NUnit, un framework de test unitaire

Alexandre MIRABEL et Julien PERREIRA



Sommaire

I.	Ρ	résentation de NUnit	3
1	ι.	Introduction	3
2	2.	Utiliser NUnit dans Visual Studio 2013	3
3	3.	Les annotations	5
II.	Е	xemples d'utilisation de NUnit	6
1	ι.	Ouverture du projet et lancement des tests	6
2	2.	Utilisation des annotations [Test] et [TestFixture] sur le projet « Operator »	9
3	3.	Le projet « Operator » avec des tests fonctionnels	9
4	1 .	Utilisation de l'annotation [ExpectedException] sur le projet « Operator »	9
III.		Glossaire	9
1	ι.	Framework	9
2	2.	Annotations1	0
IV.		Sources1	0

I. Présentation de NUnit

1. Introduction

Un test unitaire est une méthode qui permet de tester un morceau de code d'un programme. Les données associées à un contrôle, les procédures d'utilisation et les procédures d'exploitation peuvent être testées. Ces tests ont pour but de savoir si une application est prête à l'emploi.

NUnit est un framework de test unitaire. Il peut-être téléchargé à l'adresse suivante : <u>http://nunit.org/index.php?p=download</u>

2. Utiliser NUnit dans Visual Studio 2013

Pour utiliser NUnit, il faut utiliser une version de Visual Studio. Nous allons utiliser la version Express de 2013 de Visual Studio :

http://www.microsoft.com/fr-fr/download/details.aspx?id=40787

Ensuite, il faut installer un composant additionnel à Visual Studio 2013. Il faut cliquer sur le menu déroulant « Outils » et aller sur « Gestionnaire de package NuGet ». Ensuite, i lfaut lciquer sur « Console du Gestionnaire de package », comme ci-dessous :



La console du gestionnaire de package s'ouvre et il faut écrire la commande « Install-Package NUnitTestAdapter » pour installer le composant NUnit pour Visual Studio 2013 :



3

Le composant s'installe et il doit y avoir dans la console du gestionnaire de package les messages suivants :



Enfin, dans la solution de Visual Studio 2013, il faut ajouter une référence au framework NUnit. Dans un premier temps, il faut faire un clic droit sur « Références » et cliquer sur « Ajouter une référence... »



Il faut ajouter la dll « nunit.framework.dll » qui se trouve dans le dossier « C:\Program Files (x86)\NUnit 2.6.3\bin\framework » comme ci-dessous :



Ensuite, il faut cliquer sur le bouton « OK » et la référence est ajoutée à la solution comme ci-dessous :



Dans l'explorateur de solution, on peut vérifier que la dll a bien été ajouté comme cidessous :



3. Les annotations

NUnit utilise les annotations pour savoir quelles méthodes sont des méthodes de test. Elles se notent au dessus de la définition d'une méthode. Les principales annotations utilisées sont les suivantes :

- [TestFixture]: Se place avant une classe et permet de savoir quelles classes sont des classes de test.
- > [Before] : La méthode annotée sera exécutée avant la méthode annotée [Test].
- [Test] : Se place avant une méthode de test.

- > [After] : La méthode annotée sera exécutée après la méthode annotée [Test].
- [ExpectedException(typeof(ArgumentException), ExpectedMessage="expected message")] : Le méthode doit renvoyé l'exception choisit.
 « typeof(ArgumentException) » est le type d'exception que l'on attend et « ExpectedMessage » est le message que l'on attend.

II. Exemples d'utilisation de NUnit

Dans cette partie, nous allons voir comment utiliser les annotations et créer des tests basiques à travers deux exemples (Sources et machine virtuelle fournit avec le tutoriel).

1. Ouverture du projet et lancement des tests

Pour les tests, nous avons utilisé Visual Studio 2013 Express. Pour ouvrir un projet, il faut aller dans « Fichier » et cliquer sur « Ouvrir un projet... » :

