420-2N6-RO

**EXAMEN Formatif**

**Programmation 2   
Profil réseautique**

%

Durée : 2h00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom :** |  | |  | **Note:** |
|  |  | |  | **/ 50** |
|  |  |  |  |  |

**Aucun** **point** n’est octroyé si la capture d’écran n’est pas présente dans les réponses demandant d’écrire du code.

**ATTENTION !** Il y Six (6) questions dans cet examen FORMATIF, chaque question est sur une page différente.

Vous avez le droit aux notes de cours qui vous sont fournies.

## Création d'une classe simple /2

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

À partir du diagramme UML, compléter la classe dans le fichier Q1\_livre.py

La méthode obtenir\_info() doit retourner le titre du livre, l'auteur et la date de publication.

Quand la classe est terminée, les commandes :

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

Devraient donner ceci dans le terminal :



## Création d'une classe un peu plus complexe /15

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

À partir du diagramme UML, complétez la classe dans le fichier Q2\_contact.py

Prenez note :

1. Les attributs adresse, email, et num\_telephone doivent être optionnels lors de la création d'une nouvelle instance.
2. TOUTEFOIS, vous devez faire une vérification, l'instance doit recevoir un email et/ou un num\_telephone lors de la création.
3. Si le constructeur ne reçoit pas de valeur pour email et num\_telephone, il faut soulever une erreur de type ValueError.

La méthode mettre\_a\_jour() n'a pas à être implémentée. Vous devez simplement mettre un pass.

## Création d'une classe parent /15

Vous avez deux classes dans le fichier Q3\_chat\_et\_chien.py, ces deux classes sont très similaires.

Vous devez compléter la classe AnimalDeCompagnie et modifier les classes Chien et Chat afin qu'ils héritent de la classe AnimalDeCompagnie.

Vous devez réduire la répétition de code.

A la fin du document, il y a des fonctions print() pour tester rapidement le fonctionnement des classes. Le résultat des fonctions print() devrait rester inchangé.

Résultat des print() :

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

## Ajout de méthodes /15

Le fichier Q4\_voiture.py contient un classe Voiture faisant partie d'un programme de gestion pour une entreprise de revente de voitures usagées.

Vous devez ajouter deux méthodes à cette classe.

1. La méthode calculer\_prix\_vente qui devra prendre en paramètres une valeur prix et une valeur kilometrage et calculer un prix de vente final.
   1. Un rabais est calculé en fonction du kilometrage. Le rabais est de 0.05$ par kilomètre.
   2. Le rabais ne peut pas dépasser 7000$
   3. Cette méthode doit être indépendante de l'instance. Elle ne fait pas référence à l'instance et peut être appelée avec les valeurs qu'on a besoin pour faire le calcul.
2. La méthode obtenir\_prix qui doit retourner le prix avec rabais de l'instance. Cette méthode ne doit pas faire les calculs, mais simplement appeler la méthode calculer\_prix\_vente avec les valeurs correspondantes à l'instance en paramètres.

Après que vous avez terminé le code de ces deux méthodes, les tests rapides au bas du fichier Q4\_voiture.py devraient imprimer le résultat suivant dans le terminal :



## Ajout de propriétés /15

La classe Patient dans Q5\_patient.py a les attributs nom, grandeur, poids, age et num\_telephone.

La grandeur d'un patient est en centimètres tandis que le poids est en kilogrammes.

Ajoutez les propriétés grandeur\_pouces et poids\_livres afin d'obtenir la grandeur en pouces et le poids en livres.

1 pouce est égal à 2,54 centimètres.

1 kilogramme est égal 2,20462 livres.

Résultat attendu des print() à la fin du fichier Q5\_patient.py :

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

## Attributs et méthode /15

Un logiciel pour la gestion des bibliothèques du Québec est en train d'être développé. Des instances de la classe Bibliothèque seront utilisées pour représenter chacune des bibliothèques au Québec.

Dans le fichier Q6\_bibliotheque.py, vous devez ajouter deux attributs.

1. Un attribut liste\_livres qui sera propre à chaque bibliothèque crée. Cet attribut correspond à une liste qui devra être vide lors de la création de l'instance.
2. Un attribut liste\_livres\_Quebec qui consistera en une liste de livres partagée entre toutes les bibliothèques du Québec.

Vous devez ensuite compléter la méthode ajouter\_livre. Cette méthode prendra un livre en paramètre. Le comportement de la méthode est décrit dans le diagramme de flux suivant :

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Lorsque les listes de livres sont imprimées, vous devriez obtenir le résultat suivant :

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

FIN !

Enregistrez ce fichier Word et remettez-le au professeur selon ses directives.