Travaux Dirigés
Langage Orienté Objet / Java

Partie 4 – Collections

The Learning Souls Game

Ce TD doit être réalisé à partir du code créé dans la partie précédente.

Nous avons jusqu'à présent mis en place des mécanismes qui font perdre de la vie et de la stamina à nos personnages, ainsi que de la durabilité à leurs armes. Les combats, même lorsqu'ils sont gagnés, peuvent être rudes, et il est nécessaire de fournir à nos protagonistes les moyens de se rassasier et de réparer. Nous allons donc créer des consommables (***consumables***) sous la forme de nourriture (***food***), de boissons (***drinks***) et de kits de réparation (***repair kits***) permettant de remettre tout ce qui doit l'être en état !

NB : nous allons dans cette partie plutôt nous focaliser sur des tests avec le héro. Cependant, les consommables et les sacs concernent aussi les monstres. En effet, il n'est pas aberrant d'imaginer que nos monstres puissent aussi (dans une version future du jeu) utiliser des consommables au cours des combats.

1. Character et Weapon

Les consommables vont permettre de régénérer des ressources intimement liées à certaines classes. Ce sont en particulier les points de vie et la stamina des instances de **Character**, ainsi que la durabilité des instances de **Weapon**. De manière à en avoir une représentation de référence dans tout le jeu, nous allons créer des constantes statiques dans les classes concernées. Ces constantes seront publiques, puisque leur but est d'être utilisées au besoin par les autres classes du jeu.

1.1. Créez :
- **Character.LIFE_STAT_STRI**NG, une chaine contenant « ***life*** ».
- **Character.STAM_STAT_STRING**, une chaine contenant « ***stamina***».
- **Weapon.DURABILITY_STAT_STRING**, une chaine contenant « ***durability***».

1.2. Transformez les **toString()** de **Character** et **Weapon** pour qu'elles utilisent maintenant ces chaines (au lieu de la version « en dur » que nous avions implémentées précédemment)

1.3. Même si on ne les utilisera pas directement dans cette partie, généralisez l'approche pour d'autres statistiques du jeu (protection, buff)

2. Consumable

Tous les consommables ont plusieurs points communs. Ils ont un nom (**name**) et offrent une certaine capacité (**capacity**) à régénérer une des statistiques (**stat**) du jeu. Ils proposent une méthode **use()** à déclencher pour les utiliser : elle a pour effet de retourner un montant de points de régénération et de les consommer, c'est à dire de diminuer la capacité du consommable.

2.1. Créez la classe **consumables.Consumable** avec :

- Les attributs privés **String name**, **int capacity** et **String stat**.
- Un constructeur public à 3 paramètres permettant de fixer son nom, sa capacité et la le nom de la statistique à laquelle il est dédié.
- Les getters publics pour les 3 attributs.
- Surcharge de la méthode **toString()** qui retourne une chaine de la forme :

  nom [capacité statistique point(s)]

  Exemple : « ***Hot Coffee [10 stamina point(s)]*** »

- Le setter **setCapacity** qui servira en interne (y compris dans les sous-classes !) à modifier la capacité lors d'un **use()**.
- La méthodes publique **use()** qui par défaut retourne le montant total de la capacité, et fait passer l'attribut correspondant à 0. (Le consommable est alors vide : il ne reste plus que l'emballage…)

2.2. Créez **consumables.drinks.Drink**, sous classe de **Consumable** qui représente un consommable lié à la stat **stamina** de **Character**.

- Créez **consumables.drinks.Coffee**, **consumables.drinks.Whisky** et **consumables.drinks.Wine**, 3 sous-classes de **Drink** dont le **toString** (avant consommation) est :

  Hot Grandmother Coffee [10 stamina point(s)]

  12 years old Oban [150 stamina point(s)]

  Pomerol 2008 [30 stamina point(s)]

2.3. Créez **consumables.food.Food**, sous classe de **Consommable** qui représente un consommable lié à la stat **life** de **Character**.

