

TP Programmation Orientée Objet

Langage Java

Sujet

On souhaite modéliser un jeu de cartes faisant intervenir différents types de personnages. Ces personnages sont caractérisés par leur nom et un nombre de points de vie initial attribué aléatoirement. Ces personnages peuvent combattre et/ou soigner. Dans le cadre de ce TP les personnages se restreindront à :

- des guerriers pouvant combattre
- des soigneurs pouvant soigner
- et des paladins pouvant combattre et soigner.

Les paladins ne combattent ni ne soignent de la même façon que les guerriers ou les soigneurs. Les personnages pouvant combattre ont des points d'attaques qui déterminent le nombre de points de vie que perdra leur adversaire lorsqu'ils le blessent.

Les personnages pouvant soigner ont des points de soin qui déterminent le nombre de points de vie qu'ils redonnent à un personnage lorsqu'ils le soignent.

On associe également aux personnages un message permettant de décrire chaque action (combat et soin) qu'ils font.

La classe `java.util.Random` permet de générer des nombres aléatoire par l'intermédiaire par exemple de la méthode `int nextInt(int n)` qui retourne un nombre aléatoire compris entre 0 et n.

Question 1

Modéliser les différents personnages, en veillant à avoir plusieurs personnages de chaque type (par exemple plusieurs guerriers avec des caractéristiques/attaques différentes). Cette modélisation devra utiliser des classes abstraites et interfaces et avoir au moins 4 niveaux de hiérarchie.

Coder en Java cette modélisation.

Question 2

Ajouter une méthode `String allInfo()` qui retourne une chaîne de caractères qui servira à l'affichage des caractéristiques des personnages. Vous pourrez utiliser les méthodes :

- `getClass()` : retourne la classe de l'objet sur lequel on l'applique
- `getName()` : retourne le nom de la classe sur laquelle on l'applique
- `getSuperclass()` : retourne la super classe de la classe sur laquelle on l'applique

Question 3

Implémenter des guerriers, soigneurs et paladins qui combattront et soigneront différemment les uns des autres. Voici quelques exemples :

- l'attaquant ne blesse son adversaire que si celui-ci est proche : à chaque attaque on pourra déterminer aléatoirement la distance entre les personnages

- certains combattant font plus de dommages lorsqu'ils sont loin
- certains combattants peuvent subir eux-mêmes des dommages lorsqu'ils attaquent
- le soigneur ne peut soigner que si le blessé est proche
- le soigneur perd des points de vie lorsqu'il soigne
- la récupération du blessé prend plusieurs jours
- ...

Question 4

Tester votre programme avec plusieurs personnages.

Vous devez arriver ici à la fin du premier TP

Question 5

Modifier votre programme pour que l'ensemble des personnages intervenant dans votre jeu soit dynamique en utilisant des collections. Vous pourrez éventuellement rajouter une méthode permettant de créer à la volée des personnages.

Question 6

Ajouter ce qu'il faut pour traiter les exceptions.

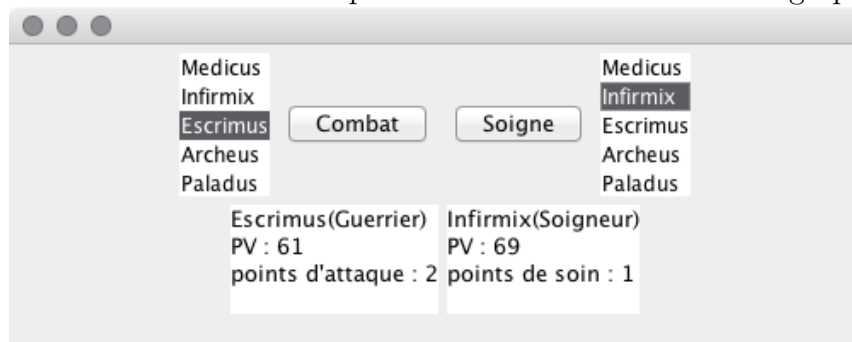
Vous devez arriver ici à la fin du deuxième TP

Question 7

Le composant graphique `JList` permet d'afficher un ensemble d'éléments sous forme de liste et permet de sélectionner un ou plusieurs éléments. Son constructeur fait appel à la méthode `toString()` de chaque élément pour l'afficher dans la liste. Veillez donc à avoir une méthode `toString()` pour vos personnages.

Question 8

Ecrire la classe `Fenetre` permettant d'obtenir l'interface graphique suivante :



Question 9

La méthode `addListSelectionListener` permet de rajouter un écouteur de type `ListSelectionListener` sur un composant `JList`. Ajouter un écouteur sur chaque composant `JList` afin que les caractéristiques du personnage sélectionné s'affichent dans la zone de texte correspondante.

Question 10

Ecrire le code des écouteurs des boutons combat et soigne.

Question 11

Comment faire pour que chaque personnage soit créé à la volée (sans prévoir de classe explicite dans la hiérarchie) et ait sa propre façon de soigner et/ou combattre ? Coder cette solution.