TP 4 : L'héritage, la nourriture et d'autres intéractions !

Aujourd'hui nous allons nous amuser en créant de divers types de nourriture, en utilisant l'héritage. Puis, nous allons continuer créer des interactions entre les joueurs et leurs pokemons : les joueurs pourront nourrir et caresser leurs pokemons.

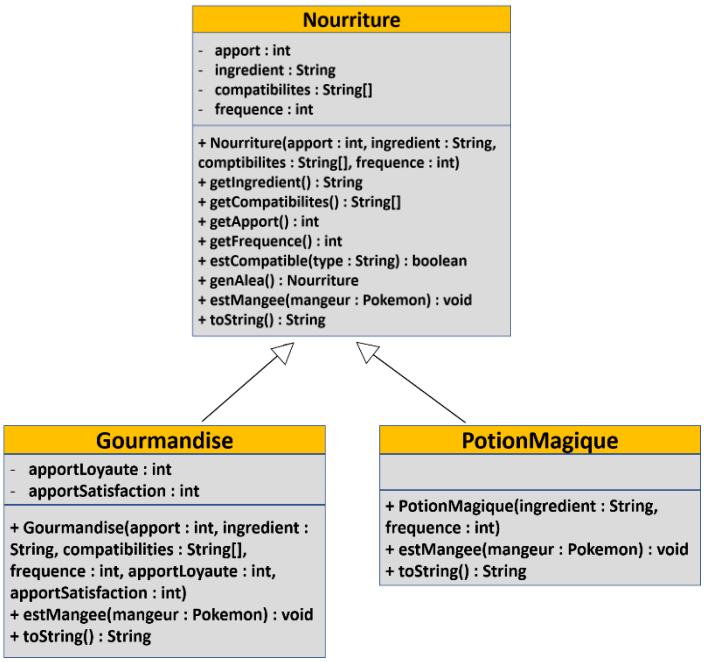
Lien utile : https://pokemondb.net/pokedex/all .

1. Pour l'instant, la nourriture existait à part des pokemons -- elle était générée, affichée, et on pouvait chercher ses compatibilités avec un certain type de pokemon déjà dans notre entourage.
   Aujourd'hui nous allons commencer en permettant aux pokemons de manger la nourriture. Juste pour s'entrainer, nous allons créer :

   - Une méthode avec la signature `void estMangee(Pokemon mangeur)` dans la classe Nourriture. À chaque fois qu'une nourriture est mangée, l'appétit du pokemon mangeur diminuera par l'apport nutritionnel de la nourriture (ou deviendra 0 si l'apport est supérieur à l'appétit du pokemon).
   - Une méthode avec la signature `void mange(Nourriture nourriture)` dans la classe Pokemon, qui appelle la méthode `estMangee(Pokemon mangeur)` , et qui assure donc la modification de l'appétit du pokemon.
   - Assurez-vous que les compatibilités d'un pokemon avec la nourriture sont bien prises en compte dans vos méthodes.
   - Testez les deux méthodes dans votre classe principale : premièrement essayez de faire manger à un pokemon un type de nourriture avec lequel il est bien compatible. Est-ce que son appétit a bien diminué ? Continuez à nourrir le pokemon pour voir que son appétit ne baisse jamais en dessous de 0. Essayez après de nourrir un pokemon qui n'est pas compatible avec cette nourriture. Qu'est-ce qui se passe ?

2. Il y aura divers types de nourriture dans ce jeu. Le type le plus commun, c'est celui qui soulage seulement l'appétit d'un pokemon. Mais on aura également de la nourriture spéciale, qui changera également la satisfaction et la loyauté des pokemons. Finalement, il y a également des types de nourriture qui peuvent modifier d'autres attributs des pokemons -- par exemple leurs niveaux.

   Nous allons utiliser l'héritage pour créer deux sous-classes de Nourriture -- correspondant à la nourriture qui change la satisfaction/loyauté des pokemons, et à celle qui modifie leurs niveaux. Le diagramme de classe suivant montre la relation entre les trois classes.