- Créez **consumables.food.Hamburger et consumables.food.Americain**, 2 sous-classes de **Food** dont le **toString** (avant consommation) est :

  Uncle Greg's spicy Maroilles burger [40 life point(s)]

  Friterie 2000's Best of the Best [3000 life point(s)]

3. Les « Menus Best OF »

Histoire de créer des menus équilibrés (…), nous allons proposer des menus contenant divers consommables.

3.1. Créez la classe **consumables.MenuBestOfV1** avec :

- Un attribut **menu** de type tableau de **Consumable** de taille 5 et contenant un **Hamburger**, du **Wine**, un **Americain**, un **Coffee**, et un **Whisky**.
- La méthode **toString()** dont l'appel produit une chaine telle que :

```
MenuBestOfV1 :
1 : Uncle Greg's spicy Maroilles burger [40 life point(s)]
2 : Pomerol 2008 [30 stamina point(s)]
3 : Friterie 2000's Best of the Best [3000 life point(s)]
4 : Hot Grandmother Coffee [10 stamina point(s)]
5 : 12 years old Oban [150 stamina point(s)]
```

- Un **main** pour tester la création du menu et son affichage

3.2. Créez la classe **consumables.MenuBestOfV2** par copier/coller de la version 1 et remplacez le tableau **menu** par un **java.util.HashSet<Consumable>**.

- Testez le **main**

3.3. En dupliquant **MenuBestOfV2**, créez **MenuBestOfV3** en faisant en sorte qu'il hérite de **HashSet<Consumable>** : il n'a donc plus besoin de l'attribut **menu** (que l'on supprimera !).

- Testez le **main**.
- Notez que pour **MenuBestOfV2** et **MenuBestOfV3** l'affichage du menu ne respecte pas l'ordre d'ajout des consommables. Avec un ajout dans le même ordre qu'en 3.1, on peut obtenir :

```
MenuBestOfV2 :
1 : Pomerol 2008 [30 stamina point(s)]
2 : Uncle Greg's spicy Maroilles burger [40 life point(s)]
3 : Hot Grandmother Coffee [10 stamina point(s)]
4 : 12 years old Oban [150 stamina point(s)]
5 : Friterie 2000's Best of the Best [3000 life point(s)]
```

- Regardez la documentation officielle de **java.util.HashSet** sur https://docs.oracle.com/javase/8/docs/api/java/util/HashSet.html.
  Qu'est-il précisé à propos de l'itération ?

3.4. Regardez la documentation officielle de **java.util.LinkedHashSet**.

Qu'est-il précisé au niveau de l'itération ?

- Créez par copier/coller le **MenuBestOfV4** qui hérite non plus de **HashSet**, mais de **LinkedHashSet**.
- Testez le **main** et vérifiez que vous avez bien compris le point de documentation que nous venons de regarder.

4. Character

Maintenant que les consommables sont disponibles, nous allons faire en sorte que nos **Character** puissent les utiliser.

4.1. Créez une méthode ***privée*** **drink** qui prend en paramètre une boisson (**Drink**).

- La méthode utilise la boisson et remonte le niveau de stamina du personnage avec le montant retourné par l'appel à **use()**.
  Rappel : Le niveau de **stamina** d'un personnage ne peut dépasser la valeur de **maxStamina**. Si le chiffre retourné par **use** est trop grand, l'excédent de points est perdu (la statistique est remontée à son maximum).
- Pour faciliter les tests, la méthode affichera un message dans console :

```
Gregoninator drinks 12 years old Oban [150 stamina point(s)]
```

4.2. Créez une méthode ***privée*** **eat** qui prend en paramètre de la nourriture (**Food**).

- La méthode utilise la nourriture et remonte le niveau de vie du personnage avec le montant retourné par l'appel à **use()**.
  Rappel : Le niveau de vie d'un personnage ne peut dépasser la valeur de **maxLife**. Si le chiffre retourné par **use** est trop grand, l'excédent de points est perdu (la statistique est remontée à son maximum).
- Pour faciliter les tests, la méthode affichera un message dans console :

```
Gregoninator eats Uncle Greg's spicy Maroilles burger [40 life point(s)]
```

4.3. Créez la méthode **public void use(Consumable consumable)** qui fait appel à **drink** ou **eat** en fonction du type de consommable. (cf. **instanceof**).

