

Interro Python

10 janvier 2020

EPSI

Promo 2022

3eme année

2 heures

Tout documents autorisés – Internet inclus.

Livrables :

- Envoyez vos réponses par mail à hugues.levasseur@arobe.fr
- Précisez vos nom & prénom
- Pour les questions 1 ~ 3 : Vous devez ne rendre le fichier vehicule.py modifié
- Pour la question 4 : Vous devez ne rendre le fichier circuit.py modifié
- Pour les questions 5~6 : Seul le résultat m'intéresse.

Les questions 1~4 sont liées.

Les questions 5~6 sont indépendantes

Question 1 – Attributs – 2 points

Dans le fichier vehicule.py vous trouverez une classe qui permet de créer ... des véhicules.

Modifiez la classe véhicule afin que :

- Nom vaille « No Name » par défaut
- Poids vaille 1000 par défaut
- Accélération vaille 25 par défaut
- Nom soit privé (créez la méthode getNom)
- Poids soit privé (créez la méthode getPoids)
- Accélération soit privé (créez la méthode getAcceleration)
- Vitesse soit privé (créez la méthode getVitesse)

Jeu d'essai :

```
>>> from vehicule import *
>>> foo = Vehicule()
>>> bar = Vehicule("BMW Série 2", 1350, 55)
>>> foo
No Name / 0 Km/h (Vehicule / 1000 Kg / +25)
>>> bar
BMW Série 2 / 0 Km/h (Vehicule / 1350 Kg / +55)
```

Question 2 – Méthodes - 2 points chaque

Écrivez les méthodes Accelerer et Freiner de la classe Vehicule.

Méthodes :

- Accelerer : Augmente vitesse de (acceleration - poids / 100)
- Freiner : met vitesse à 0 si < à 35. Divise vitesse par 2 sinon.

Attention : La vitesse doit rester un entier.

Jeu d'essai :

```
>>> from vehicule import *
>>> foo = Vehicule("BMW Série 2", 1350, 55)
>>> foo.Accelerer()
>>> foo
BMW Série 2 / 42 Km/h (Vehicule / 1350 Kg / +55)
>>> foo.Accelerer()
>>> foo
BMW Série 2 / 84 Km/h (Vehicule / 1350 Kg / +55)
>>> foo.Freiner()
>>> foo
BMW Série 2 / 42 Km/h (Vehicule / 1350 Kg / +55)
>>> foo.Freiner()
>>> foo
BMW Série 2 / 21 Km/h (Vehicule / 1350 Kg / +55)
>>> foo.Freiner()
>>> foo
BMW Série 2 / 0 Km/h (Vehicule / 1350 Kg / +55)
>>>
```

Question 3 – Héritage – 2 points chacune

Écrivez une classe VoitureEssence qui hérite de Véhicule avec 1 différence :

- Un attribut carburation qui vaut « Essence »

Écrivez une classe VoitureDiesel qui hérite de Véhicule avec 2 différences :

- Un attribut carburation qui vaut « Diesel »
- Une méthode Accelerer qui n'augmente la vitesse que de $(0.8 * acceleration) - (poids / 100)$

Démontrez qu'une Yaris pourrie vaut mieux qu'une BMW Série 2 toute neuve !

Jeu d'essai :

```
>>> from vehicule import VoitureDiesel, VoitureEssence
>>> foo = VoitureEssence("Yaris pourrie", 950, 42)
>>> bar = VoitureDiesel("BMW Série2", 1350, 55)
>>> foo.Accelerer()
>>> foo.Accelerer()
>>> bar.Accelerer()
>>> bar.Accelerer()
>>> foo
Yaris pourrie / 66 Km/h ( VoitureEssence / 950 Kg / +42)
>>> bar
BMW Série 2 / 62 Km/h ( VoitureDiesel / 1350 Kg / +55)
>>>
```

Question 4 – Algorithmie – 5 points

Course automobile (au tour par tour ...)

Vous trouverez dans circuit.py un script à compléter.

Principe :

Le joueur parcourt les 10 cases du circuit

A chaque tour, il choisit entre Accélérer, Freiner ... ou ne rien faire.

Le chiffre dans circuit c'est l'angle du virage (0 => Ligne droite, 90 => Virage à 90 °, ...)

Si la vitesse est plus grande que (200 – angle) => La voiture sort de piste.

Ex :

- Vitesse = 120, angle = 50 => Ok
- Vitesse = 120, angle = 90 => Sortie de piste.

Attention : Ne faire le test de « sortie de piste » que si on est dans un virage (on peut dépasser les 200 km/h en ligne droite)

A chaque tour, vous devez :

Ligne 1 : Afficher le N° du tour

Ligne 2 ~ 5 : Demander à l'utilisateur ce qu'il fait

Faire varier sa vitesse

Si il sort de route, arrêter la boucle

Ligne 6 : Afficher le statut de la voiture

Ligne 7 à 16 : Afficher le « plan » du circuit et sa progression (avec un <== à coté)

... et faire tout les calculs nécessaires au bon fonctionnement de l'algorithme

Exemple d'affichage (pour un tour)

```
01 ===== Tour N° 3 ===
02 1 : Accélérer
03 2 : Continuer
04 3 : Freiner
05 Votre choix ? :3
06 Batmobile / 600 Km/h ( VoitureEssence / 3000 Kg / +100)
07 Ligne Droite .
08 Ligne Droite .
09 Ligne Droite <===
```

- 10 Virage 50° .
- 11 Ligne Droite .
- 12 Ligne Droite .
- 13 Virage 90° .
- 14 Ligne Droite .
- 15 Virage 20° .
- 16 Virage 120° .

PS : Score à battre : 4,15 secondes ;-)

Question 5 – Calcul – 2 points

Trouvez les 10 derniers chiffres de la suite : $1^1 + 2^2 + 3^3 + 4^4 + 5^5 + \dots + 1000^{1000}$

Question 6 – Fibonacci – 3 points

La [suite de Fibonacci](#) est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent.

Sauf pour les 2 premiers termes qui valent 1

- $F_1 = 1$
- $F_2 = 1$
- $F_3 = F_2 + F_1 = 2$
- $F_4 = F_3 + F_2 = 3$
- $F_5 = 5$
- $F_6 = 8$
- $F_7 = 13$
- $F_8 = 21$
- $F_9 = 34$
- $F_{10} = 55$
- $F_{11} = 89$
- $F_{12} = 144$
- ...

Le 12eme terme (F_{12}) est le premier terme à avoir 3 chiffres.

Quel est le rang du 1^{er} terme de la suite de Fibonacci à avoir 1000 chiffres ?