

Projet Conan le barbare

Le but de ce projet est de manipuler les attributs et les méthodes de plusieurs instances de classes différentes.

1 Notre héros

Définition de la **classe Barbare** :

L'instance de cette classe sera définie par :

- son nom (pseudo que l'utilisateur pourra choisir lors de la création de l'objet ou "Conan" par défaut)
- son initiative (constante) fixée à 12
- son nombre de points d'attaque maximale (constante) fixée à 7
- son nombre de points de vie initialisé à 120 (et qui aura tendance à diminuer)

Exercice n° 1

- Rédiger le constructeur de la classe Barbare.
- Rédiger les accesseurs aux différents attributs de la classe Barbare.
- Rédiger une méthode spéciale de représentation

```
1 # Dans l'éditeur PYTHON
2 >>> combattant=Barbare("Rendal")
3 >>> print(combattant)
4 'Rendal le Barbare (120 PV)'
```

- Rédiger une méthode **decrement_PV** qui prendra en paramètre une valeur (int). Cette méthode fera diminuer les points de vie de l'instance de la classe Barbare (le nombre de points de vie perdus sera égal à valeur)

2 Le méchant

Définition de la **classe Monstre** : L'instance de cette classe sera définie par :

- son initiative (aléatoire) qui sera comprise entre 5 et 20
- son arme (aléatoire) qui sera choisie dans la liste de classe :
listeArme=["couteau", "sabre", "marteau"]
- son nombre de points d'attaque maximale (constante) qui dépendra de la puissance associée à son arme. On utilisera alors le dictionnaire de classe :
puissanceArme={"couteau":3, "sabre":6, "marteau":9}
- son nombre de points de vie initialisé à 6 (et qui aura tendance à diminuer)

Exercice n° 2

- Rédiger le constructeur de la classe Monstre (tous les attributs seront définis comme privés).
- Rédiger les accesseurs aux différents attributs de la classe Monstre.
- Rédiger une méthode spéciale de représentation

```
1 # Dans l'éditeur PYTHON
2 >>> gnome=Monstre()
3 >>> print(gnome)
4 "un monstre arme d un couteau avec une initiative de 17"
```

- Rédiger une méthode **decrement_PV** qui prendra en paramètre une valeur (int). Cette méthode fera diminuer les points de vie de l'instance de la classe Monstre (le nombre de points de vie perdus sera égal à valeur)

3 L'affrontement

Définition de la **classe Affrontement** : L'instance de cette classe sera définie par :

- le niveau (str) : difficulté pouvant être choisie par l'utilisateur et fixée par défaut à "Easy")
- le nombre de combat (int) initialisé à 0
- le nombre de coups échangés (int) initialisé à 0

Exercice n° 3

- Rédiger le constructeur de la classe Affrontement (tous les attributs seront définis comme privés).
- Rédiger les accesseurs aux différents attributs de la classe Affrontement.
- Rédiger les méthodes **incr_nbCombat** et **incr_nbCoupEchange** qui permettront d'augmenter de 1 les attributs correspondants.

4 Méthodes de classe :

4.1 Combat à mort !

Lorsqu'un héros (de type Barbare) rencontre un méchant (de type Monstre), celui qui a la plus grande initiative attaque le premier (attaquant), portant à son adversaire (attaqué) une attaque aléatoire comprise entre 1 et le nombre de point d'attaque de l'attaquant, faisant ainsi perdre à l'attaqué le même nombre de points de vie. Tant qu'il n'y a pas de mort, l'attaquant et l'attaqué échange leur rôle. A la fin du combat, on proclame le vainqueur.

Exercice n° 4

Rédiger une méthode **combatAMort** qui prendra en paramètre **heros** (de type Barbare) et **mechant** (de type Monstre) qui simulera le combat dont le déroulement sera affiché de la sorte :

```
1 # Dans l'éditeur PYTHON
2 >>> combattant=Barbare("Rendal")
3 >>> gnome=Monstre()
4 >>> tournoi=Affrontement()
5 >>> tournoi.combatAMort(combattant,gnome)
6 Combat opposant Rendal le Barbare (120 PV) a
7 un monstre armee d un couteau avec une initiative de 9
8 Rendal le Barbare (120 PV) attaque le premier.
9 PAF
10 PAF
11 PAF
12 Arghhhhh...
13 Rendal le Barbare (119 PV), a remporte le combat a mort.
```

4.2 Combat à mort !

Comme les copains du méchant (eux aussi de type Monstre) n'ont pas apprécié de voir mourir leur ami, ils se jettent un par un dans la bataille jusqu'à ce que vengeance soit faite.

Exercice n° 5

Rédiger une méthode de classe **survival** qui prendra en paramètre le nom (str) du héros (qui sera de type Barbare) qui simulera la succession de combats qui seront affichés de la même manière que précédemment avec, en plus, la conclusion dans laquelle figure le nombre de combats et le nombre de coups échangés :

```

1 # Dans l'éditeur PYTHON
2 >>>guerre=Affrontement()
3 >>> guerre.survival("Gwen")
4 Combat opposant Gwen le Barbare (120 PV) a
5 un monstre arme d un couteau avec une initiative de 18
6 un monstre arme d un couteau avec une initiative de 18
7 attaque le premier.
8 PAF
9 PAF
10 PAF
11 PAF
12 PAF
13 PAF
14 Arghhhhh....
15 Gwen le Barbare (115 PV), a remporte le combat a mort.
16
17 Combat opposant Gwen le Barbare (115 PV) a
18 un monstre arme d un marteau avec une initiative de 6
19 Gwen le Barbare (115 PV) attaque le premier.
20 PAF
21 PAF
22 PAF
23 Arghhhhh....
24
25 /// Etc ///
26
27 Combat opposant Gwen le Barbare (5 PV) a
28 un monstre arme d un sabre avec une initiative de 12
29 un monstre arme d un sabre avec une initiative de 12
30 attaque le premier
31 PAF
32 Arghhhhh....
33 un monstre arme d un sabre avec une initiative de 12,
34 a remporte le combat a mort.
35
36 Avant de mourir, Gwen a remporte 20 combats dans une
37 lutte durant laquelle se sont echanges 71 coups...

```