5. LearningSoulsGame

Pour tester tout cela, nous allons créer 2 méthodes dans **LearningSoulsGame**.

5.1. Créez la méthode **createExhaustedHero()** dont le but est de créer un héro totalement épuisé. Cette méthode :

- Instancie le héro.
- Le Frappe avec un coup à 99 (**getHitWith**)
- Lui donne une « ***Grosse Arme*** » qui épuisera toute sa stamina en une frappe (pas la peine de créer une classe) :

```
new Weapon( name: "Grosse Arme",  minDamage: 0,  maxDamage: 0,  stamCost: 1000,  durability: 100)
```

- Affiche un message et les stats du héro épuisé :

```
Create exhausted hero :
[ Hero ]            Gregoninator        LIFE:    1      STAMINA:    0      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
```

5.2. Créez la méthode **aTable()** qui :

- Instancie un **MenuBestOfV4**
- Fait consommer au héro les éléments de ce menu 1 par 1 (dans une boucle) en affichant les informations utiles (pour voir l'évolution des statistiques). Exemple pour 1 tour de boucle :

```
Gregoninator eats Uncle Greg's spicy Maroilles burger [40 life point(s)]
[ Hero ]            Gregoninator        LIFE:   41      STAMINA:    0      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Uncle Greg's spicy Maroilles burger [0 life point(s)]
```

5.3. Dans le **main** :

- Instanciez le **LearningSoulsGame**
- Exécutez **createExhaustedHero(),** puis **aTable()**
- Vérifiez qu'on obtient bien le résultat attendu au niveau des stats :

```
Create exhausted hero :
[ Hero ]            Gregooninator        LIFE:    1      STAMINA:    0      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)

Gregooninator eats Uncle Greg's spicy Maroilles burger [40 life point(s)]
[ Hero ]            Gregooninator        LIFE:   41      STAMINA:    0      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Uncle Greg's spicy Maroilles burger [0 life point(s)]

Gregooninator drinks Pomerol 2008 [30 stamina point(s)]
[ Hero ]            Gregooninator        LIFE:   41      STAMINA:   30      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Pomerol 2008 [0 stamina point(s)]

Gregooninator eats Friterie 2000's Best of the Best [3000 life point(s)]
[ Hero ]            Gregooninator        LIFE:  100      STAMINA:   30      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Friterie 2000's Best of the Best [0 life point(s)]

Gregooninator drinks Hot Grandmother Coffee [10 stamina point(s)]
[ Hero ]            Gregooninator        LIFE:  100      STAMINA:   40      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Hot Grandmother Coffee [0 stamina point(s)]

Gregooninator drinks 12 years old Oban [150 stamina point(s)]
[ Hero ]            Gregooninator        LIFE:  100      STAMINA:   50      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : 12 years old Oban [0 stamina point(s)]
```

6. RepairKit

En nous inspirant de ce qui a été fait précédemment, nous allons créer un type de consommable nommé **consumables.repair.RepairKit** qui permet de réparer une arme. A la différence des consommables précédents, un **RepairKit** ne fournit qu'un seul point de réparation à la fois lors de son utilisation.

6.1. Créez **consumables.reapir.RepairKit** qui :

- Hérite de **Consumable**.
- Possède un constructeur qui fixe le nom à « ***Repair Kit*** », la capacité initiale à 10 points, et la statistique visée à **Weapon.DURABILITY_STAT_STRING**
- Surcharge la méthode **use()** pour ne donner qu'un point à la fois.
  NB : de fait, la capacité ne diminue que d'une unité à chaque utilisation.

6.2. Dans la classe **Weapon** :

- Créez la méthode **public void repairWith(RepairKit kit)** qui remonte la durabilité de l'arme en utilisant le **use()** du kit.

6.3. Dans la classe **Character** :

- Créez la méthode **private void repairWeaponWith(RepairKit kit)** qui répare l'arme équipée avec le kit passé en paramètre et affiche un message du type :

```
Gregooninator repairs Grosse Arme (min:0 max:0 stam:1000 dur:99) with Repair Kit [10 durability point(s)]
```

- Modifiez la méthode **use(Consumable consumable)** pour qu'elle gère les consommables de type **RepairKit**.

