Java : héritage et classes abstraites

### **Modalités :**

* Travail en autonomie
* Production individuelle

### **Objectifs de l’activité :**

* Implémenter l’héritage
* Manipuler des classes abstraites
* Gérer les exceptions
* Validation des compétences n° 13, 14 et 15

### **Consignes :**

1. **Manipulation des concepts d’héritage**

Contexte de modification du programme :

* Nous avons créé précédemment des classes Guerrier et Magicien. Ces Guerriers et ces Magiciens vont tous deux hériter d’une classe abstraite « Personnage » qui va permettre de généraliser un certain nombre d’attributs et de comportements   
    
  1. Etude de tutoriels

|  |
| --- |
| * *OpenClassrooms : « l’héritage »* * *Ressource* [*vidéo*](https://www.youtube.com/watch?v=On0X8RLMqko&list=PLABHMI-ayZf6DNjsiiSii8p4Axs2pDZbg&index=8) *(UPMC)* |

1. Modifier votre programme pour répondre aux spécifications suivantes :

|  |
| --- |
| 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 un tableau.   * + Lister tous les personnages existants   + Choisir un personnage pour   + afficher toutes ses infos   + le supprimer du tableau   + modifier ses infos |

1. **Gestion des exceptions :**

|  |
| --- |
| * Modifiez votre code pour générer une exception (simuler un plantage de l’application) * Tutoriel OpenClassrooms : « [Gestion des Exceptions](https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-exceptions) » * Gérer l’exception précédente. |

### 

|  |
| --- |
| **Livrable :**   * Code des classes demandées. A déposer sous git. * Glossaire enrichi des différentes syntaxes utilisées. A déposer sous le nom ***glossaire***, dans votre répertoire individuel sous ***java→livrables*** |

### **Ressources :**

* Tutoriel *Openclassrooms* Héritage :

<https://openclassrooms.com/courses/apprenez-a-programmer-en-java/l-heritage-1>

* Tutoriel *Openclassrooms* Exceptions :

<https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-exceptions>

* *Ressource vidéo (UPMC) :*

<https://www.youtube.com/watch?v=On0X8RLMqko&list=PLABHMI-ayZf6DNjsiiSii8p4Axs2pDZbg&index=8>

* *Tutoriel d’Oracle pour Java :*

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

* Tutoriel vidéo sur les notions d’accessibilité : <https://www.youtube.com/watch?v=ejerrSN4GbQ&list=PLZbs1ERZ-TGVtIKH119wo-CTcy7Z_2s1X&index=3>
* Openclassrooms “Classe abstraites” :

<https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-classes-abstraites-et-les-interfaces> (partie concernant les classes abstraites)

* Ressource video : EPFL classe et methodes abstraites : <https://www.youtube.com/watch?v=sZ3UecFEi8c>

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

### **Modalités :**

* Travail par groupe

### **Objectifs de l’activité :**

* Activité de synthèse de type écriture de code en commun.
* Ecriture des classes correspondant à un héritage fille/garçon, avec des attributs et des méthodes spécifiques.

### **Déroulement :**

* Écriture collective au tableau