

Atelier Programmation Python

Atelier 2 : L'orienté objet avec Python

Exercice 1 : POO, les bases.

Écrire une classe Python nommée `Personne` et qui possède trois attributs définissant certaines caractéristiques d'une personne réelle : **taille** (en m), **poids** (en kg) et **age**.

Cette classe Python aura deux méthodes :

- **calculIMC(self)** qui détermine l'IMC de la personne.
- **interpretationIMC(self)** qui affiche :

"Le résultat du calcul de l'IMC indique que la personne est en insuffisance pondérale (maigreur)" si le résultat retourné par **calculIMC(self)** est inférieur ou égal à 18.5.

"La personne est obèse" si le résultat est supérieur ou égal à 30.

"La personne est en surpoids ou corpulence normale" sinon.

Rappel : L'indice de masse corporelle (IMC) est donné par la formule « $\text{poids}/(\text{taille}^2)$ »

Tests de vérification :

```
>> julien = Personne (1.87, 95, 26)
```

```
> julien.calculIMC(self)
```

```
27.16
```

```
>> julien.interpretationIMC(self)
```

```
"La personne est en surpoids ou corpulence normale"
```

Exercice 2 : Implémenter les concepts de la Programmation Orientée Objet.

Les classes sont souvent utilisées pour modéliser des objets dans le monde réel. Nous pouvons représenter les données sur une personne dans un programme par une classe **Personne**, contenant le *nom de la personne, son prénom, son numéro de téléphone*, et *son email*. Une méthode `__str__` peut imprimer les données de la personne.

Un travailleur est une personne ayant un emploi. Dans un programme, un travailleur est naturellement représenté comme une classe **Travailleur** dérivée de la classe **Personne**, parce qu'un travailleur est une personne, c'est-à-dire, nous avons une relation **est-un**.

La classe **Travailleur** étend la classe **Personne** avec des données supplémentaires, par exemple le *nom de l'entreprise, l'adresse de l'entreprise* et le *numéro de téléphone* du travail. La fonctionnalité d'impression (la méthode spéciale `__str__`) doit être modifiée en conséquence.

Travail à faire : Mettre en œuvre cette classe Travailleur.

Un scientifique est un type spécial de travailleur. La classe Scientifique peut donc être dérivée de la classe **Travailleur**. Ajouter des données sur la discipline scientifique (physique, chimie, mathématiques, informatique, ...).

Travail à faire : Mettre en oeuvre la classe Scientifique.

Enfin, faites un programme principal de démonstration où vous créez et imprimez des instances de classes **Personne**, **Travailleur** et **Scientifique**. Imprimez le contenu des attributs de chaque instance.

Vous allez maintenant construire un carnet d'adresses en utilisant la classe **Personne** définie dans le l'exercice précédent afin d'enregistrer vos contacts (amis, famille, ...).

Chaque entrée du carnet d'adresses sera une instance de la classe **Personne**.

Le carnet d'adresses doit vous permettre de chercher et retourner les informations de vos contacts. **Travail à faire :**

Créer une classe **CarnetAdresses** contenant les méthodes suivantes :

1. La méthode constructeur **__init__**
2. La méthode d'impression **__str__**
3. Une méthode nommée **ajouter_contact** qui vous permet d'ajouter une personne à votre carnet d'adresses.
4. Une méthode **chercher_contact** qui recherche un contact par son nom parmi les personnes enregistrées dans votre carnet d'adresses et affiche dans une nouvelle ligne chaque contact possédant le nom en question.

Cette méthode doit accepter deux arguments :

- Un argument obligatoire : le nom du contact à rechercher.
- Un deuxième argument optionnel : le prénom du contact, permettant de réduire le résultat si plusieurs contacts ont le même nom.