

Développement orientée objet - Java

TP 08 - Partie 1

Lionnel Conoir, Isabelle Delignières, Wajdi Elleuch et Rémi Synave

Ce TP est la suite du tdt05 durant lequel vous avez développé une classe `Pokemon`. Cette classe est générique et permet de créer tous les pokemons possible et imaginable mais elle n'est vraiment pas pratique.

En effet, pour créer un pokemon d'une certaine espèce (prenons un alakazam), vous devez aller chercher dans le pokedex toutes ses statistiques et ensuite créer le pokemon comme suit :

```
Pokemon alakazam1 = new Pokemon("BigAlakazam", 65, "psy", 55, 50, 45, 120);
```

ou encore

```
Pokemon alakazam2 = new Pokemon("Alakazamounet", 65, "psy", 55, 50, 45, 120);
```

Il serait bien plus pratique d'avoir un type `Alakazam` pour créer nos pokemons. D'autant plus que un alakazam est un pokemon.

On pourrait alors créer des alakazam comme suit :

```
Alakazam alakazam1 = new Alakazam("BigAlakazam");
```

```
Alakazam alakazam2 = new Alakazam("Alakazamounet");
```

TODO

1. Développez une classe pour représenter une espèce de pokemon (Alakazam si vous voulez). Réutilisez au mieux le code existant. Pensez à l'héritage.
2. Faites la même chose pour une seconde espèce.
3. Créez une méthode abstraite

```
public abstract String getEspece();
```


dans la classe `Pokemon` et faites le nécessaire pour que tout fonctionne.
4. Ajoutez l'appel à cette méthode dans la méthode `toString` de la classe `Pokemon`.
5. ~~Faites de même pour les 149 espèces restantes.~~ Étudiez le code de vos deux classes. Quelles sont les différences ? Peut-on retrouver ces informations quelque part de manière automatique ? Pensez vous qu'il est possible d'automatiser l'écriture des classes ? (la réponse est oui, rendez vous à la partie 2).