## Exercice 3 – Programmation objet

**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' ?

Il s'agit d'une **méthode**.(0,5)

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

Il s'agit du **constructeur**.(0,5)

**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)' ?

Ce sont des **attributs**.(0,5)

**Question 2** – Écrire la méthode 'isRun'.

```
 9
10      def isRun(self):
11          return (self.timeStart != 0 and self.timeEnd == 0)
12
```

**Question 3a** – Écrire la méthode 'run'.

```
20
21      def run(self):
22          if self.isRun():
23              self.timeEnd = time_ns() // 1000000
24          else:
25              self.timeStart = time_ns() // 1000000
26          return True
27
```

**Question 3b** – Écrire la méthode 'lapse'.

```
27
28      def lapse(self):
29          if self.isRun():
30              self.timeLapse.append(time_ns() // 1000000)
31              return True
32          return False
33
```

**Question 3c** – Écrire la méthode 'reset'.

```
12
13      def reset(self):
14          if self.isRun() is False:
15              self.timeStart = 0
16              self.timeEnd = 0
17              self.timeLapse = []
18              return True
19          return False
20
```

***Remarque 1***

*Les passages surlignés en jaune sont facultatifs. Leur absence dans la réponse n'entraîne aucune pénalité.*

***Remarques 2***

*Pour la définition des méthodes, l'oubli de 'self' en argument n'est pénalisé qu'une seule fois ;*

*Pour l'appel à une méthode, l'oubli de 'self.' avant le nom de la méthode n'est pénalisé qu'une seule fois ;*

*Pour l'appel à une méthode, l'oubli des parenthèses après le nom de la méthode n'est pénalisé qu'une seule fois ;*

*Pour l'appel à la fonction 'time_ns', l'oubli des parenthèses après le nom de la fonction n'est pénalisé qu'une seule fois ;*

*L'utilisation d'un seul signe '=' au lieu de deux pour l'opérateur de comparaison d'égalité est considéré comme une erreur mineure ;*

*L'utilisation du signe '+' au lieu de la méthode .append pour l'ajout d'une valeur à une liste est considéré comme une erreur majeure.*

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

Il s'agit de la **création** d'un **objet**, d'une **instance** de la classe 'Krono'. Cette opération est donc une **instanciation**.

**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 ?

```
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          if c.isRun() is False:
16              runK = False
17              print(c.view())
18
19      else:
20          continue
21
```

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

*On valorisera toute proposition de solution indiquant l'affectation d'une même valeur de `time_ns` à plusieurs instances de la classe `Krono`.*

```
33
34      def startPlus(self, tk):
35          """Déclenche la mise en marche simultanée de plusieurs chronomètres.
36
37          Parameters
38          ----------
39          tk : TUPLE
40              objets de la classe 'Krono'.
41
42          Returns
43          -------
44          None.
45
46          """
47          if self.isRun() is False :
48              self.timeStart = time_ns()
49              for k in tk:
50                  if k.isRun() is False :
51                      k.timeStart = self.timeStart
52
```

```
In [4]: c1, c2, c3, c4 = Krono(), Krono(), Krono(), Krono()
   ...: c1.startPlus((c2, c3, c4))
   ...: for c in (c1, c2, c3, c4):
   ...:     print(c.timeStart)
1668529984645965800
1668529984645965800
1668529984645965800
1668529984645965800
```