Java : manipuler les collections

### **Modalités :**

* Travail en autonomie
* Production individuelle

### **Objectifs de l’activité :**

* Manipuler les collections Java
* Validation de la compétence n°11

### **Consignes :**

1. **Commencer par étudier ces tutoriels proposés :**

|  |
| --- |
| * OpenClassrooms : « Collections » * CodeCademy : « Data Structures » |

On va maintenant stocker nos personnages non plus dans un tableau, mais en utilisant une ArrayList.

1. **Reprendre le programme précédent :**

|  |
| --- |
| Le programme commence par vous proposer un premier menu dans lequel vous pouvez :     * Créer des personnages : pour cela, il va demander quel type de personnage vous voulez créer, puis vous demande les informations adéquates pour créer ce personnage, avec la saisie via lecture des entrées clavier (classe Scanner), des informations du personnage (Guerrier ou Magicien)   Faire la saisie de plusieurs personnages (boucle) jusqu’à ce que l’utilisateur choisisse de quitter le programme.  Les personnages seront stockés dans une ArrayList.   * + Lister tous les personnages existants   + Choisir un personnage pour   + afficher toutes ses infos   + le supprimer du tableau   + modifier ses infos |

1. **Implémentation du plateau de jeu :**

|  |
| --- |
| Maintenant que les personnages peuvent être créé, il faut maintenant être en mesure de créer le plateau de jeu sur lequel va évoluer notre personnage.  Un plateau consiste en une succession linéaire de cases. Ces dernières sont numérotées.  Chaque case peut contenir ou non un événement. On distingue deux types d'événement   * La case contient un objet qui améliorera les statistiques de votre héros * La case contient un ennemi qu’il faudra vaincre.   Pour le moment et dans un but de simplicité, les objets présents sur les cases ainsi que les ennemis peuvent être représentés par une simple chaine de caractère.  Dans cette première version du jeu il vous ai demandé de créer ce dernier en utilisant la collection de votre choix tant que cette dernière répond aux spécifications précédentes. |

|  |
| --- |
| **Livrables :**   * Code des classes demandées. A déposer sous git. * Glossaire enrichi des différentes syntaxes utilisées. Vous compléterez votre glossaire avec les principaux types de collections. A déposer sous le nom ***glossaire***, dans votre répertoire individuel sous ***java→livrables*** |

### **Ressources :**

* CodeCademy

<https://www.codecademy.com/en/courses/learn-java/lessons/data-structures/exercises/data-structures>

* Openclassrooms <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-collections-d-objets>
* Tutoriel d’Oracle pour Java

<http://docs.oracle.com/javase/tutorial/java/index.html>

|  |
| --- |
| ***Synthèse (activité de groupe)*** |

### **Modalités :**

* Travail en classe mixée

### **Objectifs de l’activité :**

* Activité de synthèse de type recherche d’erreurs.

### **Déroulement :**

* Slides fournis pour l’activité de synthèse
* Recherche individuelle puis mise en commun