

TD 2 – Concepts de la programmation orientée objet (POO)

Objectifs : Rappels python, notions d'objet, classe, attributs, méthode, introduction au diagramme de classes et diagramme de séquence

1. Passage du Python au Python Orienté Objet

Description de l'exercice : Vous disposez du code Python ci-dessous qui calcule et affiche l'aire et le périmètre d'un cercle en utilisant des fonctions. Vous aurez à transformer ce code en une structure orientée objet en utilisant des classes et des méthodes.

```
import math

def calculer_perimetre(rayon):
    return 2 * math.pi * rayon

def calculer_aire(rayon):
    return math.pi * rayon ** 2

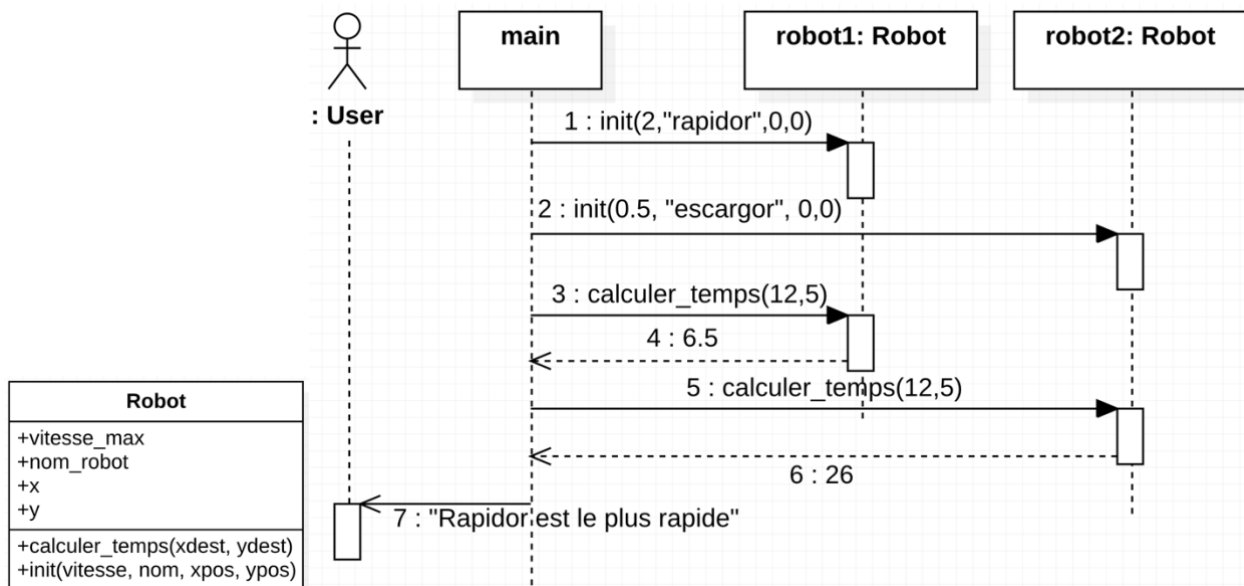
#Programme principal
rayon = float(input("Entrez le rayon du cercle: "))
perimetre = calculer_perimetre(rayon)
aire = calculer_aire(rayon)
print(f"Rayon du Cercle: {rayon}")
print(f"Périmètre du Cercle: {perimetre}")
print(f"Aire du Cercle: {aire}")
```

2. Donnez le diagramme de classes et le code de la classe Cercle, possédant un attribut rayon, et des méthodes pour calculer l'aire et le périmètre, ainsi qu'une méthode pour afficher les informations du cercle.
3. Donnez le code python du programme principal qui crée une instance de la classe Cercle avec un rayon donné, et calcule les valeurs du périmètre et de l'aire et affiche les résultats.

2. Transformer diagrammes de classes et séquence en python

Description de l'exercice : Vous disposez des diagrammes de classes et de séquences ci-dessous. En vous appuyant sur ces diagrammes répondez aux questions ci-dessous :

1. Donnez le code python de la classe Robot, comportant notamment quatre attributs : la vitesse_max en m/s, nom_robot qui est le nom donné au robot, ainsi que les positions en x et y (en mètres par rapport à l'origine 0, 0 dans un plan).
2. Donnez le code python du programme principal qui crée 2 robots, et appelle la méthode calculer_temps pour chacun d'eux, et conclut sur qui met le moins de temps.



3. Traduire code python en diagramme de classes et de séquence

Description de l'exercice : Vous disposez du code python ci-dessous. Faire le diagramme de classes et de séquence.

```

class Vehicule:
    def __init__(self, marque, modele):
        self.marque = marque
        self.modele = modele

class Conducteur:
    def __init__(self, nom):
        self.nom = nom

class Location:
    def __init__(self, vehicule, conducteur, jours):
        self.vehicule = vehicule
        self.conducteur = conducteur
        self.jours = jours

class Agence:
    def __init__(self, nom):
        self.nom = nom
        self.locations = []

    def louer_vehicule(self, vehicule, conducteur, jours):
        location = Location(vehicule, conducteur, jours)
        self.locations.append(location)

    def afficher_locations(self):
        print(f"Locations à l'agence {self.nom}:")
        for location in self.locations:
            print(f"{location.conducteur.nom} a loué un véhicule {location.vehicule.marque} {location.vehicule.modele} pour {location.jours} jours")
  
```

```
# Programme Principal
agence = Agence("Mon Agence")
conducteur1 = Conducteur("Alice")
vehicule1 = Vehicule("Toyota", "Camry")
agence.louer_vehicule(vehicule1, conducteur1, 3)
agence.afficher_locations()
```