

STRUCTURES DE DONNEES – DEVOIR

EXERCICE 1: LA POO EN QUELQUES QUESTIONS

1) On définit la classe `Identite` de la manière suivante:

```
1 class Identite:
2     def __init__(self, nom, prenom, an):
3         self.nom=nom
4         self.prenom=prenom
5         self.an=an
6     def age(self, a):
7         return a-self.an
```

Qu'affichera la console à la suite de l'exécution des instructions suivantes? Justifiez toutes vos réponses.

- a) `maria=Identite('Maria', 'Pires', 1999)`
`print(maria.nom + ' ' + maria.prenom)`
- b) `maria=Identite('Maria', 'Pires', 1999)`
`print(maria.nom + ' ' + maria.an)`
- c) `hugo=Identite('Hugo', 'Dupont', 2004)`
`print(hugo.age(2021))`

2) Ecrire le code de la classe `Voiture` qui permet d'afficher *'Ferrari Rouge'* après avoir saisi les instructions suivantes:

```
>>> F430=Voiture('Ferrari', 'rouge')
>>> print(F430)
Ferrari rouge
```

3) Expliquez en détail ce que permet d'afficher ce programme:

```
import random

class Piece:
    def alea(self):
        return random.randint(0,1)
    def moyenne(self,n):
        tirage=[]
        for i in range (n):
            tirage.append(self.alea())
        return sum(tirage)/n
p=Piece()
print(p.moyenne(100))
```

Rappel: la fonction `sum` calcule la somme des termes d'une liste.

EXERCICE 2: LA CLASSE `Date`

On construit le code d'une variable globale et d'une classe `Date` représenté ci-dessous:

```
liste = ["janvier", "février", "mars", "avril", "mai", "juin", "juillet", "aout",
"septembre", "octobre", "novembre", "décembre"]
class Date:
    def __init__(self, jour, mois, annee):
        self.jour = jour
        self.mois = mois
        self.annee = annee
    def __lt__(self, d):
        if self.annee < d.annee:
            return True
        elif self.annee > d.annee:
            return False
        else:
            if self.mois < d.mois:
                return True
            elif self.mois > d.mois:
                return False
            else:
                return self.jour < d.jour
```

La méthode `__lt__` est la méthode spéciale de comparaison ('lower than').

1) Qu'affichera la console à l'exécution des instructions suivantes? Justifiez votre réponse.

```
d1 = Date(17, 12, 2021)
d2 = Date(11, 12, 2021)
print(d1<d2)
```

2) Ecrire le code d'une méthode de la classe `Date` qui permet d'afficher '17décembre2021' après avoir saisi les instructions suivantes:

```
>>> d1 = Date(17, 12, 2021)
>>> print(d1)
17décembre2021
```