Fizz Buzz

Table of Contents

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。
- 両者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。

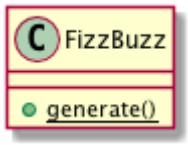
2. 設計

2.1. TODOリスト

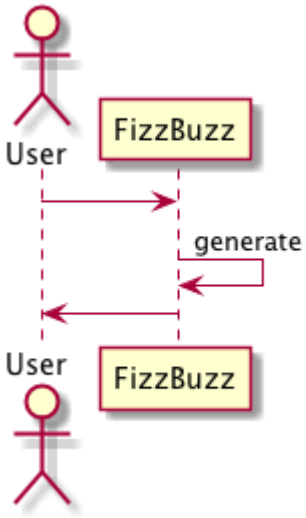
- [x] 「Fizz」を出力できるようにする
- [x] 「Buzz」を出力できるようにする
- [ ] 「FizzBuzz」を出力できるようにする
- [ ] 繰り返し実行できるようにする

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

```
const chai = require('chai');
const FizzBuzz = require('../../src/fizz_buzz');

const { expect } = chai;

describe('Tests FizzBuzz', () => {
  it('3ならばFizzを返す', () => {
    expect(FizzBuzz.generate(3)).to.equal('Fizz');
  });

  it('6ならばFizzを返す', () => {
    expect(FizzBuzz.generate(6)).to.equal('Fizz');
  });

  it('30ならばFizzを返す', () => {
    expect(FizzBuzz.generate(30)).to.equal('Fizz');
  });

  it('5ならばBuzzを返す', () => {
    expect(FizzBuzz.generate(5)).to.equal('Buzz');
  });

  it('10ならばBuzzを返す', () => {
    expect(FizzBuzz.generate(10)).to.equal('Buzz');
  });

  it('50ならばBuzzを返す', () => {
    expect(FizzBuzz.generate(50)).to.equal('Buzz');
  });
});
```

3.2. プロダクトコード

```
module.exports = class FizzBuzz {
  static generate(number) {
    let value = number;
    if (number % 3 === 0) {
      value = 'Fizz';
    } else if (number % 5 === 0) {
      value = 'Buzz';
    }
    return value;
  }
};
```

4. 参照

- Asciidoctor[http://asciidoctor.org/]
- PlantUML[http://www.plantuml.com]