



LA ROCHELLE UNIVERSITÉ

LICENCE INFORMATIQUE

Génie Logiciel - Qualité

Logicielle et Gestion de Projet

TP2 - Cas d'étude "*Jeu de rôle*"

Application du design pattern Decorator

1 Comment réaliser ce TP

Pour réaliser ce TP, vous utiliserez l'outil Visual Paradigm disponible sur les VM Linux de l'Université (image ULR - Ubuntu 22.04). Vous pouvez également installer la version Community Edition sur vos machines personnelles. Le développement en Java sera réalisé sur l'IDE de votre choix.

Le dépôt de vos travaux dans un fichier `nom_prenom.zip` sera à réaliser pour **dimanche 23 mars 18h**. Les dépôts effectués par mail ne seront pas pris en compte et une note de 0 sera appliquée. Ce TP est à réaliser individuellement.

```
nom_prenom.zip/  
├── nom_prenom.pdf (contenant tous les diagrammes)  
└── factory-vs-wild/ (contenant votre projet Maven)
```

2 Jeu de rôle - Description du cas d'étude

2.1 Rappel du cahier des charges

Dans le jeu de rôle *Factory vs. Wild*, les personnages et les monstres interagissent dans un monde dynamique. Chaque entité possède des caractéristiques spécifiques (vie, attaque, défense) et peut être affectée par des événements (prise de dégâts, état de santé, bonus d'équipement).

Les informations sur les personnages et les monstres sont enregistrées dans un fichier texte *characters_and_monsters.txt* dont voici un exemple :

```
# Format : Type; Nom; TypePersonnage; HP; ATK; DEF; Compétence
Personnage;Arthas;Guerrier;150;20;10;Coup de bouclier (Étourdit l'ennemi)
Personnage;Morgana;Mage;100;30;5;Boule de feu (Dégâts en zone)
Personnage;Legolas;Archer;120;25;8;Tir perçant (Ignore l'armure)
Personnage;Gandalf;Mage;130;35;4;Invocation d'esprit (Invoque une créature alliée)

# Format : Type; Nom; TypeMonstre; HP; ATK; DEF; Faiblesse; Résistance
Monstre;DragonRouge;Dragon;300;40;20;Glace;Feu
Monstre;OrcSanguinaire;Orc;180;25;10;Magie;Physique
Monstre;GolemDePierre;Golem;250;15;30;Eau;Physique
Monstre;SpectreNoir;Spectre;120;35;5;Lumière;Ténèbres
Monstre;Zombie;Mort_Vivant;100;10;5;Feu;Poison
```

2.2 (Re)mise en place du projet et vérification du travail

- Récupérez votre travail de la semaine dernière et vérifiez que votre projet Maven fonctionne toujours correctement.
- Récupérez le projet VisualParadigm de votre diagramme de classes. Le diagramme attendu à la fin du TP1 est donné en figure 1.

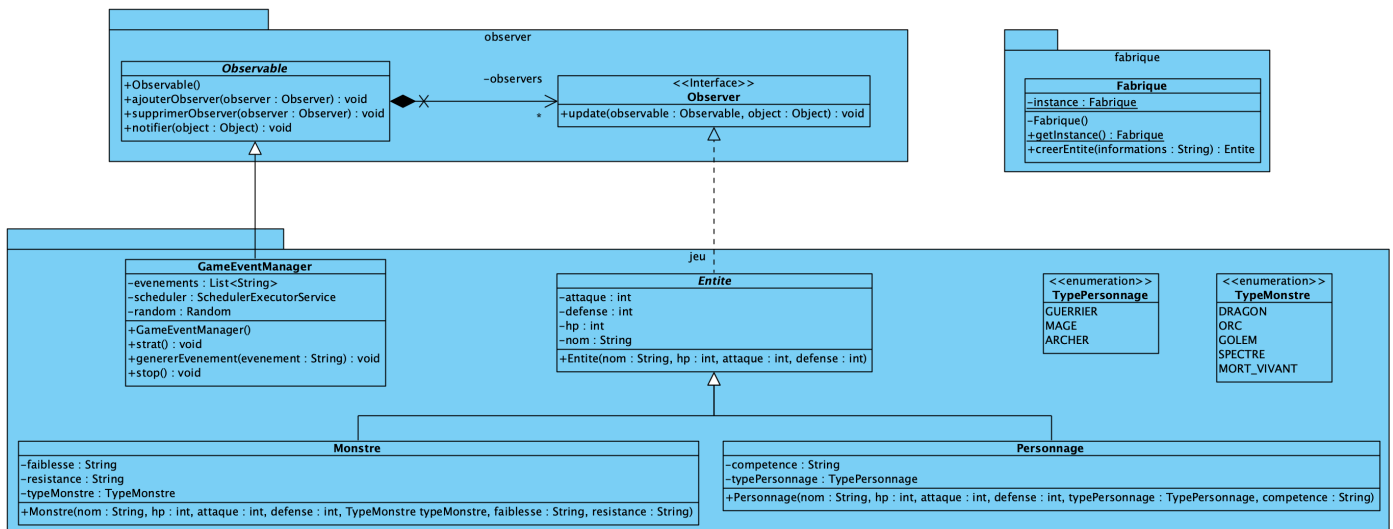


FIGURE 1 – Solution du diagramme de classes du jeu - v1



Appelez votre enseignant pour lui faire valider votre projet comme demandé dans le DM2.

3 Mise en place du design pattern Decorator

On souhaite ajouter dynamiquement des équipements aux personnages au cours de la partie. Ainsi, les équipements possibles sont :

- Epée enchantée (+10 points d'attaque)
- Bouclier mystique (+5 points de défense)
- Potion (+20 points de vie)

De la même manière, les monstres peuvent obtenir des améliorations :

- Enragé (+15 points d'attaque et -5 points de défense)
- Peau renforcée (+10 points de défense)

En vous inspirant de la description du design pattern Decorator vu en cours et lors du TD2 :

- proposez une intégration du design pattern Decorator dans le diagramme de classes précédent. Vous ajouterez pour cela un nouveau package.
- implémentez ce pattern en Java.

4 Les exceptions

La dernière partie de ce TP consiste en la mise en place d'exceptions qui pourront être déclenchées par la Fabrique lors de la création des entités :

- `FormatInvalideException` si une ligne du fichier n'a pas le bon nombre de paramètres;
- `TypeEntiteException` si le premier élément de la ligne n'est ni `Personnage` ni `Monstre`;
- `TypeMonstreException` si le type de monstre n'est pas reconnu;
- `TypePersonnageException` si le type de personnage n'est pas reconnu.

Après avoir consulté la documentation sur les exceptions,

- ajoutez ces 4 classes dans votre diagramme en ajoutant un package `exception`;
- implémentez ces classes et ajoutez la gestion des exceptions dans la classe `Fabrique`.

5 Les tests

Vous trouverez sur Moodle la classe `AppTest`. Téléchargez là et sauvegardez la dans `src > test > java > fr > lru`.

Lancez les tests avec la commande suivante :

```
mvn test
```

Vous n'avez pas le droit de toucher au contenu du fichier. A vous de modifier votre code pour que les tests passent!