7. Cadeau

7.1. Ajoutez une instance de **RepairKit** (!cadeau !) dans les **MenuBestOfV4** .

7.2. Modifiez **aTable()** pour qu'elle affichent les données de l'arme portée après consommation du menu.

7.3. Relancez le **main** de **LearningSoulsGame** pour vérifier les stats :

```
Create exhausted hero :
[ Hero ]            Gregooninator       LIFE:    1      STAMINA:    0      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)

Gregooninator eats Uncle Greg's spicy Maroilles burger [40 life point(s)]
[ Hero ]            Gregooninator       LIFE:   41      STAMINA:    0      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Uncle Greg's spicy Maroilles burger [0 life point(s)]

Gregooninator drinks Pomerol 2008 [30 stamina point(s)]
[ Hero ]            Gregooninator       LIFE:   41      STAMINA:   30      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Pomerol 2008 [0 stamina point(s)]

Gregooninator eats Friterie 2000's Best of the Best [3000 life point(s)]
[ Hero ]            Gregooninator       LIFE:  100      STAMINA:   30      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Friterie 2000's Best of the Best [0 life point(s)]

Gregooninator drinks Hot Grandmother Coffee [10 stamina point(s)]
[ Hero ]            Gregooninator       LIFE:  100      STAMINA:   40      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Hot Grandmother Coffee [0 stamina point(s)]

Gregooninator drinks 12 years old Oban [150 stamina point(s)]
[ Hero ]            Gregooninator       LIFE:  100      STAMINA:   50      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : 12 years old Oban [0 stamina point(s)]

Gregooninator repairs Grosse Arme (min:0 max:0 stam:1000 dur:99) with Repair Kit [10 durability point(s)]
[ Hero ]            Gregooninator       LIFE:  100      STAMINA:   50      PROTECTION:  0.00     BUFF:  0.00    (ALIVE)
Apres utilisation : Repair Kit [9 durability point(s)]
Grosse Arme (min:0 max:0 stam:1000 dur:100)
```

8. Consommer

Maintenant que les mécanismes liés aux consommables ont bien été mis en place, nous allons en équiper nos personnages afin qu'ils puissent les utiliser au cours des combats.

Comme dans la plupart des RPG, nos personnage vont avoir la capacité d'équiper un consommable (à la fois) et, au besoin, de le consommer au cours d'un combat.

8.1. Dans la classe **Character** :

- Créez un attribut **consumable** ainsi que les getters et setters publics correspondants.
- Créez une méthode **public void consume()** qui permet au personnage d'utiliser le consommable équipé.

8.2. Dans **LearningSoulsGame** :

- Modifiez la méthode **init()** pour que le héro soit équipé d'une instance de **Hamburger**.
- Créez une constante publique et statique de type String nommée **BULLET_POINT** et contenant « ***\u2219*** » : elle nous servira à insérer des puces dans les affichages de listes
- Modifiez la méthode **refresh()** pour qu'elle affiche (en plus) l'arme et le consommable équipés par le héro :

```
[ Hero ]            Gregooninator       LIFE:  100      STAMINA:   50      PROTECTION: 10.20     BUFF: 14.00    (ALIVE)
• Basic Sword (min:5 max:10 stam:20 dur:100)
• Uncle Greg's spicy Maroilles burger [40 life point(s)]

[ Lycanthrope ]     Lycanthrope         LIFE:   10      STAMINA:   10      PROTECTION: 30.00     BUFF:  0.00    (ALIVE)
```

- Transformez la méthode **fight1v1()** :
  - changez le type de la variable **action** en **int** et utilisez **scanner.nextInt()**.
    NB : **nextInt()** générera une erreur si l'utilisateur tape une chaine ne pouvant être convertie en entier. On ne traitera pas cette erreur pour l'instant (et tant pis si le programme plante).
  - lorsque c'est au tour du héro, on lit l'action en proposant 2 choix : attaquer (1) ou consommer (2). On reposera la question tant que l'utilisateur n'aura pas entré un chiffre correspondant à un de ces choix.
  - lorsque c'est au tour du monstre, on ne demande plus à l'utilisateur d'entrer une valeur : l'action est déclenchée automatiquement, et si la partie n'est pas terminée, on repasse au tour du héro.

