

Héritage



Définition

L'héritage est un **concept fondamental** de la **Programmation Orientée Objet (POO)**. Celui-ci permet de créer des **classes** (classes filles) à partir de **classes existantes** (classes parentes) utilisées comme **modèle**.

Contexte : Vous devez créer les différentes classes de personnages jouables de votre JV préféré. Le jeu aura des mises à jour régulières, et des fonctionnalités classiques devront sûrement évoluer (pensez réutilisation...)

1 Création classe parente

Personnage

Notre classe **Personnage** sera notre modèle pour toutes les classes disponibles.

2 Création classes filles

3 classes héritant directement de **Personnage**.

DPS

Support

Tank

Héritage multiple :

lorsqu'une classe hérite de **plusieurs classes mères**.

Cette fonctionnalité n'est **PAS disponible** en Java (voir interfaces).



4 Héritage Hiérarchies

Personnage

Support

Soigneur

Une classe fille peut aussi être classe mère d'une autre classe. Le **Soigneur** est bien un **Support** qui est bien un **Personnage**.

3 Utilisation de l'héritage

Utilisation de l'héritage

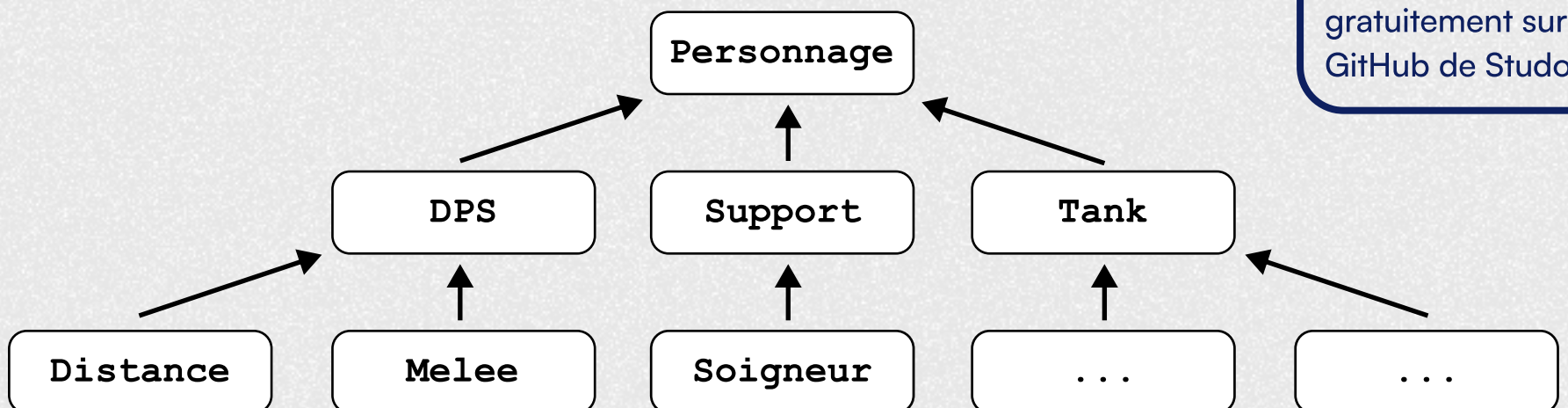
Les 3 classes filles peuvent accéder aux attributs / méthodes de **Personnage** (voir mot-clé **extends**).

5 Polymorphisme

Un **Soigneur** peut être traité comme s'il appartenait à la classe **Personnage**.

```
Personnage soigneur = new Soigneur(); // OK
soigneur.soigner(); // METHODE SPECIFIQUE = OK
soigneur.deplacer(); // METHODE GENERIQUE = OK
```

Sources disponibles gratuitement sur le GitHub de Studoby



N'importe quelle classe du jeu permet de se déplacer. La méthode **deplacer()** sera alors définie dans **Personnage**. Chaque classe fille pourra redéfinir (**@Override**) la méthode afin de fournir une implémentation spécifique (ex: un Tank ne peut pas 'autant' se déplacer qu'un DPS ou Support).