	Page de démarrage - Microsoft Visual	Studio Express 2013	pour Windows Desktop
FIC	HIER MODIFIER AFFICHER DÉB	oguer équipe	OUTILS TEST FEN
13	Nouveau projet	Ctrl+Ma	j+N -
î.	Nouveau projet d'équipe		
õ	Nouveau fichier	Ctrl+N	
脅	Ouvrir un projet	Ctrl+Ma	j+O
2	Ouvrir un fichier	Ctrl+O	ou unir lo
	Ouvrir depuis le contrôle de code source	:e	puvni ie
ta	Se connecter au projet d'équipe		veautés
	Fermer		
×	Fermer la solution		🛛 🖇 pour V
Ш	Enregistrer les éléments sélectionnés	Ctrl+S	top
	Enregistrer les éléments sélectionnés so	ous	riop
1	Enregistrer tout	Ctrl+Ma	^{j+S} ouverez des i
	Exporter le modèle		les fonctionna
	Contrôle de code source		 ess 2013 pou
₽	Mise en page		sultant les sec
	Imprimer	Ctrl+P	plus sur les nouv
	Account Settings		013 pour Window
	Projets et solutions récents		 r les nouveautés d
×	Quitter	Alt+F4	

Ensuite, il faut aller dans le dossier du projet et cliquer sur le fichier « Operator » comme cidessous :

Ouvrir un projet							
🔾 🗢 📕 « Mes doct	✓→ Rechercher dans	: Operator	٩				
Organiser 🔻 Nouveau	u dossier			- 1	2		
Microsoft Visual St	Nom	Modifié le	Туре	Taille			
Projects	퉬 Operator	17/06/2014 09:19	Dossier de fichiers				
	🌗 packages	17/06/2014 09:19	Dossier de fichiers				
🔆 Favoris	😕 Operator	02/06/2014 14:45	Microsoft Visual S	1 Ko			
🧮 Bureau 😑							
📃 Emplacements ré							
〕 Téléchargements							
🍃 Bibliothèques							
Documents							
🔛 Images							
👌 Musique							
📑 Vidéos							
· · ·							
Nom	du fichier : Operator		✓ Tous les fichiers p	orojet (*.sln;*.d	-		
	- F						
			Ouvrir	Annuler			

Lorsque le projet est ouvert, il doit apparaitre dans l'explorateur de solution :



Ensuite, pour démarrer les tests, il faut cliquer sur le bouton « Démarrer » et être en mode « Debug » et en « x86 » :

<u>G</u> ÉNÉRER	<u>D</u> ÉBOGUER	ÉQUIPE	<u>o</u> utils	TE <u>S</u> T
🕨 Démarrer 🕇	Debug 🝷	x86	-	۽ 🎜
D.f				

La fenêtre suivante s'affiche, et il faut cliquer sur « File » et sur « Open Project... » pour ouvrir le projet que nous voulons tester :

NUnit	
ile View 1	Tests Tools Help
New Proj	ject Ctrl+N
Open Pro	oject Ctrl+O
Close	
Save	Ctrl+S
Save As	
Reload P	roject Ctrl+L
Reload T	ests Ctrl+R
Select R	luntime
Recent P	rojects
Exit	
Status	

Il faut aller dans le dossier du projet puis dans le dossier « bin » et enfin dans le dossier « Debug » et choisir l'exécutable du projet (dans notre cas « Operator ») et cliquer sur « Ouvrir » :

File View Tests Tools Help Tro Open Project	TU NUnit				-	
Nom Organiser Nom Modifié le Type Images Musique Vidéos Vidéos Images Musique Vidéos Vidéos Images Musique Vidéos Vidéos Vidéos Images Musique Vidéos Images Im	File View Tests Tools He	elp				
Organiser Nom Modifié le Type Bureau Nom Implacements re Implacements re Images Operator.vshost Documents Images Musique Vidéos Vidéos Images Musique Vidéos Vidéos Vidéos Images Musique Vidéos Vidéos Images Ordinateur Nom du fichier : Operator Projects & Assemblies(*.nunit,* • Ouvrir Annuler	NU Open Project				×	
Organiser • Nouveau dossier Favoris Nom Modifié le Type Bureau Images Musique Vidéos Images Musique Vidéos Images Nom du fichier : Operator Projects & Assemblies(*.nunit,* •	COC Coperator > bin > Debug • + Rechercher dans : Debug				ug 🔎	
Nom Bureau Emplacements ré Téléchargements Operator Operator.vshost 02/06/2014 14:45 Applicatic Documents Documents Images Musique Vidéos Vidéos Nom du fichier : Operator Projects & Assemblies(*.nunit,* ~ Ouvrir Annuler	Organiser 🔻 Nouveau	dossier				
Bureau Emplacements re Téléchargements Documents Documents Images Musique Vidéos Vidéos Ordinateur Nom du fichier : Operator Projects & Assemblies(*.nunit,* *	🗙 Favoris	Nom		Modifié le	Туре	
 Emplacements re Téléchargements Operator Operato	🤜 Bureau	🚳 nunit.framework.dll		10/10/2013 17:01	Extension	*
Image: Image: Image: Image: <t< th=""><th>🗐 Emplacements ré</th><th>Operator</th><th></th><th>17/06/2014 09:34</th><th>Applicatio</th><th></th></t<>	🗐 Emplacements ré	Operator		17/06/2014 09:34	Applicatio	
Bibliothèques Documents Images Musique Vidéos Rordinateur Nom du fichier : Operator Projects & Assemblies(*.nunit,* ▼ Ouvrir Annuler	\rm Téléchargements	Operator.vshost		02/06/2014 14:45	Applicatio	
Nom du fichier : Operator	 Bibliothèques Documents Images Musique Vidéos Groupe résidentiel 					T T
	n Ordinateur 💽 Nom d	 In fichier : Operator 	•	Projects & Assemblies(* Ouvrir	*.nunit,* 👻	

