**올바른 코딩 방법**

작성자 : 배성혁 (2010-03-10)

Code Review를 통해, 많은 부분 개발자들이 기본 개념 부재, 무지, 잘못된 습관을 가지고 코딩을 수행하고 있음을 알 수 있었습니다.

이에, 정리가 될지 모르겠지만, 사례 위주로 나열을 하고, 문제점을 개선해 보겠습니다.

다만 순서가 없으므로 생각나는 대로 추가될 예정이므로 Update 날짜를 잘 보기시 바랍니다

1. Value Type 과 Reference Type을 구분해라.

Value Type은 null 값을 가지지 않고, stack에 저장되지만, reference type은 null값을 가지고, heap에 저장된다. reference type의 변수를 사용할 시에는 null인지 먼저 검사해야 한다.

2. 메소드에 전달된 인자가 올바른지 먼저 확인해라.

public static void ForTesting(IsolationLevel isolationLevel, params Action[] actionsToTest)

{

if (actionsToTest == null || actionsToTest.Length == 0)

throw new ArgumentNullException("actionsToTest");

actionsToTest.ForTesting(isolationLevel);

}

메소드 수행에 필요한 인자가 제대로 된 값인지 검사도 하지 않고, 무작정 메소드를 진행하는 것은 자원을 낭비하고, Transaction 등에 위배될 여지를 남기는 방식이다.

3. Int.Parse, Boolean.Parse, DateTime.Parse 함수를 사용하지 마라.

대신 TryParse() 함수 또는 RCL.Core.RwConvert.DefValue() 를 사용해라.

Parse 함수는 Parsing에 실패했을 때, 예외를 발생시킨다. ( 대부분 그 예외에 대한 대처는 하지 않았다.)

4. 되도록 direct casting은 하지 마라

Value type 의 변수의 수형을 변환하고자 한다면, TryParse 함수느 RwConvert.DefValue 를 사용해라.

Reference Type의 변수의 수형을 casting 하고자 한다면 다음과 같이 두가지 방법 중에 하나를 사용해라.

방법 1 : is 로 먼저 검사한 후 direct casting을 수행한다.

if (attr is InheritanceAttribute)

return (InheritanceAttribute)attr;

방법 2 : as 로 indirect casting 후 null 검사를 수행해라.

var result = attr as InheritanceAttribute;

if(result!=null)

return result;

5. NOT 연산자 사용을 자제해라.

if (!\_reader.IsDBNull(index))

result = GetInt32(index);

보다는

if (\_reader.IsDBNull(index) == false)

result = GetInt32(index);

가 읽기 쉽다.

6. If 절 안에 복잡한 로직이 담긴 코드를 넣지 말고, 그 로직을 표현할 수 있는 변수를 선언해서 사용해라.

if(ds!=null && ds.Tables.Count > 0 && ds.Tables[0].Rows.Count > 0)

{

// some codes....

}

위의 코드 보다는

bool dataSetIsNotEmpty = (ds!=null && ds.Tables.Count > 0 && ds.Tables[0].Rows.Count > 0);

if( dataSetIsNotEmpty )

{

// some codes...

}