39. Софтуерни контракти. Пред-условия, пост-условия и инварианти към методи на клас. Синтаксис в .NET. Примери.

Във .NET Framework 4, Code Contracts е framework създаващ по-подходящ синтаксис, за да се изразят класове и функции с множество изходни точки. Code Contracts поддържа 3 типа contracts:

1.preconditions,

2.Postconditions,

3.invariants

Preconditions

се занимават с предусловия, които следва да бъдат проверени за да може метод да се изпълни.

Postconditions

се занимават с условия, които следва да бъдат проверени в момента, когато метод е завършил изпълнението си – коректно или с хвърлено изключение.

invariants

описват условия, които следва да са винаги true за времето на която и да е инстанция на класа. Казано по друг начин, invariant указват условие, което следва да се поддържа през времето на всяко взаимодействие между класа и негов клиент — което значи при всяко изпълнение на public members, включително и на constructors.

Code Contracts API се състои от static methods на клас Contract. методът Requires() се ползва за preconditions , а Ensures() за postconditions.

using System.Diagnostics.Contracts;

public class Calculator

{

public Int32 Sum(Int32 x, Int32 y)

{

Contract.Requires<ArgumentOutOfRangeException>(x >= 0 && y >= 0);

Contract.Ensures(Contract.Result<Int32>() >= 0);

if (x == y)

return 2 \* x;

return x + y;

}

public Int32 Divide(Int32 x, Int32 y)

{ Contract.Requires<ArgumentOutOfRangeException>(x >= 0 && y >= 0);

Contract.Requires<ArgumentOutOfRangeException>(y > 0);

Contract.Ensures(Contract.Result<Int32>() >= 0);

return x / y;

}

}

Въведен е т. нар . Code Contracts rewriter, който пробразува кода на етап компилация

след анализ на целта на preconditions или postconditions. Той разширява автоматично

кода и поставя ново-генерираните блокове там, където им е мястото.

Това означава, че разработчикът не се грижи къде да постави postcondition и

дали ги е дублирал някъде в кода (особено при добавяне на нова exit point ).

Синтаксис:

При preconditions изразът съдържа input parameters и възможно е и друг method или property от същия клас. Към метода следва да се добави и атрибут ‘Pure’ за да се отбележи, че няма да се променят данни. Properties (getters) се подразбира че са pure.

При postconditions обикновено се реферира и друга информация, като например връщана стойност, или начална стойност на local variable. Това става с конструкции :

Contract.Result<T> ....... - за да провери стойност (от тип T) връщана от метод и Contract.OldValue<T> ..... - за да вземе стойност (съхранявана в специална променлива) от началото на изпълнението на метода. Има възможност да се провери и условието в момент на генериране на exception (ако това стане) по време на изпълнение на метод.

Това става с Contract.EnsuresOnThrow<TException> ....