

Module 05 – Tests Unitaires en C++

TEST



Objectifs

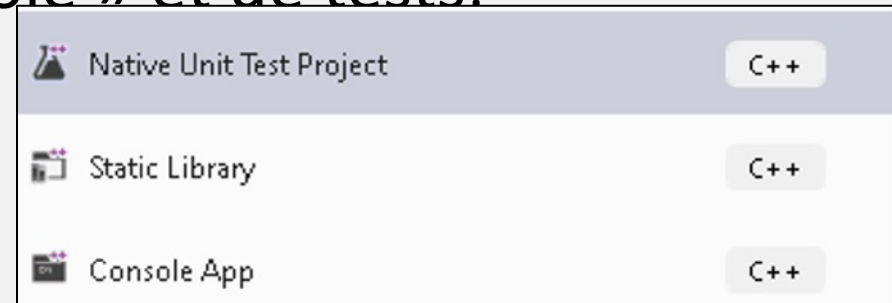
- Mettre en place des tests unitaires avec le cadre de Microsoft
- Nomenclature demandée dans le cours

But des tests unitaires

- Tester des unités : ici des fonctions ou des méthodes
- Les tests doivent répondre à l'acronyme FIRST:
 - Fast : doit être rapide
 - Independent : doit être autonome
 - Repeatable : doit avoir le même résultat d'une exécution à l'autre
 - Self-Verifying : ne doit pas nécessiter d'intervention humaine pour savoir si le test est réussi ou non
 - Torough : doivent être rigoureux, c'est-à-dire couvrir tous les cas idéaux, les cas limites, les erreurs de paramètres, de dépassement de valeurs et les problèmes potentiels de sécurité

