

Tous ces exercices prennent place au sein du même code. N'effacez pas et ne commentez pas votre code au fur et à mesure des exercices.

Exercice 1

A l'aide d'un objet Java, créer un personnage ayant pour nom "Jean Paul".

L'objet a trois attributs :

Paul a un nom : "Jean Paul"

Paul a 100 points de vie.

Paul 10 points de force.

Présentez au joueur Paul en l'affichant dans la console. La console affichera :

"Vous incarnez le héros suivant :"

(affichage de l'objet Paul)

Exercice 2

A l'aide d'un objet Java, créez une arme qui s'appelle "Épée rouillée".

L'arme a plusieurs attributs :

Elle a un nom : "Épée rouillée".

Elle a un nombre de dégâts : 10

Elle a un pourcentage de chance d'infliger des dégâts critiques : 20%

Exercice 3

A l'aide d'un objet Java, créez une armure qui s'appelle "Cartons scotchés".

L'armure a plusieurs attributs :

Elle a un nom : "Cartons scotchés"

Elle a un nombre de points de défense : 10

On équipe l'armure à Jean-Paul via un attribut "armor"

Exercice 4

Jean-Paul est équipé de l'épée. On va donc ajouter un attribut "equippedWeapon" à Jean-Paul, et cette arme sera l'épée rouillée.

Exercice 5

Un personnage peut frapper pour infliger des dégâts. La formule est la suivante :

Les chances de coup critique sont égales aux chances de critique de l'arme plus 1% par point de force du personnage.

On tire un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur aux chances de coup critique, alors c'est un coup critique.

Si c'est un coup critique, l'arme inflige le double de ses dégâts. Sinon, elle inflige simplement ses dégâts.

La force du personnage augmente les dégâts finaux de 10% par point de force du personnage.

Note : Pour avoir un nombre aléatoire en Java; on utilisera `Math.random()`

Créer une fonction qui prend en argument un personnage, et qui renvoie le nombre de dégâts infligés par le personnage.

En plus de renvoyer les dégâts infligés, la fonction affichera le message suivant :

<nom du personnage> utilise <nom de l'arme> et tente d'infliger <dégâts calculés> dégâts !

Exercice 6

Lorsqu'un personnage reçoit des dégâts, ces dégâts sont modifiés en fonction de l'armure.

Les dégâts modifiés sont ensuite soustraits aux points de vie du personnage.

La formule est la suivante : Chaque point de défense réduit les dégâts de 1%.
Donc, si le personnage a 25 points de défense, il reçoit 25% de dégâts en moins.

Créer une fonction qui fait prendre des dégâts à un personnage selon les dégâts potentiels reçus.

La fonction affichera le message suivant :

<nom du personnage> reçoit <dégâts calculés> ! Il lui reste <nombre de points de vies restants> points de vie !

Exercice 7

Créer un tableau de 2 éléments. Ce tableau va contenir 2 personnages qui serviront d'ennemis.

Ennemi 1 :

Nom : "Slim"

Points de vie : 25

Force : 3

Arme :

Nom : "Crachat"

Dégâts : 2

Chances de coup critique : 10%

Armure :

Nom : "Aucune"

Points de défense : 0

Ennemi 2 :

Nom : "Gobelin"

Points de vie : 30

Force : 6

Arme :

Nom : "Dague"

Dégâts : 10

Chances de coup critique : 5%

Armure :

Nom : "Cotte de maille"

Points de défense : 15

Exercice 8

Créer une fonction qui fait s'affronter deux personnages.

Au sein de la fonction, nous effectuerons une boucle qui s'arrêtera lorsque l'un des deux personnages sera mort (points de vies inférieurs ou égales à 0).

A chaque itération de la boucle, le premier personnage tape le second. Si le second personnage est en vie, il frappe le premier.

La fonction affichera au début, avant la boucle :

<Nom du premier personnage> affronte <nom du deuxième personnage> !

Quand un personnage gagne, la fonction affiche le nom du vainqueur.

Exercice 9

Terminer le logiciel pour que Jean-Paul affronte successivement les créatures du tableau, puis on affichera si Jean-Paul est encore en vie ou non.

Exercice 10

On veut désormais améliorer notre jeu pour pouvoir gérer un inventaire.

Les Armes et Armures pourront désormais être équipées et retirées, elles devront pouvoir aller dans l'inventaire lorsque retirées.

Chaque Arme et Armure possédera un poid et une "icône" pour la représenter. Pour l'instant l'icône sera une simple chaîne de caractères.

Exercice 11

Généralisation et Interfaces : on veut pouvoir utiliser des potions lors de notre combat. Ces potions devront pouvoir être stockées dans l'inventaire et utilisées lorsque nécessaire. Modifiez le code précédent pour créer un objet PotionSoin qui redonnera de la vie quand utilisé. Modifier Armes et Armures si nécessaire.

Exercice 12

Interactivité, maintenant que nous avons plus d'options pour notre combat, rendez ce dernier interactif via une interface simple. Vous êtes désormais Jean-Luc ! (d'ailleurs pourquoi ne pas demander le nom du joueur ?)

Combattez les monstres présents dans la liste d'ennemis, le logiciel devra demander quelle action sera réalisée ce tour par notre héro.