

EXERCICE « *FizzBuzz* (Open/closed principle) »

CFPT-EI

Octobre 2017

Préambule

Les exercices de cette série seront effectués avec *Visual Studio* en mode *Console*.
Nom de la solution : `ExFizzBuzzTDD`

1 Mise en situation

Vous êtes chargé de concevoir, écrire et tester la/les classes permettant de réaliser une application remplissant le cahier des charges du problème classique « *FizzBuzz* » :

- Write a program that prints the numbers from 1 to 100 :*
- *for multiples of three print « Fizz » instead of the number*
 - *for multiples of five print « Buzz » instead of the number*
 - *for numbers which are multiples of both three and five print « Fizz-Buzz » instead of the number*

Sortie console

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
...
98
Fizz
Buzz
```

2 Travail à réaliser

2.1 *Test Driven Development (TDD)*

En utilisant une approche *TDD* (c'est à dire l'écriture des tests unitaires avant le code des classes), concevez une première version de la classe `FizzBuzzer`, cf. Fig.1

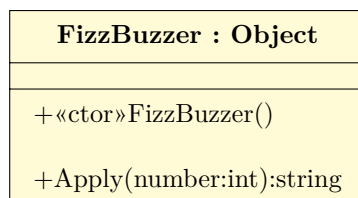


FIGURE 1 – diagramme de la classe `FizzBuzzer`

2.2 Respect du principe « *Open/closed* »

2.2.1 Définition

Le respect du principe « *Open/closed* »¹ demande que les entités logicielles (classes, méthodes...) soient :

- ouvertes pour l'extension
- fermées pour la modification

En d'autres mots, l'ajout de fonctionnalité doit se faire en ajoutant une/des classes **sans modifier les classes existantes**.

Peut-on ajouter une fonctionnalité ?

- sans modifier les classes existantes
- en créant une/des nouvelles classes
- et en modifiant les paramètres passés au constructeur

Si oui : 😊 vos classes respectent le principe « *Open/closed* ».

Si non : ☹️ vous devez retravailler (*refactor*) les classes.

2.2.2 Vérification pour la classe `FizzBuzzer`

Le client désire ajouter la fonctionnalité « *for multiples of seven print « Bang » instead of the number* ».

Peut-on ajouter cette fonctionnalité ?

- sans modifier la classe `FizzBuzzer`
- en créant une/des nouvelles classes
- et en modifiant les paramètres passés au constructeur de `FizzBuzzer`

2.2.3 Retraitail de la classe `FizzBuzzer`

Retravaillez la classe `FizzBuzzer` afin de respecter le principe « *Open/closed* ».

1. https://en.wikipedia.org/wiki/Open/closed_principle

3 Travail à rendre

Déposez sur *Moodle* un fichier ZIP (`votre_nom.zip`) contenant :
— le projet *Visual Studio* `ExFizzBuzzTDD`