

TP 1 - Revisited

Enseignant: Abdoul Majid Thiombiano/[@thiomajid](#)

Description du projet

Première étape d'une longue série, nous allons implémenter un *text-based RPG* appelé *ConsoleQuest*.

Au cours de ce TP, nous allons définir la structure de base de notre jeu via les classes `GameEngine`, `Weapon` et `Character`.

Etape 1

Pour commencer, nous allons créer un dossier nommé **ConsoleQuest** qui contiendra tout le code pour ce projet. Après quoi, créer un dossier nommé **tp1** dans lequel sera écrit le code pour ce TP.

Info

Chaque classe devra être définie dans **son propre fichier**.

Etape 2

Créer la classe `Character`. La classe *Character* est définie comme suit:

- Les attributs **privés** suivant: String name, double health, double speed, double attack.
- Ecrivez les méthodes `getter` et `setter` pour tous les attributs ci-dessus.
- Redéfinir la méthode `toString` de la classe *Character* pour retourner les informations du personnages sous la forme : `Character(name=<name>, health=<health>, speed=<speed>, attack=<attack>)`.

Exemple:

Les méthodes *getter* et *setter* pour l'attribut *name* sont comme suit:

```
public String getName(){
    return name
}

public void setName(String name){
    this.name = name;
}
```

NB:

La signature de la méthode `toString` est définie comme suit pour toutes les classes en Java.

```
@Override // ceci indique qu'on redéfinit la méthode toString
public String toString(){
    // vos instructions
}
```

Etape 3

Créer la classe `Weapon` dans son propre fichier. Elle est définie comme suit:

- Les attributs **privés** suivant: `String name`, `double attack`.
- Ecrivez les getter et setter pour tous les attributs de la classe.
- Redéfinir la méthode `toString` pour retourner les informations de l'objet sous la forme: `Weapon(name=<name>, attack=<attack>)`.

Etape 4

- Dans la classe `Character` ajouter l'attribut **privé** `weapon` de type `Weapon`.
- Définir la méthode `equipWeapon` qui permet au personnage d'équiper une arme de son inventaire. Lorsque le personnage équipe une arme, on ajoute l'attaque de la nouvelle arme à l'attaque du personnage. La méthode `equipWeapon` est définie comme suit:

```
public void equipWeapon(Weapon newWeapon){
    // vos instructions
}
```

- Définir la méthode `unequipWeapon` qui permet de retirer l'arme actuelle du personnage. Lorsqu'un personnage retire son arme, on décrémente son attaque la valeur de l'attaque de l'arme qu'il avait.

```
public void unequipWeapon(){
}
```

- Définir un *getter* pour l'attribut `weapon`.
- Modifier la méthode `toString` pour inclure les informations de l'arme de l'utilisateur si il en a une.

Comme l'attribut *weapon* de la classe *Character* est de type *Weapon* (qui est un type de référence) alors pour retirer l'arme on utilise l'instruction `weapon = null;`

Etape 5

Dans la classe `Character`, nous allons définir une nouvelle méthode nommée `attack` qui permet à un personnage d'en attaquer un autre. Lorsqu'un personnage attaque un autre le combat continue jusqu'à ce que l'attribut `health` de l'un des personnages soit **négatif ou nul**.

```
public void attack(Character otherPlayer){  
  
}
```

Exemple

On a deux joueurs *P1* et *P2* si *P1* attaque *P2* alors on a:

```
P2.health -= P1.getAttack();
```

Et si *P2* attaque *P1* alors on a l'instruction

```
P1.health -= P2.getAttack();
```

NB: Le combat se fait **tour par tour** et c'est le personnage le plus **rapide** qui commence le tour.

A la fin on affiche le nom du gagnant du combat sous la forme: *<personnage_1> a remporté le combat contre <personnage_2>*.

Etape 6

Créer la classe `GameEngine` qui sera chargée de gérer l'exécution de notre jeu.

La classe `GameEngine` est définie comme suit:

- La méthode `run` qui démarre le jeu. Lorsque le programme démarre le menu suivant est affiché à l'utilisateur:
 - 1: Jouer
 - 2: Quitter

```
public void run(){  
    // vos instructions  
}
```

- Si le joueur décide de *jouer* alors on crée un objet de type `Character` dont on saisira toutes les informations. Après créer une arme de votre choix que le personnage va équiper.
 - Afficher les informations de votre personnage.
 - Créer un deuxième personnage auquel vous allez attribuer une arme.
 - Faites combattre les deux personnages et afficher le nom du gagnant.
 - Afficher le message "*A la prochaine sur ConsoleQuest*" et arrêter le programme.
- Si l'utilisateur choisit de quitter le jeu alors on lui affichera le message "*A la prochaine sur ConsoleQuest*".

Etape 7

Créer le fichier `Program.java` qui contiendra la fonction principale de notre programme. Dans la fonction principale créer une instance de `GameEngine` et lancer l'exécution du jeu.

"Simplicity is the highest sophistication.", Leonardo Da Vinci.