

Chapitre 8 - Programmation Orientée Objet (POO)

La Programmation Orientée Objet (**POO**) est un paradigme de programmation qui utilise des "objets" pour représenter des données et des méthodes. Un **objet** est une **instance** d'une **classe**, qui peut être considérée comme un modèle ou un plan définissant les attributs et les comportements de l'objet.

1- EXEMPLE : POO POUR DEFINIR DES CERCLES

```
class Cercle :
    contenu = {}
    nb = 0

    def __init__(self,x = 250,y = 250 ,r = 0 , nom = "", couleur= "white") :
        self.centre = [x,y]
        self.r = r
        self.nom = nom
        self.couleur = couleur
        Cercle.nb = Cercle.nb + 1
        self.insererDansCanvas()

    def insererDansCanvas(self):
        xA = self.centre[0] - self.r
        yA = self.centre[1] - self.r
        xB = self.centre[0] + self.r
        yB = self.centre[1] + self.r
        num = canvas.create_oval(xA,yA,xB,yB,fill=self.couleur)
        Cercle.contenu[self.nom] = [self,num]

    def perimetre(self) :
        p = self.r * 2 * pi
        return p

    def surface(self) :
        s = pi * self.r ** 2
        return s

    def agrandir(self,k) :
        self.r = self.r * k

    def translater(self,X,Y) :
        self.centre[0] = self.centre[0] + X
        self.centre[1] = self.centre[1] + Y

    def __str__(self) :
        p = self.perimetre()
        s = self.surface()
        return f"""
Cercle {self.nom}
Centre ({self.centre[0]},{self.centre[1]})
Rayon = {self.r}
Couleur = {self.couleur}
Périmètre = {p}
Surface = {s}
"""
```

```
from math import pi
from tkinter import Tk , Canvas
```

```
def fct(event) :
    s = input("Nouvel objet (x,y,r,nom,couleur) : ")
    s = s.split(',')
    c = Cercle(int(s[0]),int(s[1]),int(s[2]),s[3],s[4])
```

```
# Main
fenetre = Tk()
canvas = Canvas(fenetre,width=500, height=500,bg='black')
canvas.grid()

A = Cercle(100,200,50,"A", "#f00")
A.translater(10,20)
A.agrandir(5)
B = Cercle(x=300,nom="B",r=50,y=100)

canvas.bind("<ButtonPress-1>",fct)
fenetre.mainloop()
```

Exécution des lignes ci-dessous	Résultat dans le shell
<pre>A = Cercle(10,20,50,"A") print(A)</pre>	
<pre>print(A.r)</pre>	
<pre>print(A.centre)</pre>	
<pre>A.translater(10,20) A.agrandir(5) print(A)</pre>	
<pre>B = Cercle(x=40,nom="B",r=1,y=5) print(B)</pre>	

<code>print(Cercle.nb)</code>	
<code>print(A.nb)</code>	

2- EXERCICE : POO POUR DEFINIR DES RECTANGLES

Ecrire dans un fichier nommé *rectangle.py*, le script python de la classe Rectangle qui permet en exécutant les lignes du programme principal ci-dessous, de retrouver dans la console, le résultat de la figure donnée ci-contre :

```
r1 = Rectangle(100,200,50,80,"rectangle 1","red")
print(r1)
r1.translater(10,20)
r1.agrandir(2)
print(r1)
r2 = Rectangle(x=300,nom="rectangle 2",l=100,y=300,h=100)
print(r2)
```

```
>>> (executing file "rectangle.py")
```

```
Rectangle rectangle 1
  Centré sur le point (100,200)
  Longueur = 50
  Largeur = 80
  Couleur = red
  Perimetre = 260
  Surface = 4000
  | | |
```

```
Rectangle rectangle 1
  Centré sur le point (110,220)
  Longueur = 100
  Largeur = 160
  Couleur = red
  Perimetre = 520
  Surface = 16000
  | | |
```

```
Rectangle rectangle 2
  Centré sur le point (300,300)
  Longueur = 100
  Largeur = 100
  Couleur = white
  Perimetre = 400
  Surface = 10000
  | | |
```