
Devoir de contrôle SEMESTRE 1

Matière : INFORMATIQUE

Classes : MP2, PC2 et PT2

Durée : 1h

L'indentation dans les instructions Python, la clarté de la copie seront prises en considération dans le barème.

L'objectif du problème est de créer une application qui permet de gérer les informations d'une population.

Afin de réaliser cette application, nous devons définir une classe **Adresse**, une classe **Personne** et une classe **Population**.

A. La classe Adresse

La classe **Adresse** permet d'instancier des objets représentant les différentes adresses d'une population.

Le constructeur de cette classe initialise les attributs suivants :

- « ville » de type chaîne de caractères
- « rue » de type chaîne de caractères
- « codePostal » de type entier désignant le code postal formé de quatre chiffres

Définir une classe **Adresse** contenant les méthodes suivantes :

1. un constructeur `__init__` qui initialise les attributs d'objet ville, rue et codePostal.

NB. Si le code postal donné comme paramètre n'est pas formé de quatre chiffres, cette méthode doit lever une exception et initialiser le code postal à 1111.

2. une méthode `__str__` qui retourne une chaîne de caractères contenant les différentes informations d'une adresse.

3. une méthode `__eq__` qui prend en paramètre deux objets de la classe Adresse et retourne **True** si les valeurs de leurs attributs sont égales, sinon **False**.

Exemple

```
>>> a1=Adresse("Ville 1", "Rue 1", 3011)
```

```
>>> print(a1)
```

```
Rue 1,3011-Ville 1
```

```
>>> a2=Adresse("Ville 2", "Rue 2", 30125)
```

```
le code postal doit être composé de quatre chiffres
```

```
>>> print(a2)
```

```
Rue 2,1111-Ville 2
```

```
>>> print(a1==a2)
```

```
False
```

B. La classe Personne

La classe **Personne** permet d'instancier des objets représentant une population.

Le constructeur de cette classe initialise les attributs suivants :

- « nom » de type chaîne de caractères
- « prenom » de type chaîne de caractères
- « adresse » une instance de la classe Adresse

Définir une classe **Personne** contenant les méthodes suivantes :

4. un constructeur `__init__` qui initialise les attributs d'objet nom, prenom et adresse.
5. une méthode `__str__` qui retourne une chaîne de caractères contenant les différentes informations d'une personne.

Exemple

```
>>> pr=Personne("Adams", "Jule", a1)
```

```
>>> print(pr)
```

```
Adams ; Jule ; Rue 1,3011-Ville 1
```

6. une méthode `__eq__` qui prend en paramètre deux objets de la classe **Personne** et retourne **True** si les valeurs de leurs attributs sont égales, sinon **False**.
7. une méthode **demenager** qui permet de modifier l'adresse d'une personne par une adresse passée en paramètre. Cette méthode lève une exception si le paramètre donné n'est pas une adresse valide.

Indication :

`isinstance(objet,classe)` retourne **True** si *objet* est une instance de la classe *classe* , **False** sinon.

Exemple

```
>>> pr.demenager("Ville 4")
```

```
vous devez saisir une adresse complète
```

```
>>> pr.demenager(Adresse("Ville 4","Rue 1",9012))
```

```
>>> print(pr)
```

```
Adams ; Jule ; Rue 1,9012-Ville 4
```

C. La classe Population

Définir la classe **Population** qui possède une liste de personnes comme attribut et les méthodes suivantes :

8. un constructeur `__init__` qui permet d'initialiser l'attribut **personnes** à la liste vide.
9. une méthode `__contains__` qui prend comme paramètre *p* un objet de la classe **Personne** et retourne **True** si *p* est une personne de la population, **False** sinon.

Exemple

```
>>> P=Population()
```

```
>>> print(pr in P)
```

```
False
```

10. une méthode **ajouterPersonne** qui prend en paramètre *p* un objet de la classe **Personne** et l'ajoute à la population s'il n'est pas encore ajouté, sinon un message "la personne existe déjà dans la population" sera affiché.

Exemple

```
>>> P.ajouterPersonne(pr)
```

```
>>> a3=Adresse("Ville 3", "Rue 3", 3013)
```

```
>>> P.ajouterPersonne(pr)
```

```
la personne existe déjà dans la population
```

```
>>> P.ajouterPersonne(Personne("Daniel", "Lili", a2))
```

```
>>> P.ajouterPersonne(Personne("Smith", "Jane", a1))
```

```
>>> P.ajouterPersonne(Personne("Doe", "Brown", a3))
```

```
>>> P.ajouterPersonne(Personne("Doe", "John", a1))
```

11. une méthode **voisin** qui prend en paramètre une adresse *a1* objet de la classe **Adresse**, retourne une liste de tuples contenant les noms et les prénoms des personnes ayant cette adresse.

Exemple

```
>>> P.voisin(a1)
```

```
[('Smith', 'Jane'), ('Doe', 'Brown')]
```

12. une méthode **sauvegarder_vers_fichier** qui permet de sauvegarder la population dans un fichier intitulé "population.txt"

Voici un extrait du fichier "population.txt" qui sera créé lors de l'appel de cette méthode

