

Cours : Programmation orientée objet en C#  
Cote du cours : 26978  
Session : Hiver 2022

## Laboratoire de révision

### Les classes Adresse, Contact et CarnetContact

#### Partie 1 : Adresse

Dans un premier temps, nous allons écrire une classe nommée **Adresse**. Une adresse est caractérisée par un numéro (entier), le nom de la rue (chaîne), la ville (chaîne), le code postal (chaîne) et la province (chaîne). Vous devez prendre en considération les règles suivantes pour les attributs :

- Le code postal est une chaîne composée de 6 caractères. Sinon on affecte la valeur « K1K1K1 ».
- La province est une chaîne composée de deux lettres en majuscule. Sinon on affecte la valeur « QC ».

On vous demande de :

- 1- Définir la classe **Adresse** et ses attributs.
- 2- Définir les propriétés associées aux attributs.
- 3- Définir un constructeur initialisant le numéro, le nom de rue, la ville, le code postal et la province.
- 4- Définir une méthode nommée **Egalite** permettant de tester l'égalité de l'adresse courante avec l'adresse passée comme paramètre. Deux adresses sont égales si elles ont le même numéro, le même nom de rue et la même province.
- 5- Définir la méthode **ToString** permettant de convertir un objet adresse vers une chaîne de caractères. La représentation de l'adresse est comme suit :

91, boulevard de la Gappe, Gatineau, QC, J8T-0B5

- 6- Surcharger l'opérateur « == » pour comparer deux adresses.

#### Partie 2 : Contact

Dans un second temps, nous allons écrire une classe nommée **Contact**. Un contact est caractérisé par son code (entier), son nom (chaîne), son prénom (chaîne), son téléphone (entier) et son adresse (Adresse). Vous devez prendre en considération les règles suivantes pour les attributs :

- Le code **s'incrmente** lors de la création d'un nouveau contact.
- Les prénoms sont en minuscules et les noms en majuscules.

- Le téléphone, S'il existe, doit être composé de dix chiffres. Sinon on affecte la valeur « 613111111 » comme valeur par défaut.

On vous demande de :

- 1- Définir la classe **Contact** et ses attributs.
- 2- Définir les propriétés associées aux attributs.
- 3- Définir un constructeur initialisant le nom et le prénom.
- 4- Définir un constructeur initialisant le nom, le prénom et le téléphone.
- 5- Définir un constructeur initialisant le nom, le prénom, le téléphone et l'adresse.
- 6- Définir une méthode nommée **HasPhone** permettant de retourner true si le contact courant possède un numéro de téléphone et false sinon.
- 7- Définir la méthode **ToString** permettant de convertir un objet contact vers une chaîne de caractères. La représentation du contact est comme suit :

Code : 1

Nom : BOUHLEL mohamed

Téléphone : (613)123-4567

Adresse : 91, boulevard de la Gappe, Gatineau, QC, J8T-0B5

- 8- Définir une méthode nommée **Egalite** permettant de tester l'égalité du contact courant avec le contact passé comme paramètre. Deux contacts sont égaux s'ils ont le même code et le même prénom.

### Partie 3 : CarnetContact

En troisième lieu, nous allons écrire une classe nommée **CarnetContact**. Un carnet de contact est caractérisé par un ensemble de contact.

On vous demande de :

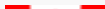

- 1- Définir la classe **CarnetContact** avec ses attributs.
- 2- Définir les propriétés associées aux attributs.
- 3- Définir un constructeur.
- 4- Définir une méthode nommée **AddContact** permettant d'ajouter un contact dans le tableau de contact.
- 5- Définir une méthode nommée **AddContact** permettant d'ajouter un contact dans le tableau de contact en fournissant son nom et son prénom.
- 6- Définir une méthode nommée **AddContact** permettant d'ajouter un contact dans le tableau de contact en fournissant son nom, son prénom et son téléphone.
- 7- Définir une méthode nommée **AddContact** permettant d'ajouter un contact dans le tableau de contact en fournissant son nom, son prénom, son téléphone et son adresse.
- 8- Définir une méthode nommée **FindContact** permettant de retourner le contact dont le code est passé comme paramètre.
- 9- Définir une méthode nommée **FindContact** permettant de retourner le premier contact dont le nom et la province sont passés comme paramètres.
- 10- Définir une méthode permettant de retourner la liste des contacts dont la province est passée comme paramètre.
  - a. Peut-on nommer la méthode **FindContact**? *Pourquoi?* (Répondre sous forme de commentaire dans le code)

b. Si ce n'est pas possible, alors nommer la méthode **FindContactProvaince**.

- 11- Définir une méthode nommée **DeleteContact** permettant de supprimer un contact à partir de son numéro.
- 12- Définir une méthode **ToString** permettant de convertir le carnet de contact vers une chaîne de caractères. La signature de la méthode est comme suit : `public String ToString()`. La représentation du carnet de contact est libre mais doit afficher tous les contacts avec leur adresse.

## Partie 4 : Extension

1. Ajouter à la classe **Adresse** une méthode d'extension nommée **GetProvince** permettant de déterminer la province à partir du code.

		Codes postaux canadiens															
<u>NL</u>	<u>NS</u>	<u>PE</u>	<u>NB</u>	<u>QC</u>		<u>ON</u>			<u>MB</u>	<u>SK</u>	<u>AB</u>	<u>BC</u>	<u>NU/NT</u>	<u>YT</u>			
<u>A</u>	<u>B</u>	<u>C</u>	<u>E</u>	<u>G</u>	<u>H</u>	<u>J</u>	<u>K</u>	<u>L</u>	<u>M</u>	<u>N</u>	<u>P</u>	<u>R</u>	<u>S</u>	<u>T</u>	<u>V</u>	<u>X</u>	<u>Y</u>

2. Ajouter à la classe **Contact** une méthode d'extension nommée **Appartenance** permettant de tester si le contact appartient à la province passée comme paramètre.
3. Ajouter à la classe **CarnetContact** une méthode nommée **NbContact** permettant de retourner le nombre de contact dont la province est passée comme paramètre.

## Partie 5 : TestCarnet

Enfin, on vous demande d'écrire une classe nommée **TestCarnet** pour tester les deux classes précédentes. Cette classe contient uniquement la méthode **Main** et doit réaliser les opérations suivantes :

- 1- Créer deux adresses nommées respectivement d1 et d2.
- 2- Afficher l'adresse d1.
- 3- Créer 4 contacts nommés respectivement **c1**, **c2**, **c3** et **c4**. Utiliser différents constructeurs pour créer vos objets.
- 4- Afficher les contacts **c1** et **c2**.
- 5- Tester l'égalité de **c1** et **c2** et afficher le résultat.
- 6- Est-ce que c1 possède un téléphone.
- 7- Définir le carnet d'adresse cd1.
- 8- Ajouter vos contacts c1, c2, c3 et c4 à cd1.
- 9- Ajouter d'autres contacts en appelant les autres méthodes d'ajout de contact (AddContact). Vous devez utiliser toutes les méthodes.
- 10- Chercher et afficher le contact dont le numéro est 5.
- 11- Chercher et afficher le contact dont le numéro est 500.
- 12- Chercher et afficher le contact dont le nom est « Bouhleh » et la province est « QC »
- 13- Afficher le nombre de contact de la province « QC ».
- 14- Afficher la liste des contacts de la province « ON ».
- 15- Afficher la liste des contacts de la province « QC ».

- 16-** Supprimer le contact dont le numéro est 5.
- 17-** Supprimer le contact dont le numéro est 500.
- 18-** Afficher le carnet de contact.