

Rapport 68

Préliminaire

Non-respect de la nomenclature précise (-1). Dans les paramètres de la fonction solution, on a : `(String texte, List<String> ordre)`. Or, il est attendu : `(String str, List<Character> ordre)`

Simplicité

En terme de simplicité, ce programme correspond à mes attentes. Il est simple à comprendre et à modifier grâce aux commentaires et aux noms de variables explicites. Mais aussi grâce à la simplicité de la syntaxe Java.

Tests

Il passe tous les tests fournis, sauf le premier. Je pense que son erreur vient de son `split` qui ne prend en compte que les caractères non-alphabétiques, ce qui pose problème pour le mot "666".

Expected [[666, the, the, number, of, beast]] but was [[the, the, number, of, beast,]] testSolution()	
Expected	Actual
-[666, the, the, number, of, beast]	+ [the, the, number, of, beast,]

Parmi les tests supplémentaires :

```
// Test avec une chaîne contenant des mots spéciaux
assertEquals(List.of("Bonjour", "le", "monde", "!", "Comment", "ça", "va", "?"), exercice.simplicité.SimplicitéMeilleur.solution("Bonj

// Test avec une chaîne contenant des caractères spéciaux
assertEquals(List.of("abc", "def", "ghi", "jkl"), exercice.simplicité.SimplicitéMeilleur.solution("abc@def#ghi$jkl", List.of("a", "d",
```

Il ne passe pas le premier test car il ne prend pas en compte les mots spéciaux (ici "!" et "?").

Expected [[Bonjour, le, monde, !, Comment, ça, va, ?]] but was [[Bonjour, le, monde, ... testSoluti... ↑ ↓ ↺	
Expected	Actual
-[Bonjour, le, monde, !, Comment, ça, va, ?]	+ [Bonjour, le, monde, Comment, va, , ça]

Il passe le deuxième test car il prend en compte les caractères spéciaux.

Note : 16,5/20

Détail

- -1 pour le non-respect de la nomenclature
- -1 pour le premier test supplémentaire qui ne passe pas
- -2 pour le premier test fourni qui ne passe pas
- +0,5 pour car le test supplémentaire est dur à passer (peut être même faux) => je saurais l'expliquer à l'oral