LS4 - Travaux pratiques 6

12 mars 2014

Objectifs : introduire les objets.

On considère le programme objet.py suivant :

```
import math
class Point():
    """Une classe Point."""
    nombre_de_points = 0
    def __init__(self,x=0,y=0):
        """ initialise un Point """
        self.x = x
        self.y = y
        Point.nombre_de_points += 1
    def __del__(self):
        print("Le point",self.x,self.y,"est détruit.")
        Point.nombre_de_points += -1
    def distancenull(self):
        return math.sqrt(pow(self.x,2)+pow(self.y,2))
    def distance(self,p):
        return math.sqrt(pow(self.x-p.x,2)+pow(self.y-p.y,2))
```

On exécute successivement les bouts de code suivants.

Que produit leur évaluation ?

```
from objet import *
p = Point(3,4)
print(p.__doc__)
```

q1 :

- ☐ Elle affiche la chaîne "Une classe Point." sur la sortie standard.
- ☐ Elle initialise une instance de classe Point.
- ☐ Elle échoue à cause d'une erreur.

```
p1 = Point(2,3)
p2 = Point()
print(Point.nombre_de_points)
```

q2 :

- ☐ Elle affiche la chaîne "1".
- ☐ Elle affiche la chaîne "2".
- ☐ Elle affiche la chaîne "3".

```
print(p.distancenull())
```

q3 :

- ☐ 0.121426951
- ☐ 1.4142135623730951
- ☐ 5.0
- ☐ 25.0

```
print(p.distance(p1))
```

q4 :

- ☐ 0.121426951
- ☐ 1.4142135623730951
- ☐ 5.0
- ☐ 25.0

```
del p
print(Point.nombre_de_points)
```

q5 :

- ☐ Elle affiche la chaîne "Le point 3 4 est détruit." sur la sortie standard.
- ☐ Elle échoue à cause d'une erreur.
- ☐ Elle affiche la chaîne "2" sur la sortie standard.
- ☐ Elle affiche la chaîne "3" sur la sortie standard.

1 Modèle de données pour le décompte des heures travaillées

On souhaite définir un objet dont le rôle est de décompter le nombre d'heures travaillées par un employé donné.

Question 1

Définir une classe CompteEpargneTemps() pour gérer le nombre d'heures travaillées par un employé. Le constructeur de cette classe doit initialiser deux attributs nom et solde, avec les valeurs par défaut "Michel" et 0.

Définir une méthode 'ajouter' qui permet d'ajouter un certain nombre d'heures à un compte.

Définir une méthode 'enlever' qui permet d'enlever un certain nombre d'heures à un compte.

Définir une méthode 'afficher' qui permet d'afficher le nom de l'employé et le solde de son compte épargne temps.

Définir une méthode 'nombre' qui retourne le nombre d'employés existants.

q6 :

Question 2

Écrire un programme `test_compte_epargne_temps.py` qui successivement :

Crée un compte pour "Marie" avec 30 heures.

Crée un compte pour "Ahmed" avec 24 heures.

Affiche le compte de Ahmed.

Ajoute 10 heures.

Affiche le compte de Ahmed.

Enlève 20 heures.

Affiche le compte de Ahmed.

Affiche le compte de Marie.

Détruit son compte.

Affiche le nombre de comptes.
Détruit le compte de Ahmed.
q7 :

2 Modèle de données pour une boîte de réception de messages

Question 1

On définit un objet représentant une boîte de réception pour des messages SMS. Pour cela, on définit une classe `memoire_SMS`. Un objet de cette classe (messagerie) contiendra une liste de SMS. Chaque SMS est un n-uplet `(vu,numéro,temps,texte)` où `vu` est un booléen qui indique le statut (vu ou pas vu), le numéro, le temps et le texte sont des chaînes de caractères. Les méthodes suivantes doivent être implémentées pour réaliser les services d'une boîte de réception :

`ajouter(numéro,temps,text)` crée un nouveau tuple SMS et l'insère à la suite des autres messages. vu est mis à False.

`nombre()` retourne le nombre de messages.

`non_lu()` retourne les indices des messages non vus.

`nombre_non_lu()` retourne le nombre de messages non vus.

`message(i)` retourne `(numéro,temps,texte)` du message d'indice i et change le statut vers vu. S'il n'y a pas de message d'indice i retourne `None`.

`messages_non_lu()` affiche tous les messages non vus et change leur statut.

`efface(i)` efface le message d'indice i.

`effacenum(num)` efface tous les messages du numéro num.