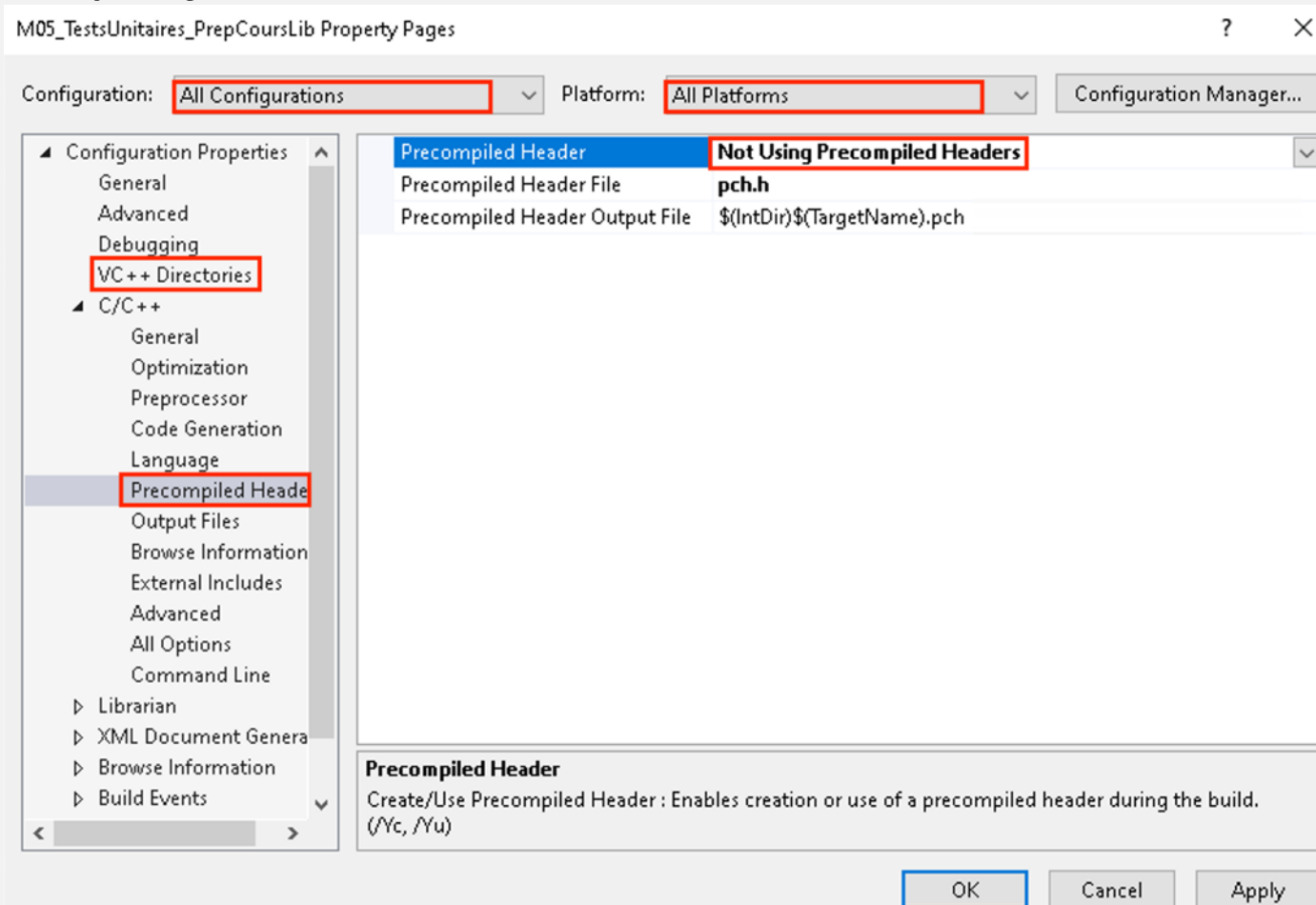
Création des tests unitaires

- Créez un projet de tests qui porte le nom du projet à tester suivi de « Tests »
 - Le gabarit à utiliser dans le cours est « Native Unit Test Project »
 - Exemple : « M05_TestsUnitaires_PrepCoursLibTest » pour tester des fonctions / classes du projet « M05_TestsUnitaires_PrepCoursLib »
- Les unités à tester doit être déclarées et définies dans un projet de type « Static Library ». Dans notre exemple « M05_TestsUnitaires_PrepCoursLib »
- Pour utiliser ce projet dans votre projet « Console » et de tests. n'oubliez pas de l'ajouter comme référence



Désactiver les entêtes précompilées (pch)

- Propriétés du projet et :



Nomenclature

- Il faut un projet de tests par projet à tester
- Il faut un fichier de tests par fichiers CPP à tester
- Le nom du projet de tests doit être composé du nom du projet à tester suivi de « Tests »
- Le nom de chaque fichier de tests doit être le nom du fichier CPP à tester suivi de « Test »
- Méthodes de tests :
 - Un méthode de test par cas de test
 - Le nom doit être composé de trois parties :
 - Nom de la fonction/méthode à tester
 - Description du cas de test
 - Comportement attendu
 - Le test doit être rédigé en suivant la structure des 3A : Arranger, Agir, Auditer

Exemple 1

```
1 #include <CppUnitTest.h>
2
3 #include "../M05_TestsUnitaires_PrepCoursLib/mesmath.h"
4
5 using namespace Microsoft::VisualStudio::CppUnitTestFramework;
6
7 namespace M05TestsUnitairesPrepCoursLibTest
8 {
9     TEST_CLASS(mesmathTest)
10    {
11    public:
12
13        TEST_METHOD(calculerMinimum_UnElement_LElement)
14        {
15            // Arrangeur
16            constexpr int nombreElements = 1;
17            int valeurs[nombreElements] = { 1 };
18            int valeurAttendue = 1;
19
20            // Agir
21            int valeurCalculee = calculerMinimum(valeurs, nombreElements);
22
23            // Auditer
24            Assert::AreEqual(valeurAttendue, valeurCalculee);
25        }
26    };
27 }
```

100 % No issues found Ln: 24 Ch: 53 Col: 62 TABS CRLF

Test Explorer

Test run finished: 1 Tests (0 Passed, 1 Failed, 0 Skipped) run in 188 ms 0 Warnings 0 Errors

Test	Duration	Trait
✗ M05_TestsUnitaires_PrepCoursLibTest (1)	114 ms	
✗ M05TestsUnitairesPrepCoursLibTest (1)	114 ms	
✗ mesmathTest (1)	114 ms	
✗ calculerMinimum_UnElement_LElement	114 ms	

✗ calculerMinimum_UnElement_LElement

Source: mesmathTest.cpp line 13

Duration: 114 ms

Message:

Assert failed. Expected:<1> Actual:<42>

Stack Trace:

mesmathTest::calculerMinimum_UnElement_LElement() line 25

Exemple 2

The screenshot displays a C++ test method in a code editor and its execution results in the Test Explorer.

Code Editor:

```
27 TEST_METHOD(calculerMinimum_Null_Exception)
28 {
29     // Arrange
30     int* valeurs = nullptr;
31     int nombreElements = INT_MAX;
32
33     // Agir & Auditer
34     Assert::ExpectException<std::invalid_argument>([&]() { calculerMinimum(valeurs, nombreElements); });
35 }
```

Test Explorer:

Test run finished: 2 Tests (1 Passed, 1 Failed, 0 Skipped) run in 155 ms

Test	Duration	Trait
✗ M05_TestsUnitaires_PrepCoursLibTest (2)	86 ms	
✗ M05TestsUnitairesPrepCoursLibTest (2)	86 ms	
✗ mesmathTest (2)	86 ms	
✓ calculerMinimum_Null_Exception	< 1 ms	
✗ calculerMinimum_UnElement_LElement	86 ms	

Test Detail Summary:

- ✓ calculerMinimum_Null_Exception
- Source: [mesmathTest.cpp](#) line 27
- Duration: < 1 ms

Références

- GitHub du cours
- <https://learn.microsoft.com/en-us/visualstudio/test/writing-unit-tests-for-c-cpp>
- <https://learn.microsoft.com/en-us/visualstudio/test/how-to-use-microsoft-test-framework-for-cpp>