

TD n°13 – Héritage, polymorphisme, collections

Le projet Pokemons

Consignes

- Ce projet se réalise sur plusieurs séances, par groupe de 2 étudiants.
 - Votre enseignant responsable en TD vous donnera ses indications de rendu.
 - Le code doit être correctement commenté (pas trop, ni trop peu), robuste (gestion des exceptions), et associé à des TestCase JUnit.
 - Un travail minimum est demandé lors de chaque séance de TD. La réalisation de ce travail est obligatoire et sa bonne réalisation assure la moyenne au groupe.
 - Des extensions seront suggérées ou laissées au libre choix des étudiants. Toute extension correctement implémentée améliore la note du groupe.
 - Au contraire, tout code similaire entre groupes diminue la note de tous les groupes concernés.
 - Lors du DS, une partie des questions portera sur ce projet.
-

Les Pokemons

- 1^{ère} partie obligatoire

Les classes des Pokemons

Les Pokemons sont de gentilles créatures passionnées par la programmation objet en général et par le polymorphisme en particulier.

Nous commençons par créer une classe `Pokemon`. Un `Pokemon` se définit par (entre autres) :

- Un nom, une taille, et un poids ;
- Un attribut pv correspondant à ses points de vie ;
- Un attribut pc correspondant à sa puissance de combat ;
- Un type :

En Java, le type d'un `Pokemon` sera représenté par un **type énuméré** qui permet de définir et d'utiliser un ensemble fixe de constantes comme des variables objets. **Lire la documentation sur les types énumérés (<http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>) et implémenter pour les types des Pokemons décrits plus bas;**

- Une méthode `estVivant()` indiquant si un `Pokemon` est toujours en vie (`pv > 0`) ou non ;
- Une méthode `attaquer(Pokemon p)` permettant au `Pokémon` courant d'attaquer le `Pokemon` passé en paramètre. Cette méthode inflige des dégâts aux points de vie des Pokemons. Elle sera implémentée plus spécifiquement plus tard.
- Une redéfinition de la méthode `toString()` héritée de la classe `Object` qui affiche les informations du `Pokémon`.

Les Pokemons décrits à travers la classe `Pokemon` servent de base à la modélisation de ces petites créatures. Le travail consiste maintenant à décrire les Pokemons en fonction de leur type. Il existe plus d'une quinzaine de types différents¹. Nous nous concentrons ici sur quatre catégories :

¹ [http://www.pokepedia.fr/Catégorie:Pokémon par type](http://www.pokepedia.fr/Catégorie:Pokémon_par_type)

1. Les Pokemons de type EAU : ils possèdent en plus un nombre de nageoires. Ces Pokemons ne se déplacent que dans la mer à une vitesse = $(\text{poids} * \text{nombre de nageoires}) / 25$;
2. Les Pokemons de type FEU : ils possèdent en plus un nombre de pattes. Ces Pokemons se déplacent sur la terre à une vitesse = $\text{nombre de pattes} * \text{poids} * 0,03$;
3. Les Pokemons de type PLANTE : ils se déplacent sur la terre très lentement à une vitesse = $10 / (\text{poids} * \text{taille})$;
4. Les Pokemons de type ELECTRIK : ils possèdent un nombre de pattes, un nombre d'ailes, ainsi qu'une intensité. Au contraire des Pokemons de type PLANTE, ils se déplacent sur la terre ou dans les airs très vite à une vitesse = $(\text{nombre de pattes} + \text{nombre d'ailes}) * \text{intensité} * 0,05$.

Les Pokemons de type EAU sont super efficaces contre les Pokemons de type FEU et leur infligent deux fois plus de dégâts ($2*pc$). Par contre, ils sont très peu efficaces contre les Pokémon de type PLANTE ou de type EAU et ne leur infligent que la moitié des dégâts ($0.5*pc$). Ils infligent des dégâts normaux aux Pokémon de type ELECTRIK.

Les Pokemons de type FEU sont super efficaces contre les Pokemons de type PLANTE et leur infligent deux fois plus de dégâts ($2*pc$). Par contre, ils sont très peu efficaces contre les Pokemons de type EAU ou de type ELECTRIK et ne leur infligent que la moitié des dégâts ($0.5*pc$). Ils infligent des dégâts normaux aux Pokémon de type FEU.

Les Pokemons de type ELECTRIK sont super efficaces contre les Pokemons de type EAU et leur infligent deux fois plus de dégâts ($2*pc$). Par contre, ils sont très peu efficaces contre les Pokemons de type ELECTRIK ou de type PLANTE et ne leur infligent que la moitié des dégâts ($0.5*pc$). Ils infligent des dégâts normaux aux Pokémon de type FEU.

Enfin, les Pokemons de type PLANTE sont super efficaces contre les Pokemons de type ELECTRIK et leur infligent deux fois plus de dégâts ($2*pc$). Par contre, ils sont très peu efficaces contre les Pokémon de type PLANTE ou de type FEU et ne leur infligent que la moitié des dégâts ($0.5*pc$). Ils infligent des dégâts normaux aux Pokémon de type EAU.

- **Créez quatre classes `PokemonFeu`, `PokemonEau`, `PokemonPlante`, et `PokemonElectrik` qui héritent de la classe `Pokemon` et qui représentent les quatre types de Pokemons susmentionnés.**
- **Ajouter les constructeurs, accesseurs, ainsi que les méthodes `attaquer` et `calculerVitesse`. Factoriser la déclaration de la fonction `calculerVitesse`.**
- **Mettre à jour la méthode `toString()` affichant par exemple pour un Pokemon de type ELECTRIK :**

```
« Je suis le Pokemon Pikachu (type ELECTRIK). Mon poids est de 6 kg,
ma vitesse est de 2,5 km/h. J'ai 2 pattes, ma taille est de 0,4m et
mon intensité est de 50 mA. »
```

Ou de type FEU :

```
« Je suis le Pokemon Salameche (type FEU). Mon poids est de 12 kg,
ma vitesse est de 3,9 km/h. J'ai 2 pattes, ma taille est de 0,65m. »
```

Ou encore :

```
« Je suis le Pokemon Carapuce (type EAU). Mon poids est de 12 kg,
vitesse est de 1,9 km/h. J'ai 4 nageoires, ma taille est de 0,81m. »
```

Et enfin :

« Je suis le Pokemon Bulbizarre (type PLANTE). Mon poids est de 15 kg, ma taille est de 0,7m, ma vitesse est de 0,9 km/h. »

Les tests unitaires

Définir des *TestCase* sur les Pokemons permettant de tester les différentes classes et les méthodes *estVivant*, *calculerVitesse* et *attaquer*.