

Rapport TP6

ROCCA Sébastien - M1 G2 ILSN

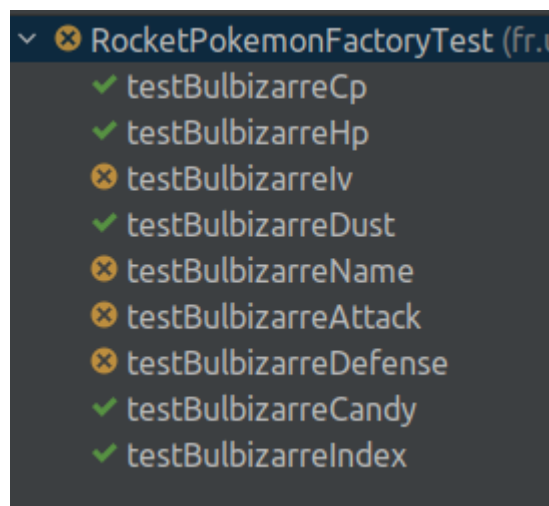
Introduction

J'ai ajouté la classe `RocketPokemonFactory` dans le package `java.fr.univavignon.pokeded.api` puis crée une classe spéciale de test `RocketPokemonFactoryTest` qui fait exactement les mêmes tests que ma classe `IPokemonFactoryTest` développée précédemment.

Tests

Lors des tests que j'ai effectués en instanciant la classe `RocketPokemonFactory` et en la comparant avec les résultats attendus à partir du Pokémon Bulbizarre, j'ai observé que différents tests ont échoués.

En particulier les tests pour obtenir les IV (`testBulbizarreIv`), le nom du Pokémon (`testBulbizarreName`), son attaque (`testBulbizarreAttack`) ainsi que sa défense (`testBulbizarreDefense`).



Test ayant échoué

Les valeurs ne correspondant pas à celle désirée.

Explications

Lorsqu'on analyse le code de la classe `RocketPokemonFactory` on peut découvrir plusieurs "problèmes".

- L'attribut `index2name`, la `map` initialisée n'est pas correcte :
 - L'indice -1 n'est pas valide (les indices des Pokémon allant de 0 à 151).
 - L'indice 0 ne correspond pas à Bulbizarre.
 - Qui lui est déplacé en indice 1 et n'est pas correcte ("Bulbasaur" au lieu de "Bulbizarre").
- La méthode `createPokemon` :
 - La condition `if` de départ pour attribuer un nom de Pokémon n'est pas valide si l'indice du Pokémon est 0 (retournera toujours `MISSINGNO`).
 - La condition `if` selon l'index saisi pour créer le Pokémon n'est pas valide, car un Pokémon ne peut pas avoir un index inférieur à 0 en temps normal.
 - Et le `else` correspondant génère une défense, une attaque et une endurance (stamina) aléatoire (en appelant la méthode `generateRandomStat`) ne correspondant donc pas aux attributs par défaut du Pokémon souhaité.

Le Pokémon généré par cette version implémentant l'interface `IPokemonFactory` n'est donc jamais valide.