

Exercice 1

Questions à réponses courtes

1.1 Quelle est la différence entre une classe et une instance de classe ?

1.2 Définir ce qu'est un attribut de classe et une méthode de classe.

1.3 Quelle fonction est appelée lorsque l'on fait une *instanciation* ?

1.4 Pour créer la classe fruit, vous écrivez :

- `class Fruit() :`
- `class Fruit :`
- `def Fruit() :`

1.5 Pour créer un objet *banane*, vous écrivez :

- `banane = Fruit`
- `banane = Fruit()`
- `Fruit(banane)`

1.6 Quel constructeur est correct ?

```
1 # a
2 def init(self):
3     x = 0
4 # b
5 def __init__(self):
6     x = 0
7 # c
8 def __init__(self):
9     self.x = 0
```

1.7 classe Personne

```
1 class Personne:
2     def __init__(self, nom, année_naissance, lieu_naissance):
3         self.nom = nom
4         self.année_naissance = année_naissance
5         self.lieu_naissance = lieu_naissance
6
7     def age(date)
8         return date - année_naissance
```

1. Corriger la méthode de classe `age`

2. On instancie un nouvel objet nommé *marc* de la manière suivante : `marc = Personne(self, 'Marc Assin', 1989, 'Bergerac')` . Quel type d'erreur cela retourne t-il ?

Exercice 2

Point, Segment, Polygone

2.1 Classe Point

En géométrie, un point est constitué de 2 coordonnées, x et y . Vous allez définir complètement la classe `Point` à partir des renseignements suivants :

2.1.1 instantiation

L'instanciation se fera avec

```
1 def __init__(self, coord_x, coord_y):
2     self.x = ...
3     self.y = ...
```

2.1.2 Méthode pour lire la valeur des coordonnées d'un point

```
1 def get_coordo(self):
2     return ...
```

2.1.3 Méthode pour traduire

Le point doit posséder une méthode qui calcule ses nouvelles coordonnées à partir d'une translation dans la direction vx, vy .

```
1 def traduire(self, ...
2     self.x = ...
3     ...
```

Questions :

1. Compléter les scripts de ces méthodes
2. Préciser s'il s'agit de SETTER / GETTER pour chacune
3. Ecrire les instructions du programme qui devra :
 - créer les points $p1 = (0, 0)$ et $p2 = (3, 4)$
 - faire une translation sur chacun des points avec le vecteur $(vx, vy) = (2, 2)$
 - obtenir les valeurs des nouvelles coordonnées

2.2 Classe Segment

La classe `Segment` va traiter les données issues de 2 Points

2.2.1 instantiation

L'instanciation se fera avec

```
1 def __init__(self, point1, point2):
2     self.p1 = ...
3     self.p2 = ...
```

2.2.2 Méthode de tracé d'un segment : trace

On utilisera les méthodes du module Turtle :

```
1 import Turtle
2 tt = Turtle() # instanciation de l'objet tt à partir de la classe
   Turtle
3 tt.up() # lever le crayon
4 tt.goto(x,y) # se deplace à la position x,y
5 tt.down() # baisser le crayon
```

2.2.3 Méthode de calcul de distance : fonction distance

Ajouter une méthode de calcul de distance à partir de l'expression mathématique :

$$d = \text{math.sqrt}((p2.x - p1.x)^2 + (p2.y - p1.y)^2)$$

Questions :

1. Ecrire les méthodes de tracé et de calcul de distance de la classe Segment
2. Ecrire les instructions qui :
 - crée l'objet segment1 à partir des points p1 et p2
 - trace le segment entre ces 2 points
 - calcule la longueur du segment

2.3 Classe Polygone

1. Créer la classe Polygone
2. Implémenter une méthode qui trace le polygone
3. Implémenter une méthode qui fait une translation du polygone puis trace celui-ci
4. Ecrire les instructions qui :
 - créent un carré de coordonnées (0,0), (2,0), (2,2), (0,2)
 - trace le polygone
 - fait une translation de vecteur $(vx, vy) = (2, 2)$
 - trace le polygone