Laboratoire: Tests unitaires

# Objectif(s)

* Créer des tests unitaires:
  + Méthodes
  + Exception
  + Déterminer des valeurs de tests

# Créer la branche de la séance

## Création du Repository et de la branche de la séance

1. Dans Gitkraken, clonez votre repository GitHub **A21\_3W6\_Labos\_NOM\_PRENOM**
2. Cliquez sur la branche de base: **MAIN**
3. Créez la branche de la séance: **S06\_TestsUnitaires**
4. Copiez la solution de départ dans votre branche

# Créer un projet de tests NUNIT

## Créer le projet pour Les Tests NUNIT

1. Ajoutez à la solution de départ un projet de type Tests NUnit Net 5.0 Core **App\_TestsUnitaires\_TestsNU**
2. À partir du projet **App\_TestsUnitaires\_TestsNU,** ajoutez une référence au projet: **Appl\_TestsUnitaires**

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary** INIT Création projet Tests NUnit
   * **Description**: Création du projet de tests NUnit
5. Cliquez sur **Commit**
6. "Poussez" **Push** les modifications sur le repo en ligne **Remote**: cliquez sur 

# Tester FizzBuzz 1 méthode par scénario

## Identifiez les scénarios à tester

1. Examinez la méthode à tester: **GetOutput**
2. Identifiez les scénarios à tester pour couvrir toutes les classes d'équivalence et des DONNÉES (valeurs) de tests:
   * Divisible par 3 et 5: 15
   * Divisible par 3: 3
   * Divisible par 5: 5
   * Non divisible par 3 ou 5 : 7

## Tester avec divisible par 3 et 5

1. Renommez la classe de tests générée par défaut **FizzBuzz\_Tests**
2. Supprimez la méthode Setup (pas utile pour l'instant)

Suggestion: Affichez la classe **E1\_FizzBuzz** à côté de la classe de tests (clic droit sur une des classe, Nouveau groupe de documents vertical)

1. Renommez la méthode Test1: GetOutput\_Duv3And5\_ReturnFizzBuzz
2. Placez en commentaires les trois sections A:

public void GetOutput\_Div3And5\_ReturnFizzBuzz()

{

// Arrange

// Act

// Assert

Assert.Pass();

}

1. Dans Arrange, générez une variable Valeur = 15
2. Dans Act, exécutez la méthode de FizzBuzz GetOutput en passant Valeur en paramètre : Vous devez ajouter une référence de classe au namespace Appl\_TestsUnitaires (Using)
3. Créez une variable résultat pour mettre le résultat final du Act: ici une seule étape donc on peut faire 1 seule ligne

var Result = E1\_FizzBuzz.GetOutput(Valeur);

1. Dans Assert, remplacez **Assert.Pass()** (sinon le test réussi tout le temps) par la comparaison du Result et du résultat attendu.

// Assert

// par défaut: sensible aux majuscules/minuscules

Assert.That(Result, Is.EqualTo("FizzBuzz"));

1. Exécutez le test

## Tester avec divisible par 3 seulement

1. Créez la méthode de test: GetOutput\_Div3\_ReturnFizz
2. Placez en commentaires les trois sections A:

public void GetOutput\_Div3\_ReturnFizz ()

{

// Arrange

// Act

// Assert

Assert.Pass();

}

1. Complétez les sections du test:
2. Exécutez le test
3. Faites une erreur volontaire dans le résultat attendu: fizz en minuscule
4. Exécutez le test
5. Ajoutez IgnoreCase

// Insensible aux majuscules/minuscules

Assert.That(Result, Is.EqualTo("fizz").IgnoreCase);

1. Exécutez le test

## Tester avec divisible par 5 seulement

1. Créez la méthode de test: GetOutput\_Div5\_ReturnBuzz
2. Placez en commentaires les trois sections A:

public void GetOutput\_Div5\_ReturnBuzz ()

{

// Arrange

// Act

// Assert

Assert.Pass();

}

1. Complétez les sections du test:
2. Exécutez le test

## Tester avec NON divisible par 3 ou 5

Suggestion: Affichez la classe **E1\_FizzBuzz** à côté de la classe de tests (clic droit sur une des classe, Nouveau groupe de documents vertical)

1. Créez la méthode de test: GetOutput\_NonDiv3OrDiv5\_ReturnParametre
2. Placez en commentaires les trois sections A:

public void GetOutput\_NonDiv3OrDiv5\_ReturnParametre ()

{

// Arrange

// Act

// Assert

Assert.Pass();

}

1. Complétez les sections du test: Attention au type de données de retour: 7 versus "7"
2. Exécutez le test

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary** FCT Tests BuzzFizz
   * **Description**: Tests des scénarios BuzzFizz 1 méthode par scénario
5. Cliquez sur **Commit**
6. "Poussez" **Push** les modifications sur le repo en ligne **Remote**: cliquez sur 

# Tester FizzBuzz 1 méthode plusieurs scénarios

## Tester FizzBuzz

1. Créez une nouvelle classe public de tests FizzBuzz\_Tests\_v2
2. Ajoutez la décoration [TestFixture] à la classe et la référence au framework NUnit
3. Ajoutez la référence au projet Appl\_TestsUnitaires
4. Ajoutez la méthode de tests GetOutPut\_Div3OrAnd5\_ReturnFizzBuzz, avec la décoration (annotation) [Test]
5. Placez en commentaires les trois sections A:
6. Ajoutez les quatre TestCase et passez-les en paramètres dans la méthode

[Test]

[TestCase(15, "FizzBuzz")]

[TestCase(3, "Fizz")]

[TestCase(5, "Buzz")]

[TestCase(7, "7")]

public void GetOutPut\_Div3OrAnd5\_ReturnFizzBuzz(int response, string expectedResult)

{

// Arrange

1. Le Arrange est vide: remplacé par les testCase, dans ce cas-ci.
2. Complétez les étapes du test

[Test]

[TestCase(15, "FizzBuzz")]

[TestCase(3, "Fizz")]

[TestCase(5, "Buzz")]

[TestCase(7, "7")]

public void GetOutPut\_Div3OrAnd5\_ReturnFizzBuzz(int response, string expectedResult)

{

// Arrange

// Act

var result = FizzBuzz.GetOutput(response);

// Assert

Assert.That(result, Is.EqualTo(expectedResult));

}

1. Exécutez le tes

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary** FCT Tests BuzzFizz
   * **Description**: Tests des scénarios BuzzFizz 1 méthode plusieurs scénarios
5. Cliquez sur **Commit**
6. "Poussez" **Push** les modifications sur le repo en ligne **Remote**: cliquez sur 