iP Net Institute of Technology | Next Generation IT Professionals

**Année Académique 2019- 2020**

# DEVOIR 1 – SEMESTRE 02

**Niveau :** LICENCE

**Filières :** Génie Logiciel, Développement Web, Web Design, Cyber Sécurité, Réseau Système et Sécurité

**EPREUVE :** Algorithme & Programmation Structurée avec Python II CSC 242

DUREE : 3 HEURES Chargé du Cours : KPIZIA Taliou

**NB : Epreuve à traiter sur machine.**

## Problème :

### GESTION DES INTERVENTIONS DANS LA SOCIETE ZOUBA

La gestion d'une intervention regroupe le coût du déplacement et le coût de la main-d'œuvre.
Le coût du déplacement est calculé en tenant compte de la distance entre l'agence dont relève le client et du client ; le tarif kilométrique retenu pour l'intervention dépend de la puissance du véhicule
Le coût de la main-d'œuvre est déterminé à partir du coût horaire du technicien chargé de l'intervention et de la durée de celle-ci, toute heure commencée étant entièrement comptabilisée. Le coût horaire du technicien est obtenu en majorant le coût horaire correspondant à son grade par un coefficient dépendant de son ancienneté dans la société

**de 5 à 10 ans majoration de 5 %**
**de 11 à 15 ans majoration de 8 %**
**plus de 15 ans majoration de 12 %**

La société ZOUBA a décidé de développer un outil logiciel permettant de comparer le montant dû payé par le client et le coût total des interventions relatives à ce contrat.
L'outil sera développé en python qui est un langage de programmation orienté objets.
Ci-dessous, la définition en algorithmique des classes :

**Intervention = classe**
**Privé**
numero : entier
date : chaîne
duree : entier
tarifkm : réel /* tarif kilométrique retenu */
technicien : Employé /* employé ayant effectué l'intervention*/
**Public**
Constructeur **Intervention(numero :entier,date :chaine,duree :chaine,tarifKm :reel, technicien :Employe)**
Procédure **affiche( )**

fonction **fraisKm(dist: réel): réel**
fonction **fraisMo( ) : réel**
**Fin classe Intervention**

Le Constructeur qui permet d'initialiser tous les attributs
La méthode **affiche( )** qui affiche les informations concernant une intervention et le technicien
La méthode **fraisKm(dist: réel): réel** calcule les frais kilométriques occasionnés par une intervention, la distance parcourue étant passée en paramètre
La méthode **fraisMo( ) : réel** calcule et retourne les frais de main-d'œuvre occasionnés par une intervention.

**Employe= classe**
**Privé**
numero : entier
nom : chaîne
qualification : Grade
dateEmbauche : chaîne
**Public**
fonction **coutHoraire( ) : réel**
Fonction **getNumero() : entier** /* qui retourne le numéro employé */
Fonction **getNom(): chaîne** /* fonction qui retourne le nom de l'employé */
Fonction **getQualifiction(): Grade** /* qui retourne la qualification de l'employé*/
Fonction **getDateEmbauche() : chaîne**
Fonction **getAncienete(date :chaine) :entier**
**Fin classe Employe**

La méthode **coutHoraire( ) : réel** détermine et retourne le coût horaire de l'employé en fonction de sa qualification et de son ancienneté
**La méthode getAncienete(date :chaine) :entier** qui retourne le nombre entier d'années écoulées à partir d'une date fournie en paramètre jusqu'à la date du jour.

**Grade = classe**
**Privé**
code : caractère
libelle : chaîne
taux : réel
**Public**
fonction **getCode() : caractère** /* retourne le code du grade*/
fonction **getLibelle() : chaine** /* retourne le libellé du grade*/
fonction **tauxHoraire( ) : réel**
**Fin classe Grade**

La méthode **tauxHoraire( ) : réel** retourne le taux horaire spécifique du grade

**Contrat = classe**

**Privé**

numero: entier
date: chaîne
client: Client
montantContrat : réel /* montant payé par le client*/
interventions : tableau {l :500) de intervention
nbIntervention : entier /* nombre d'interventions effectives réalisées pour le contrat et */

**Public :**

fonction **montant( ) : réel** /* retourne le montant du contrat payé par client */
fonction **ecart( ) : réel**

**Fin classe Contrat**

La méthode **ecart( ) : réel** détermine et retourne l'écart entre le montant du contrat et le coût des interventions effectuées sur ce contrat

**Client = classe**

**Prive**

numero : entier
nom : chaîne
adresse : chaîne
codePostale : chaîne
ville : chaîne
nbKm : entier /* distance du client à la société en km */

**Public**

fonction **distance (): réel**

**Fin classe Client**

La méthode **distance** retourne la distance en kilomètres qui sépare le site du client de la société

**TRAVAIL A FAIRE**

1/ Implémentez en python toutes les classes décrites plus haut.

2/Ecrie un module pour tester pour un employé l'écart entre le montant de son contrat et le cout de ses interventions en créant les objets adéquats avec vos jeux de données.

**Consignes :**

A la place du tableau **interventions : tableau {l :500}** de la classe **Contrat** utilisez une liste pour stocker les interventions

Considérez les dates comme des objets **Datetime** décrit ci-dessous au lieu de chaine de caractère.

**NB :**

Le module **Datetime** est un module python qui permet de manipuler des dates et des durées sous formes d'objets.

Exemple :

**from datetime import datetime**
datetime(2020,7,19)
La méthode now() renvoi la date du jour
La différence entre deux objets **Datetime** donne un objet **timedelta**. Cet objet représente une durée en jours, secondes et microsecondes.

Exemple :

duree= datetime.now()- datetime(2019,7,19)
nbJours=duree.days