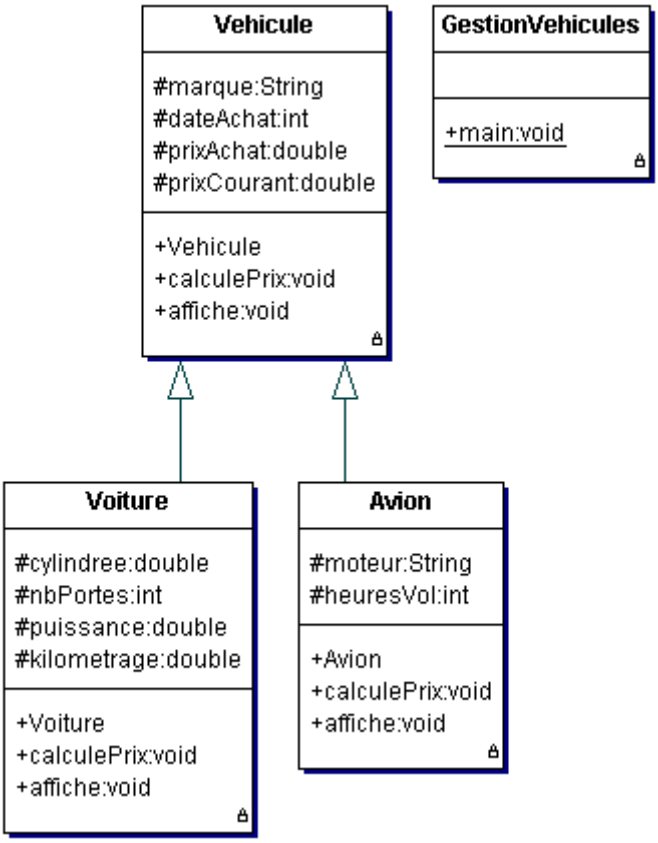
TD n°5 – Héritage et Polymorphisme

Introduction : Flotte de véhicules



Reprendre l'exercice de la semaine précédente dont les classes sont rappelées ci-dessus. Améliorez la méthode `main` en tenant compte du fait que tous les véhicules ont un type commun: `Vehicule`.

Les points colorés

Reprenez la classe `ColoredPoint` qui hérite de la classe `Point`. Dans le main, créer un tableau de points pouvant référencer à la fois des points « simples » et des points colorés.

Ce tableau contiendra :

- 5 points de coordonnées (0.0, 5.0), (1.0, 4.0), (2.0, 3.0), (3.0, 2.0), (4.0, 1.0) ;
- 5 points colorés de coordonnées (1.0, 1.0), …, (5.0, 5.0) et de couleurs rouge, bleu, vert, noir, blanc (peu importe la correspondance coordonnées-couleur).

1. Vous ferez appel aux différentes méthodes des classes `Point` et `ColoredPoint` sur ce tableau (`toString, getDistance, projX, projY, equals`, …).
2. Pour chaque objet du tableau, tester les méthodes `instanceof` et `getClass().getName()` vues en cours, pour observer les effets du polymorphisme.
3. Enfin, créer un nouvel objet de type `ColoredPoint` et affectez-lui le point bleu contenu dans votre tableau. Quelle manipulation devez-vous faire et pourquoi ?

Pokemons

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

Le but de cet exercice est de partir des classes données en annexe et de factoriser le code.

Dans SupportCours, vous trouverez un répertoire pokemons contenant les classes correspondant à 4 types de Pokemons : eau, electrik, feu et plante.

Vous devez :

1. Inclure ce code dans votre Java Project TD4 dans Eclipse.
2. Lire et comprendre le code des classes. En particulier, le type d'un Pokemon est 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 comprendre le code de la classe enum Type;**
3. Créer la classe `Pokemon` dans laquelle vous factoriserez le code commun à tous les types de Pokemons. Cette classe devra regrouper les attributs communs et les méthodes communes.
4. Ecrire dans chaque classe une méthode attaquer. Vous devez **factoriser le type** du paramètre de la fonction afin de permettre à chaque type de Pokemons d'attaquer n'importe quel autre Pokemon (aidez-vous des méthodes vues en cours pour connaître le type effectif du paramètre). La méthode doit vérifier les conditions suivantes :

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.

Lever une exception dans le cas où le type du paramètre n'est pas bon.

5. Ecrire une méthode main permettant de mettre en œuvre le polymorphisme et tester les différentes méthodes des classes.