Nguyễn Quốc Vương Anh
gdt.anv@gmail.com

I. Mục tiêu

1. Luôn luôn kiểm tra sự hợp lệ của đối tượng ObjectARX .NET
2. OpenMode
3. Giữ đối tượng trong phạm vi nhỏ

II. Luôn luôn kiểm tra sự hợp lệ của đối tượng ObjectARX .NET

- Việc cast đối tượng trong lập trình ObjectARX .NET là rất thường xuyên. Ví dụ ta thực hiện cast như sau thì hợp lệ:

```
BlockTable bt = tr.GetObject(db.BlockTableId, OpenMode.ForRead) as BlockTable;
```

Bởi vì đối tượng db.BlockTableId là ID của BlockTable trong db. Vì vậy, cậu lệnh GetObject sẽ lấy được đối tượng BlockTable này. Tuy nhiên, câu lệnh GetObject chỉ trả về đối tượng DBObject, nên ta cần phải cast qua đối tượng BlockTable.

- Câu lệnh cast sau là không hợp lệ:

```
Ellipse ell = tr.GetObject(db.BlockTableId, OpenMode.ForRead) as Ellipse;
```

Mặc dù không hợp lệ nhưng sẽ không có cảnh báo gì khi chúng ta code. Bởi vì Ellipse cũng là class con của DBObject. Và hoàn toàn "có thể" cast thành công. Sau câu lệnh này thì $ell = null$. Nếu tiếp tục sử dụng đối tượng ell sẽ dẫn đến chương trình bị crash.

- Kết luận: Câu lệnh cast rất thường được sử dụng, tuy nhiên nếu sử dụng với các đối tượng không hợp lệ thì làm cho chương trình hoạt động không như ý muốn.
- Cần luôn luôn check đối tượng sau khi thực hiện cast.

```
BlockTable bt = tr.GetObject(db.BlockTableId, OpenMode.ForRead) as BlockTable;
if (null == ell) {//Do something to handle this case;}
```

III. OpenMode

- Một đối tượng trong ObjectARX .NET sẽ có 2 trạng thái OpenMode.ForRead và OpenMode.ForWrite.
- Giống như tên gọi của chúng. Nếu đối tượng được "open for read" thì ta có thể truy vấn các thuộc tính của nó. Nếu đối tượng được "open for write" thì ta có thể truy vấn và sửa đổi các thuộc tính của nó.
- Một đối tượng đang ForRead có thể dùng hàm UpgradeOpen để thành ForWrite.

- Một đối tượng đang ForWrite có thể dùng hàm DowngradeOpen để thành ForRead.
- Các hàm IsReadEnabled và IsWriteEnabled để kiểm tra trạng thái của đối tượng. Tuy nhiên, các hàm này không được khuyến khích sử dụng. Hoàn toàn có thể loại bỏ được vai trò của các hàm này bằng các thói quen lập trình tốt.

IV. Giữ đối tượng trong phạm vi nhỏ

- Như đã nói ở trên, thói quen lập trình tốt có thể giúp tránh phải sử dụng hàm IsReadEnabled và IsWriteEnabled. Không những vậy, nó còn giúp chúng ta giảm thiểu thời gian debug code, code có tính reliable cao hơn, dễ maintain hơn.
- Thói quen đầu tiên cho ObjectARX .NET là giữ đối tượng ở phạm vi nhỏ. Chúng ta không nên sử dụng một đối tượng trên một phạm vi code dàn trải. Vì như vậy sẽ khó kiểm soát trạng thái của đối tượng, cũng như khó theo dõi được flow của chương trình.
- Việc tách hàm là rất quan trọng khi lập trình. Và tên hàm chính là thứ đại diện cho nhiệm vụ của hàm đó. Đặt tên hàm có tính gợi nhớ/mô tả cao sẽ giúp cho việc code và maintain thuận lợi hơn.