        //load other members
        ar >> b1; ar >> b2;
    }
}

A::A() {}
void A::Serialize( CArchive& ar ) {
    CObject::Serialize( ar ); // Always call base class
Serialize.
    if ( ar.IsStoring() ) {
        ar << pab;
        ar << a1; ar << a2;
    } else {
        ar >> pab;
        ar >> a1; ar >> a2;
    }
}

void Create_WriteObject() {
B b;
    //khởi tạo nội dung các thuộc tính cổ điển
    b.b1 = 2; b.b2 = 2.345;
    //khởi tạo nội dung các thuộc tính của đối tượng con ba
    b.ba.a1 = 1; b.ba.a2 = 1.234; b.ba.pab = &b;
    //khởi tạo nội dung các thuộc tính của đối tượng con pba
    b.pba->a1 = 3; b.pba->a2 = 3.1416; b.pba->pab = &b;
    //tạo động đối tượng pba1 và khởi tạo nội dung cho nó
    b.pba = new A;
    b.pba1->a1 = 4; b.pba1->a2 = 4.567;
    b.pba1->pab = &b;
    //tạo file chứa đối tượng cần ghi
}

```

```

    CFile theFile;
    theFile.Open("e:\\persist.bin", CFile::modeCreate |
CFile::modeWrite);
    //tạo đối tượng archive kết hợp với file
    CArchive archive(&theFile, CArchive::store);
    //ghi đối tượng b lên file
    b.Serialize (archive);
    //đóng các đối tượng đã dùng
    archive.Close();
    theFile.Close();
}

//Hàm đọc nội dung đối tượng đã ghi
void ReadObject() {
B b;
    //tạo file chứa đối tượng cần đọc
    CFile theFile;
    theFile.Open("e:\\persist.bin", CFile::modeRead);
    //tạo đối tượng archive kết hợp với file
    CArchive archive(&theFile, CArchive::load);
    //đọc đối tượng b từ file
    b.Serialize (archive);
    //đóng các đối tượng đã dùng
    archive.Close();
    theFile.Close();
}

//chương trình bắt đầu từ hàm _tmain
int _tmain(int argc, TCHAR* argv[], TCHAR* envp[]) {
    int nRetCode = 0;

    // initialize MFC and print and error on failure
    if (!AfxWinInit(::GetModuleHandle(NULL), NULL,
::GetCommandLine(), 0)) {
        // TODO: change error code to suit your needs
        cerr << _T("Fatal Error: MFC initialization failed") <<
endl;
        nRetCode = 1;
    } else {
        // TODO: code your application's behavior here.
        Create_WriteObject(); // hoặc ReadObject()
    }
    return nRetCode;
}

```