Exercice a faire :

On prend la classe qui est la

on s'arrange pour avoir la fonction frapper et on fait combattre les gladiateurs

<?php

class Personnage

{

private $\_force;

private $\_experience;

private $\_degats;

public function frapper(Personnage $persoAFrapper)

{

$persoAFrapper->\_degats += $this->\_force;

}

public function gagnerExperience()

{

$this->\_experience++;

}

// Mutateur chargé de modifier l'attribut $\_force.

public function setForce($force)

{

if (!is\_int($force)) // S'il ne s'agit pas d'un nombre entier.

{

trigger\_error('La force d\'un personnage doit être un nombre entier', E\_USER\_WARNING);

return;

}

if ($force > 100) // On vérifie bien qu'on ne souhaite pas assigner une valeur supérieure à 100.

{

trigger\_error('La force d\'un personnage ne peut dépasser 100', E\_USER\_WARNING);

return;

}

$this->\_force = $force;

}

// Mutateur chargé de modifier l'attribut $\_experience.

public function setExperience($experience)

{

if (!is\_int($experience)) // S'il ne s'agit pas d'un nombre entier.

{

trigger\_error('L\'expérience d\'un personnage doit être un nombre entier', E\_USER\_WARNING);

return;

}

if ($experience > 100) // On vérifie bien qu'on ne souhaite pas assigner une valeur supérieure à 100.

{

trigger\_error('L\'expérience d\'un personnage ne peut dépasser 100', E\_USER\_WARNING);

return;

}

$this->\_experience = $experience;

}

// Ceci est la méthode degats() : elle se charge de renvoyer le contenu de l'attribut $\_degats.

public function degats()

{

return $this->\_degats;

}

// Ceci est la méthode force() : elle se charge de renvoyer le contenu de l'attribut $\_force.

public function force()

{

return $this->\_force;

}

// Ceci est la méthode experience() : elle se charge de renvoyer le contenu de l'attribut $\_experience.

public function experience()

{

return $this->\_experience;

}

}