

ASI Aventure

partie 1

30 janvier 2017

Remarque

- pas de gestion d'Exception
- pour chaque classe référez vous à la javadoc
- pas d'abstraction
- utilisation des collections interdites
- limitez vous à ce que vous avez vu en cours
- méthode toString() à faire systématiquement

Pour les spécifications il faut vous référer aux liens sur moodle, javadoc et modèle UML :

<https://moodle.insa-rouen.fr/mod/resource/view.php?id=23420>
https://moodle.insa-rouen.fr/pluginfile.php/355/mod_resource/content/0/Projet_ASIAventure/ASIAventure/index.html

1 classe Entite

1. constructeur
2. **Monde** (classe vide pour le moment)
3. **getNom** / **getMonde**
4. **toString**
5. **equals** (même nom / même monde)

2 classe Monde

La classe **Monde** va permettre le stockage de toute entité créée.

1. constructeur (*String*)
2. **getNom** / **getEntite**
3. **ajouter**, Remarques :
 - Pour le moment on considère qu'un utilisateur ne créera pas 2 fois la même entité dans le même monde ;
 - pas de equals dans monde, on considère que les **Monde** sont uniques de par leur instance. Sinon le problème suivant se pose : 2 **Monde** sont égaux s'ils ont le même nom et les mêmes entités or l'égalité d'**Entite** teste l'égalité des **Monde**.
4. lorsque la méthode **ajouter** fonctionne, penser à modifier le constructeur d'**Entite**
5. le **toString** du monde doit retourner son nom et les noms de ses entités.

3 types énumérés : Etat / EtatDuJeu

Les créer.

4 classe Objet

Faire la méthode `estDeplacable` concrète (pas de *abstract*), par défaut tout objet est non déplaçable.

5 classe PiedDeBiche

Il y a juste à faire le constructeur (attention un pied de biche est déplaçable).

6 classe ElementStructurel

Il y a juste à faire le constructeur.