

## Fantaisie Travaux Pratiques n°5

« hashCode, HashSet, HashMap,  
EnumMap, TreeMap, classe dégénérée »

Ce TP poursuit le TP4 sur les monstres. Afin d'avoir toujours une version stable par sujet de TP :

- Ouvrir Eclipse, sélectionner le même workspace que pour le premier TP (par exemple : Z:\Licence3\Semestre1\POO2\workspace).
- Copier le projet fantaisieTP4 (en le sélectionnant puis ctrl+c),
- Coller le projet (ctrl+v), et renommer le fantaisieTP5.

Dans ce TP vous travaillerez donc uniquement dans ce nouveau projet.

### 1. Les armes

Toutes les armes ne permettent pas de combattre dans tous les milieux. Par exemple un lance-pierre est inefficace dans l'eau ...

Ajouter dans la classe Arme un attribut zonesDeCombat de type « HashSet » et générer son getter.

Ajouter au constructeur un vararg de type « ZoneDeCombat » qui permettra d'initialiser l'attribut zonesDeCombat.

Ci-dessous un tableau récapitulatif des zones de combat par arme :

Type d'arme	Zone de combat possible
Arc	Aquatique, Aérien, Terrestre
Boomerang	Aérien, Terrestre
Epée	Aquatique, Terrestre
Lance-pierre	Aérien, Terrestre

### 2. Les Hommes

#### Gérer ses armes

Ajouter l'attribut armes de type « EnumMap » dans la classe « Homme » qui associe une liste d'armes à une zone de combat que possède l'homme.

Ajouter la méthode *ajouterArmes* possédant un paramètre d'entrée : un vararg contenant les armes à ajouter. L'arme devra être ajoutée dans chaque liste où elle est efficace.

Ajouter la méthode *supprimerArme* possédant un paramètre d'entrée : l'arme à supprimer. Attention il faut supprimer l'arme dans chaque liste de chaque zone.

### Choisir une arme adaptée

Créer un attribut armeChoisie de type « Arme » et générer son getter.

Créer la méthode *choisirArme* qui possède un paramètre d'entrée : le monstre à combattre et un paramètre de sortie : l'arme choisie.

Récupérer la zone de combat du monstre pour récupérer la liste d'armes adéquate (à condition d'avoir des armes permettant de se battre dans cette zone sinon la méthode retournera null).

Créer un ensemble trié armesTriees dans l'ordre naturel contenant toutes les armes de la liste. L'ordre naturel d'une arme porte :

- Sur le fait qu'elle soit opérationnelle ou non : l'ordre portera d'abord sur l'arme opérationnelle.
- Si les deux armes sont opérationnelles le tri se fera sur le nombre de points de dégât qu'elles occasionnent.
- Enfin si les deux armes ne sont pas opérationnelles ou qu'elles occasionnent le même nombre de point de dégât alors le tri portera sur l'ordre alphabétique de leur nom.

Si l'ensemble armesTriees n'est pas vide, récupérer la force de vie du monstre à combattre et créer une vue armesAdaptees en ne gardant que les armes qui occasionnent le même nombre de dégât ou un nombre inférieur. Si cette liste n'est pas vide affecter l'attribut armeChoisie avec la première arme de la liste armesAdaptees, sinon on prendra la dernière arme de la liste triée armesTriees. Retourner l'arme choisie.

Par exemple :

- On veut tuer un *guillotimort*, monstre terrestre (80 pts de vie) et que l'on a un arc (50pts de dégât) et un boomerang (20pts de dégât). On choisira l'arc car son nombre de dégât est le plus proche de 80 tout en restant inférieur.
- On veut tuer un *aqualave*, monstre marin (30 pts de vie). On peut utiliser l'arc (50pts de dégât) ou l'épée (80pts de dégât). Aucune arme n'a un nombre de dégât inférieur ou égal donc on utilisera l'arc dont le nombre est le plus proche de 30.

**Aide :** Pour pouvoir créer la vue armesAdaptees il faut utiliser un objet issu de la classe dégénérée « KeyArme » ayant un constructeur ne prenant que les points de dégât générés et lui passer la force de vie du monstre.

### 3. Sélection des personnages dans le camp des hommes

Le groupe des hommes sont tous les hommes et héros qui entrent dans la grotte. Mais dans chaque salle seulement 3 hommes avec les armes les plus appropriées combattront le camp des monstres. Dans cette partie nous devons choisir les trois personnages les mieux armés et ayant le plus de force pour entrer dans la bataille dans le camp des hommes.

**Le groupe des hommes**

Créer la classe « GroupeHommes » contenant un seul attribut groupe de type « TreeSet ». Créer la méthode *ajouterHommes* qui ajoute des hommes au groupe. Pour cela, cette méthode possède un vararg de type « Homme » en paramètre d'entrée.

**Le comparateur d'hommes**

Créer la classe interne « ComparateurHommes » qui compare deux hommes sur leur force restante (du plus fort au moins fort), puis selon l'ordre naturel (l'ordre alphabétique de leur nom).

**Le comparateur d'armes**

Créer la classe interne « ComparateurArmes » qui contient un attribut monstre initialisé par le constructeur.

Récupérer la force de vie du monstre et les points de dégât occasionnés par les deux armes.

- Si les forces d'attaque sont différentes alors placer ces deux armes dans un « TreeMap » classementForce avec pour clé les points de dégât et pour valeur l'arme.

La meilleure arme est celle dont les points de dégât sont égaux ou immédiatement en dessous de la force de vie du monstre. S'il n'y en a pas alors la meilleure arme sera celle ayant le moins de points de dégât.

Il n'y a plus qu'à trancher : si la meilleure arme est la deuxième alors retourner un nombre positif, sinon un nombre négatif.

- Si les forces d'attaque sont identiques alors retourner le résultat de leur ordre naturel.

**Le choix des participants à la bataille dans le camp des hommes**

Créer la méthode *choixDuCamp* qui utilisera les deux comparateurs que l'on vient de créer. Cette méthode possède un paramètre d'entrée : la bataille dans laquelle les hommes entreranno et un paramètre de sortie : la liste des hommes allant combattre.

Dans un premier temps nous allons trier les hommes en fonction de l'adéquation de leurs armes avec le monstre à combattre et dans un deuxième temps nous en sélectionnerons 3.

Première partie :

- A partir du paramètre d'entrée bataille récupérer un monstre de la liste du camp des monstres (méthode *sélectionner* que vous avez copié dans la classe « Camp » depuis Moodle au TP4).
- Créer un « TreeMap » hommesArmes avec pour clé une arme et pour valeur un ensemble d'hommes possédant cette arme. Les clefs seront triées dans l'ordre défini par le comparateur de la classe interne « ComparateurArmes » et les valeurs triées par le comparateur de la classe interne « ComparateurHommes ».

- Pour chacun des hommes du groupe lui faire choisir sa meilleure arme (*choisirArme* de la classe « Homme »). S'il a effectivement une arme et que cette arme se trouve dans hommesArmes alors ajouter l'homme considéré à l'ensemble associé à l'arme. Si elle ne se trouve pas dans hommesArmes alors créer une nouvelle entrée.

Deuxième partie :

- Récupérer les hommes de l'arme la mieux placée. S'il n'y en a pas assez (moins de 3) passer aux hommes de l'arme en seconde position etc, jusqu'à avoir 3 hommes ou que la liste soit vide.
- Pour chaque homme sélectionné, le faire rejoindre la bataille.

Tester : Télécharger sous Moodle le fichier « TestGestionArmes » et le placer dans le paquetage « *testsfonctionnels* »