

EXERCICE 1:

1)

Ici, l'héritage est une mauvaise idée car même si tous les soldats ont une vitesse, une puissance de frappe et un certain nombre de point de vie total, ceux ci varient en fonction des types de soldats. D'un point de vue sémantique, tous ces soldats ne sont pas les fils d'un soldat "mère".

2)

En Java, les interfaces sont un exemple de pattern Composite. C'est un composite sur car en Java, toutes les méthodes d'une interfaces doivent être implémentées quand on l'implémente.

EXERCICE 2:

3)

Il faudrait faire en sorte que les méthodes utilisées pour le camion et les soldats soient les même. Il suffit donc de créer une interface Military.

4)

La classe ArmyTruck n'est pas une réalisation du pattern Composite car il ne contient pas des Military mais il contient de Soldier. De ce fait, un camion ne peut pas contenir un autre camion, alors qu'une liste de Military peut contenir des camions.

Ce n'est pas grave car c'est comme cela qu'a été défini un camion.