- Modifiez le constructeur de la classe Nourriture pour prendre en compte les fréquences variables de divers types de nourriture. Le nouveau constructeur aura la signature Nourriture(int, String, String[], int) -- le dernier paramètre indiquera la valeur de la fréquence. Vous allez devoir prendre en compte ce nouveau constructeur dans la méthode genAlea() de la même classe.
- Créez une nouvelle classe Gourmandise. Dans le dialogue de création de la classe, trouvez l'endroit où on indique qu'un classe aura une superclasse (cette valeur est mise à Java.lang.Object par défaut). Changez ce texte par le nom de votre superclasse, notamment Nourriture.
- Au-delà des attributs et méthodes qu'elle hérite de la classe Nourriture, les objets de type Gourmandise vont également avoir deux attributs de plus, de type entier : apportSatisfaction et apportLoyaute. Ecrivez un constructeur pour cette classe, avec la signature Gourmandise(int, String, String[], int, int, int). Ce constructeur remplacera votre ancien constructeur hérité de la classe Nourriture : il prendra en entrée des valeurs pour l'apport, le nom de l'ingrédient, les compatibilités du pokemon, la fréquence avec laquelle elle est générée, l'apport de satisfaction et l'apport de loyaute. N'oubliez pas : lorsqu'on appelle la superclasse d'une classe donnée on utilise le mot super.
- Modifiez les méthodes String toString() est void estMangee(Pokemon mangeur) de la classe Nourriture pour prendre en compte le fait que les gourmandises contiennent plus d'attributs et modifient la satisfaction et la loyaute d'un pokemon. Attention : les attributs de la superclasse sont privés et alors ils ne sont pas accessibles directement via this.<nom d'attribut> dans la classe Gourmandise. Qu'est-ce qu'il faut utiliser pour y avoir accès aux attributs de la superclasse dans la sous-classe ?

- La classe Gourmandise hérite la méthode genAlea() de la classe Nourriture. Cependant, cette méthode retourne un objet de type Nourriture. Ecrivez une méthode genAlea() dans la classe Gourmandise qui retourne, soit un objet de type Gourmandise (si le numéro généré aléatoirement est inférieur à la fréquence), soit null. L'objet retourné de type Gourmandise a les mêmes paramètres que l'objet courant (this).

- Créez une méthode void miseANiveau() dans la classe Pokemon, qui ne fait qu'augmenter le niveau du pokemon par 1. Ceci sera une préparation pour la deuxième sous-classe de nourriture, PotionMagique. La mise à niveau des pokemons étant un processus compliqué, nous la laisserons pour un TP plus tard.
- La deuxième sous-classe de nourriture qu'on considère est la classe PotionMagique. Celle-ci ne soulage pas la faim d'un pokemon ; elle met le pokemon à niveau. Son apport nutritionnel est alors mis à 0 par défaut. Les potions magiques sont rares et elles sont compatibles avec tout type de pokemon.
- Répétez la procédure ci-dessus : faites un constructeur pour cette classe (signature PotionMagique(String, int), prennant en entrée le nom d'ingrédient, ainsi que sa fréquence), puis mettez à jour les méthodes String toString() et void estMangee(Pokemon mangeur).
- Refaites la méthode genAlea() pour la classe PotionMagique, de la même façon que vous l'avez fait pour la classe Gourmandise.

3. Pour tester les nouvelles méthodes et classes créées, on va créer une gourmandise et une potionMagique, comme par exemple :
    - barreChocolatee est un objet de la classe Gourmandise avec un apport nutritionnel de 20, un apport de satisfaction de 5 et un apport de loyaute de 7. Sa fréquence sera de 10. Vous pouvez choisir librement les compatibilités de cette nourriture.
    - mojito est un objet de la classe PotionMagique qui a une fréquence de 2 (elle est très rare).
    - Affichez ces deux types de nourriture. Faites-les manger par deux pokemons compatibles avec elles. Vérifiez, en affichant les statistiques du pokemon mangeur, si vos méthodes mange et estMangee marchent correctement.
    - Essayez de générer les deux types de nourriture et affichez les résultats.