- [optionnel] : créez une méthode **title()** qui affiche le titre du jeu dans la console

```
##############################
#   THE LEARNING SOULS GAME  #
##############################

[ Hero ]            Gregooninator       LIFE:  100      STAMINA:   50      PROTECTION: 10.20     BUFF: 14.00    (ALIVE)
• Basic Sword (min:5 max:10 stam:20 dur:100)
• Uncle Greg's spicy Maroilles burger [40 life point(s)]

[ Lycanthrope ]     Lycanthrope         LIFE:   10      STAMINA:   10      PROTECTION: 30.00     BUFF:  0.00    (ALIVE)

Hero's action for next move : (1) attack | (2) consume > 5
Hero's action for next move : (1) attack | (2) consume > 1

Gregooninator attacks Lycanthrope with Basic Sword (ATTACK:8 | DMG : 6)

[ Hero ]            Gregooninator       LIFE:  100      STAMINA:   30      PROTECTION: 10.20     BUFF: 14.00    (ALIVE)
• Basic Sword (min:5 max:10 stam:20 dur:99)
• Uncle Greg's spicy Maroilles burger [40 life point(s)]

[ Lycanthrope ]     Lycanthrope         LIFE:    4      STAMINA:   10      PROTECTION: 30.00     BUFF:  0.00    (ALIVE)

Lycanthrope attacks Gregooninator with Bloody Claw (ATTACK:67 | DMG : 60)

[ Hero ]            Gregooninator       LIFE:   40      STAMINA:   30      PROTECTION: 10.20     BUFF:10014.00  (ALIVE)
• Basic Sword (min:5 max:10 stam:20 dur:99)
• Uncle Greg's spicy Maroilles burger [40 life point(s)]

[ Lycanthrope ]     Lycanthrope         LIFE:    4      STAMINA:    5      PROTECTION: 30.00     BUFF:  0.00    (ALIVE)

Hero's action for next move : (1) attack | (2) consume > 2

Gregooninator eats Uncle Greg's spicy Maroilles burger [40 life point(s)]

[ Hero ]            Gregooninator       LIFE:   80      STAMINA:   30      PROTECTION: 10.20     BUFF: 14.00    (ALIVE)
• Basic Sword (min:5 max:10 stam:20 dur:99)
• Uncle Greg's spicy Maroilles burger [0 life point(s)]

[ Lycanthrope ]     Lycanthrope         LIFE:    4      STAMINA:    5      PROTECTION: 30.00     BUFF:  0.00    (ALIVE)

Lycanthrope attacks Gregooninator with Bloody Claw (ATTACK:63 | DMG : 57)

[ Hero ]            Gregooninator       LIFE:   23      STAMINA:   30      PROTECTION: 10.20     BUFF:10014.00  (ALIVE)
• Basic Sword (min:5 max:10 stam:20 dur:99)
• Uncle Greg's spicy Maroilles burger [0 life point(s)]

[ Lycanthrope ]     Lycanthrope         LIFE:    4      STAMINA:    0      PROTECTION: 30.00     BUFF:  0.00    (ALIVE)

Hero's action for next move : (1) attack | (2) consume > 1

Gregooninator attacks Lycanthrope with Basic Sword (ATTACK:708 | DMG : 4)

[ Hero ]            Gregooninator       LIFE:   23      STAMINA:   10      PROTECTION: 10.20     BUFF:10014.00  (ALIVE)
• Basic Sword (min:5 max:10 stam:20 dur:98)
• Uncle Greg's spicy Maroilles burger [0 life point(s)]

[ Lycanthrope ]     Lycanthrope         LIFE:    0      STAMINA:    0      PROTECTION: 30.00     BUFF:  0.00    (DEAD)

--- Gregooninator WINS !!! ---
```