

Rapport

Analyse du Code

L'implémentation fournie présente plusieurs problèmes que nous avons identifiés :

1. Utilisation inefficace de la méthode `generateRandomStat`

- La méthode `generateRandomStat` effectue un million de boucles pour générer une statistique, ce qui est extrêmement inefficace.
- Un calcul basé sur un générateur pseudo-aléatoire standard suffirait à produire une statistique aléatoire.

2. Gestion des indices non valides

- Les indices négatifs (comme `-1`) sont mappés à des valeurs fixes (par exemple, `Ash's Pikachu`).
- Bien que fonctionnel, ce comportement pourrait ne pas respecter les spécifications de l'interface `IPokemonFactory`.

Résultats des Tests

Nos tests existants ont permis d'identifier plusieurs problèmes :

Tests Passés

- Les tests pour la création de Pokémon avec des indices valides et non valides passent avec succès.
- Les attributs fixes (nom, cp, hp, dust, candy) sont correctement assignés.

Tests Échoués

- Les statistiques générées pour `attack`, `defense`, et `stamina` peuvent dépasser la plage `[0, 1000]`.
- La méthode `generateRandomStat` est inefficace et affecte les performances globales.