

# Bases de la programmation impérative

Ensimag 1A — feuille 2

## 1. Points

On considère le programme suivant :

```
1 x1, y1 = 3, 5
2 x2, y2 = 4, 4
3
4 milieu_x, milieu_y = (x1+x2)/2, (y1+y2)/2
5 print("le milieu est (", milieu_x, milieu_y, ")")
6 print("on projette sur la droite y=5")
7 milieu_y = 5
8 print("nous sommes maintenant (", milieu_x, milieu_y, ")")
```

1.1. Réécrire une classe *Point* utilisant deux attributs  $x$  et  $y$  (comme vu en cours). Cette classe devra fournir :

- un constructeur de *Point* prenant deux coordonnées en paramètre ;
- un opérateur '+' qui retourne un nouveau *Point* dont les coordonnées sont la somme des coordonnées des opérandes ;
- un opérateur '/' qui retourne un nouveau *Point* dont les coordonnées sont les coordonnées de l'opérande de gauche divisées par ce scalaire ;
- un opérateur de conversion d'un point en chaîne de caractères, pour faciliter l'affichage.

1.2. Réécrire le code à l'aide de la nouvelle classe et des opérateurs implémentés.

## 2. Tableaux

Voici un petit programme sur des tableaux :

```
1 tab = [7, 2, 5, 9]
2 tab[3] = tab[2] + tab[1]
3 tab[2] = tab[1] + tab[3]
4 tab[4] = tab[1] + tab[2]
5
6 print(tab)
```

2.1. Qu'affiche-t-il ?

### 3. Triangles

- 3.1. Proposez une classe *Triangle* utilisant un tableau de trois points.
- 3.2. Dessiner la zone mémoire correspondant à une variable triangle. Quel est le nom utilisé pour désigner la coordonnée  $x$  du premier point d'un triangle  $t$  ?
- 3.3. Écrire une méthode renvoyant le barycentre du triangle.
- 3.4. Que penser du code suivant ?

```
1 p1, p2, p3 = Point(0, 0), Point(3, 0), Point(0, 3)
2 p4, p5, p6 = Point(0, 0), Point(4, 0), Point(0, 4)
3 t1 = Triangle(p1, p2, p3)
4 t2 = Triangle(p4, p5, p6)
5 print(t1, t2)
6 t1.points[0].x = 5
7 print(t1, t2)
8 t1.points[0] = p4
9 print(t1, t2)
10 t1.points[0].x = 5
11 print(t1, t2)
```