Après ouverture du projet, on peut voir tous les tests qui vont être lancé. Pour lancer les tests, il faut cliquer sur le bouton « Run » :

to Operator.exe - NUnit					
<u>File View Project Tests Tools H</u> elp					
Store CAUsers ALEX Documents Visual CAUsers ALEX Documents Visual Calculator Calculator Calculator Calculator Addition Test Division Test Multiplication Test Reciprocal Test Square Root Test Subtraction Test	Run Stop				
	۰ III ۲				
	: 🕮 🔍 🖻				
	· · · · · · · · · · · · · · · · · · ·				
	Errors and Failures Tests Not Run Text Output				
Ready	Test Cases : 6				

A la fin des tests, la fenêtre change comme ci-dessous :

Nu	Operator.exe - NUnit		
<u>E</u> ile	e <u>V</u> iew <u>P</u> roject <u>T</u> ests T <u>o</u> ols <u>H</u> elp	4	
Categories Tests	C:\Users\ALEX\Documents\Visual	Run Stop C:\Users\ALEX\Documents\Visual Studio 2013\Projects\Operator\ Passed: 3 Failed: 3 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Tim_ Calculator.Operator+TestClass.DivisionTest:	
		Sub vas: Infinity Calculator.Operator+TestClass.ReciprocalTest: Expected: Expected: 0 But was: Infinity Calculator.Operator+TestClass.SquareRootTest: Expected: Expected: 2 Rut was: NaM Image: Image: Image: Image: Image: Image:	
		à Calculator.Operator.TestClass.DivisionTest() dans c:\Users\ALEX Alex Operator.Operator\Operator \Operator \Operato	
		Errors and Failures lests Not Run lext Output	
Com	pietea	Test Cases : 6 Tests Run : 6 Errors : 0 Failures : 3 Time : 0,443185	11.

lci, on voit les tests qui ont réussit et ceux qui ont échoué. De plus, on peut savoir pourquoi un test n'est pas passé.

2. Utilisation des annotations [Test] et [TestFixture] sur le projet « Operator »

Dans le projet « Operator » du dossier « Operator », nous avons une classe qui implémente les méthodes des principales opérations faites sur les nombres. Dans cette classe, nous avons une classe qui test ces méthodes. Les tests ne fonctionnent pas tous car des cas n'ont pas été gérés.

3. Le projet « Operator » avec des tests fonctionnels

Le dossier « Operator2 » contient le même projet mais cette fois-ci les tests fonctionnent. Les tests fonctionnent car lorsque les valeurs en entrée sont incorrectes, le code retourne une valeur aberrante.

4. Utilisation de l'annotation [ExpectedException] sur le projet « Operator »

Le dossier « Operator3 » contient le même projet avec des tests fonctionnels et un code plus propre qui lève des exceptions au lieu de retourner des valeurs aberrantes.

III. Glossaire

1. Framework

Un framework est un ensemble cohérent de composants logiciels qui permette de créer les fondations et les grandes lignes de tout ou d'une partie d'un logiciel. Ils sont conçus pour modeler l'architecture des logiciels applicatifs, des applications web...

2. Annotations

Une annotation est une façon d'ajouter des méta-données à un code source. Elles sont utilisées dans les tests de logiciel pour dire au compilateur quelles méthodes et classes sont utilisés pour le test.

IV. Sources

https://www.nuget.org/packages/NUnitTestAdapter/

http://docs.nuget.org/docs/start-here/using-the-package-manager-console

http://www.nunit.org/index.php?p=exception&r=2.6.3

http://fr.wikipedia.org/wiki/Framework