

Fizz Buzz

# Table of Contents

1. 仕様	1
2. 設計	1
2.1. TODOリスト	1
2.2. ユースケース図	1
2.3. クラス図	1
2.4. シーケンス図	1
3. 実装	2
3.1. テストコード	2
3.2. プロダクトコード	2
4. 参照	3

# 1. 仕様

- 3で割り切れる場合は「Fizz」を出力する。
- 5で割り切れる場合は「Buzz」を出力する。
- 両方で割り切れる場合は「FizzBuzz」を出力する。
- 指定された回数だけ繰り返し実行する。

## 2. 設計

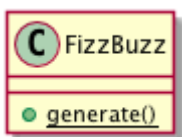
### 2.1. TODOリスト

- ☒ 「Fizz」を出力できるようにする
- ☒ 「Buzz」を出力できるようにする
- ☐ 「FizzBuzz」を出力できるようにする
- ☐ 繰り返し実行できるようにする

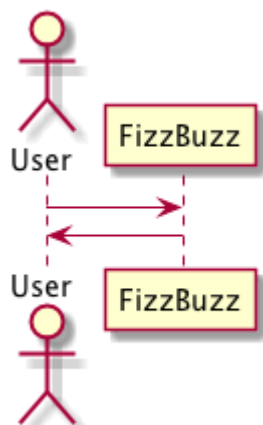
### 2.2. ユースケース図



### 2.3. クラス図



### 2.4. シーケンス図



## 3. 実装

### 3.1. テストコード

```
import pytest
from fizz_buzz.fizz_buzz import FizzBuzz

class TestFizzBuzz(object):
    def test_3ならばFizzを返す(self):
        assert FizzBuzz.generate(3) == 'Fizz'

    def test_6ならばFizzを返す(self):
        assert FizzBuzz.generate(6) == 'Fizz'

    def test_5ならばBuzzを返す(self):
        assert FizzBuzz.generate(5) == 'Buzz'

    def test_10ならばBuzzを返す(self):
        assert FizzBuzz.generate(10) == 'Buzz'

    def test_50ならばBuzzを返す(self):
        assert FizzBuzz.generate(50) == 'Buzz'

    def test_15ならばFizzBuzzを返す(self):
        assert FizzBuzz.generate(15) == 'FizzBuzz'

    def test_30ならばFizzBuzzを返す(self):
        assert FizzBuzz.generate(30) == 'FizzBuzz'
```

### 3.2. プロダクトコード

```
class FizzBuzz:
    @staticmethod
    def generate(number):
        value = number

        if value % 3 == 0 and value % 5 == 0:
            value = 'FizzBuzz'
        elif value % 3 == 0:
            value = 'Fizz'
        elif value % 5 == 0:
            value = 'Buzz'

        return value
```

## 4. 参照

- AsciiDoctor[<http://asciidoctor.org/>]
- PlantUML[<http://www.plantuml.com>]