NFA023/Android
Programmation orientée objet/Exercices

2011-2012

Exercice 1

On veut simuler un combat pour un jeu de rôle, dont les personnages ont :
- un nom ;
- un nombre de points de vies (un entier) ;
- une classe d'armure (un entier entre 0 et 20) ;

Pour frapper un adversaire, on doit jeter un dé à 20 faces. Si le dé est inférieur à la classe d'armure, on le touche, et on lui retire 1 point de vie.

On rappelle que `Math.random()` renvoie un nombre aléatoire dans [0,1[. Pour avoir un nombre entier aléatoire entre 1 et 20, on écrit

```
int dé= (int) (Math.random() * 20 + 1);
```

Question 1

Proposez des variables d'instance pour la classe `Personnage`.

Question 2

Complétez la classe `Personnage` :

```
public class Personnage {
/** Constructeur */
public Personnage(...) {}
public String getNom() {...}
public int getClasseArmure() {...}
public void retirerPointsDeVie(int pdvPerdu) {....}
/** frappe un autre personnage. Retourne true si touché */
public boolean frapper(Personnage p) {....}
/** Dit si le personnage est vivant */
public boolean estVivant() {...}
public void frapper(
/** Informations sur le personnage pour affichage */
public String toString() {}
}
```

Question 3

Ecrire un petit programme en mode console pour simuler un combat entre deux personnages, Beowulf et Grendel, jusqu'à ce que l'un d'entre eux gagne.

Question 4

On propose d'ajouter la notion d'arme. Un personnage peut avoir ou non une arme. Sans arme, il fait 1 point de vie de dégâts. Une arme donne un bonus b au jet de dé, et a un dégât variable compris entre 1 et n, n étant le dégât maximal de l'arme.

Avec une arme, on touche si

D20 ¡ Classe Armure cibre + bonus arme attaquant

Créer la classe Arme, et ajoutez la possibilité pour un personnage d'avoir une arme.

Remarque

On voit bien que le résultat de la procédure « frapper » n'est pas assez détaillé. Celle-ci pourrait retourner, non un booleen, mais un `ResultatAttaque`, qui donnerait tous les détails et permettrait un affichage.

Exercice 2

On veut écrire une classe durée, pour manipuler et comparer des durées exprimées en heures et minutes.

Les objets de la classe doivent être immuable (la valeur d'un objet créé ne doit jamais changer).

On désire avoir les méthodes suivantes :

- un constructeur (au moins) auquel on fournit heures et minutes ;
- les accesseurs nécessaires ;
- une méthode pour savoir si deux durées sont égales
- une méthode pour savoir si une durée est plus grande (strictement) qu'une autre
- une méthode pour ajouter deux durées (voir condition ci-dessus)
- une méthode toString(), qui renvoie la durée correctement formatée.

Exercice 3 : modélisation objet

On souhaite modéliser la notion de recette de cuisine, avec l'idée de pouvoir par exemple choisir une recette en fonction de ses ingrédients, et calculer le nombre de calories d'une recette.

Une recette a un titre, un texte (qui décrit le mode opératoire et ne nous intéresse pas plus que ça dans cet exercice), et des ingrédients. Les ingrédients sont des produits, dont les quantités sont exprimées en unités qui dépendent du produit. Par exemple, le lait sera exprimé en litres. Chaque produit a une certaine quantité de calories, de protéines, glucides et lipides par unité.

Proposez une architecture d'objets pour représenter la recette, sachant qu'on veut pouvoir calculer simplement les informations nutritionnelles de celle-ci.

Écrire les classes correspondantes.