`remettreazero()` efface toutes les messages.

`copier(mem,eff)` copie tous les messages de la messagerie vers la messagerie mem. Si eff est égal à True, les messages se trouvant initialement dans mem sont écrasés. Par défaut eff est égal à False.

`clone()` renvoie une nouvelle messagerie avec le même contenu.

`recherche(mot)` renvoie une liste de tous les textes de messages contenant mot.

q8 :

Question 2

Ecrivez un programme qui successivement :

¡¡ Crée une messagerie m

Ajoute les messages : (555,"10 :05","Salut") (555,"10 :07","Bonjour") (444,"10 :10","Hello, ça va ?") (444,"10 :13","Allo")

Recherche les messages contenant "Hello" et les imprime.

Crée une messagerie m1.

Copie tous les messages de m vers m1.

Imprime le nombre de messages dans m1.

Imprime le troisième message de m1.

Efface le troisième message de m.

Efface de m tous les message du numéro 555

Clone la messagerie m et la met dans m2.

Imprime les messages dans m2.

Mets à zero m.

q9 :

3 Réprésentation d'un annuaire au format CSV

On souhaite écrire un module CSV définissant une classe d'objets `Annuaire`, dont le rôle est de maintenir les associations entre des données et des personnages sous la forme d'un fichier au format CSV (Comma Separated Values) : chaque ligne représentera une entrée de l'annuaire, avec 5 champs séparés par des virgules, représentant les 5 caractéristiques du personnage concerné : nom, prénom, profession, adresse et numéro de téléphone. Par exemple :

```
Haddock,Archibald,Capitaine,Chateau de Moulinsart,421
Derkozy,Niclas,Marin,Champs-Elaises,234
Odema,Darak,Matelot,Maison Grise,555
```

ou encore :

Voir annuaire.txt. [1]

Le module devra une autre classe `Personnage`. Chaque instance de la classe `Annuaire` contiendra une liste d'instances de la classe `Personnage`, qui elles-mêmes auront pour attributs les 5 caractéristiques précédemment citées.

La classe `Personnage` contiendra un constructeur pour la classe Personnage, prenant en paramètre une chaîne de caractères au format CSV. (Rappel : utilisez split.)

Il est possible de définir une méthode spéciale pour la représentation d'un objet par une chaîne de caractères, `__str__`. Cette méthode doit renvoyer une chaîne de caractères. Lorsque cette méthode est définie, la méthode `print` d'un objet utilise la chaîne renvoyée par __str__. Définir cette méthode pour permettre un affichage agréable pour l'utilisateur.

La classe `Annuaire` contiendra un constructeur prenant en paramètre un nom de fichier au format CSV, servant à initialiser son contenu.

La classe `Annuaire` contiendra une méthode `__str__` pour `Annuaire` qui retourne un annuaire trié sous forme d'une chaîne de caractères.

La classe `Annuaire` contiendra une méthode copie prenant en paramètre un nom de fichier et y copiant les représentations des entrées de l'annuaire au format CSV, triées par ordre alphabétique. Pour cela, on pourra créer un dictionnaire et en parcourir la liste des clés triée.

La classe `Annuaire` contiendra une méthode `liste_pages_jaunes` prenant en paramètre une chaîne de caractères et renvoyant la liste des entrées de l'annuaire ayant la profession correspondante.

La classe `Annuaire` contiendra une méthode `cherche_pages_jaunes` affichant le nombre et la liste des réponses obtenues à l'aide de la méthode précédente.

La classe `Annuaire` contiendra une méthode `liste_pages_blanches` prenant en paramètre une chaîne de caractères et retournant la liste des personnes dont le nom ou le prénom contient cette chaîne.

La classe `Annuaire` contiendra une méthode `cherche_pages_blanches` affichage le nombre et la liste des réponses obtenues à l'aide de la méthode précédente.

La classe `Annuaire` contiendra une méthode `modifier` prenant un nom en paramètre et demandant à l'utilisateur d'entrer les nouvelles coordonnées de la personne correspondante. Pour cela, on pourra écrire une méthode `selectionne` qui teste si le nom apparaît dans l'annuaire et, s'il apparaît plusieurs fois, demande à l'utilisateur de choisir parmi les personnes correspondantes.

Ajouter des méthodes les méthodes `ajouter` et `supprimer` qui ont chacune un paramètre chaîne de caractères.

(optionnel) Écrire une interface textuelle simple permettant à un utilisateur de manipuler un annuaire.

q10 :

1. `Fichier fourni avec le sujet.`