TD/TP 5 : Hiérarchies de classes, polymorphisme

Exercice 1 (Une première hiérarchie de classes)**.**
Dans cet exercice, on veut modéliser de l'herbe, des souris et des fennecs. L'herbe peut pousser (méthode `pousse` qui affiche « Je pousse »). Les souris et les fennecs peuvent :

- manger : méthode `mange` qui affiche « Je mange de l'herbe » pour les souris et « je mange une souris » pour les fennecs.
- dormir : méthode `dors` qui affiche « Je dors ».

Herbe, souris et fennecs peuvent évoluer : méthode `evolue` qui lance les méthodes `pousse` pour l'herbe, et `mange` puis `dors` pour les souris et les fennecs.

(1) Sur papier : écrire une classe `Souris` qui implante les spécifications ci-dessus.

(2) Sur papier : idem pour `Fennec`.

(3) Quelle duplication constatez-vous ?

(4) Sur papier : concevoir une hiérarchie de classes qui supprime cette duplication. Préciser quelles méthodes et quelles classes sont abstraites. Rajouter la classe `Herbe` dans cette hiérarchie.

(5) Implanter cette hiérarchie de classes, en commençant par `Fenec`, `Souris` et `Animaux`. Dans chaque classe concrète, inclure une petite fonction `main` illustrant son utilisation. Il s'agit d'une expérimentation qui a vocation à être jetée ; aussi on pourra exceptionnellement se passer de tests unitaires.

Exercice 2 (Polymorphisme)**.**
On considère une classe `Polymorphisme1` avec une fonction `main` contenant les lignes suivantes :

```
Souris souris = new Souris() ;
Fennec fennec = new Fennec() ;
Animal a = souris ;
Animal b = fennec ;
a.mange() ;
b.mange() ;
a.evolue() ;
b.evolue() ;
Souris c = a ;
Souris d = (Souris) a ;
Souris e = (Souris) b ;
```

Traiter sur papier toutes les questions suivantes. Ensuite vérifiez vos réponses sur l'ordinateur.

(1) Quelles lignes déclenchent une erreur de compilation ? Mettez les en commentaire.

(2) Quelles lignes déclenchent une erreur à l'exécution ? Mettez les en commentaire.

(3) Tracez pas-à-pas l'exécution en mémoire du programme, en précisant ce qu'il affiche. Quelles décisions sont prises par le compilateur ? Par la machine virtuelle ?

Exercice 3 (Auto-quizz)**.** – Soit C une classe abstraite avec un constructeur par défaut. Peut-on créer une instance de C avec new C() ?
– Quelle est le rôle d'une méthode abstraite ?
– Une classe concrète peut-elle contenir des méthodes abstraites ? Pourquoi ?
– Une classe abstraite peut elle contenir des méthodes concrètes ?
– En Java, une classe peut-elle hériter de plusieurs classes ?

Exercice 4 (Boucle « for each »)**.** (1) Implanter une petite application `ForEach` qui :
– Alloue un tableau `t` de Souris de longueur quatre.
– L'initialise avec quatre souris.
– Fait évoluer tout ce petit monde d'une itération avec une boucle `for` usuelle.

(2) Faites évoluer tout ce petit monde d'une autre itération avec une boucle « for each » :

```
for (Souris souris : t) {
  souris.evolue()
}
```

(3) Quel est l'avantage ?

Exercice 5 (Polymorphisme suite)**.**
Implanter une petite application `Polymorphisme2` qui :

(1) Alloue un tableau `t` d'EtresVivants de longueur sept.

(2) L'initialise avec une souris, de l'herbe, deux fennec, encore de l'herbe et encore deux souris.

(3) Fait évoluer tout ce petit monde avec une unique boucle « for each ».

Exercice 6 (Optionnel)**.**
Continuez d'améliorer SimSavane 2D.