

Exercice 3 - Programmation objet

Un chronomètre est un instrument qui permet de mesurer le temps qui s'écoule entre deux instants, celui où le bouton 'run' est pressé la première fois et celui où le même bouton est pressé la seconde fois. En outre un bouton 'reset' permet de réinitialiser le chronomètre, c'est à dire d'effacer les mesures enregistrées. Enfin, un bouton permet d'enregistrer des temps intermédiaires entre le temps enregistré au début et la fin d'une mesure.

Un informaticien veut implémenter en langage Python et programmation objet une classe 'Krono' qui permettra d'instancier des chronomètres. Il utilise la fonction 'time_ns' du module 'time'. Cette fonction renvoie un nombre entier qui correspond au nombre de nanosecondes (ns) qui s'est écoulé depuis le 1^{er} janvier 1970. On rappelle que 1 seconde correspond à 10^9 ns.

Il commence par écrire ceci :

```
1 from time import time_ns
2
3 class Krono:
4
5     def __init__(self):
6         self.timeStart = 0
7         self.timeEnd = 0
8         self.timeLapse = []
```

Question 1a - En programmation objet et d'une façon générale quel nom donne-t-on à un bloc d'instructions déclaré par le mot-clé 'def' ?

Question 1b - En programmation objet et d'une façon générale quel nom donne-t-on au bloc d'instructions déclaré par 'def, __init__(self)' ?

Question 1c - En programmation objet et d'une façon générale quel nom donne-t-on aux 'variables' contenues dans le bloc d'instructions déclaré par 'def, __init__(self)' ?

Il veut écrire une méthode appelée 'isRun' qui renvoie l'état d'activité d'un objet de la classe 'Krono'.

Un chronomètre est considéré comme arrêté - la méthode renvoie alors 'False' - si la valeur des deux attributs 'self.timeStart' et 'self.timeEnd' est égale à zéro ou si la valeur de ces deux attributs est différente de 'zéro'. Un chronomètre est considéré comme 'en marche' - la méthode renvoie alors 'True' - si la valeur de 'self.timeStart' est différente de '0' et celle de 'self.timeEnd' est égale à '0'.

Question 2 - Écrire la méthode 'isRun'.

Il veut maintenant écrire une méthode 'run' qui déclenche la mise en marche d'un chronomètre arrêté, et provoque l'arrêt d'un chronomètre en activité. Lors de la mise en marche la valeur renvoyée par la fonction 'time_ns' est affectée à 'self.timeStart' et lors de son arrêt elle est affectée à 'self.timeEnd'.

Question 3a - Écrire la méthode 'run'.

Il veut ensuite écrire une méthode 'lapse' qui ajoute à l'attribut 'self.timeLapse' la valeur renvoyée par la fonction 'time_ns' lorsque le chronomètre est en marche. Ceci permet de conserver des temps intermédiaires.

Question 3b - Écrire la méthode 'lapse'.

Il veut ensuite écrire une méthode 'reset' qui permet de réinitialiser les valeurs des attributs d'un chronomètre lorsqu'il est à l'arrêt.

Question 3c - Écrire la méthode 'reset'.

Pour finir il écrit une méthode 'view' qui renvoie une chaîne de caractère sous la forme : 'hh:mm:ss.ms' correspondant au temps mesuré par un chronomètre exprimé en heure (h), minute (m), seconde (s) et milliseconde (ms). Par exemple la méthode renvoie :

'00:01:34.810\n' pour un chronométrage effectué sans prise de temps intermédiaire et

'00:14:12.761\nTemps intermédiaires\n00:05:03.840\n00:09:08.941\n' pour un chronométrage effectué avec deux prise de temps intermédiaire. On rappelle que la chaîne '\n' permet de faire un 'retour à la ligne'.

Il écrit maintenant un programme principal :

```
1  from class_Krono import *
2
3  c = Krono()
4  runK = True
5
6  while runK is True:
7      action = input("S : Start and Stop, L : Lapse >>> ")
8      action = action.upper()
9
10     if action == 'L':
11         c.lapse()
12
13     elif action == 'S':
14         c.run()
15
16
17
18
19     else:
20         continue
```

Question 4a - Quel nom donne-t-on d'une façon générale en programmation objet à l'opération réalisée en ligne 3 ?

Question 4b - Quelles instructions faut-il ajouter aux lignes 15, 16 et 17 pour afficher le temps mesuré par le chronomètre 'c' dès qu'il s'arrêtera d'une part, et sortir de la boucle d'autre part ?

Il voudrait pouvoir déclencher un chronométrage avec deux ou plusieurs chronomètres lancés au même instant.

Question 5 - Proposer une solution à ce problème par l'ajout d'une ou plusieurs méthodes dans la classe 'Krono'.