

Laboratoire #2

Objectifs

- Composition d'objets
- Insertion d'un objet dans un tableau d'objets
- Retrait d'un objet dans un tableau d'objets
- Utilisation de méthodes privées
- Utilisation de `JOptionPane.showInputDialog` et de `JOptionPane.showMessageDialog`
- Recherche.
- Fichiers texte.

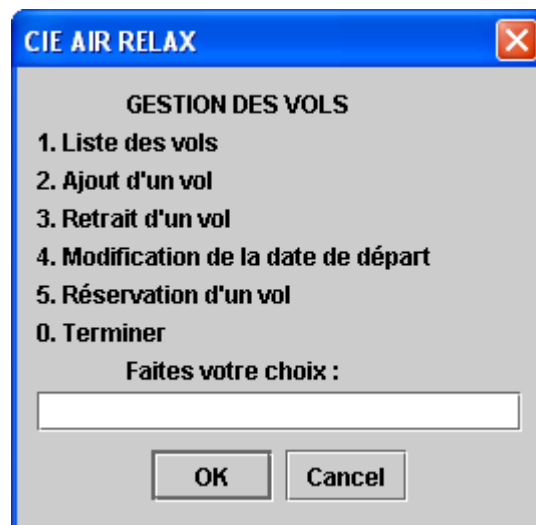
Sujet

On vous engage afin de réaliser un projet de gestion des vols d'une compagnie aérienne. Votre projet sera utilisé par différentes compagnies aériennes. Chaque compagnie dispose d'un fichier (des vols déjà existants) dont **le nom est celui du nom de la compagnie suivi de l'extension .txt** Chaque enregistrement des fichiers contient :

- Le numéro du vol (5 chiffres)
- La destination du vol (20 caractères)
- La date de départ du vol (sous la forme jj mm aaaa)
- Le nombre total des réservations à date

Les enregistrements sont par ordre croissant du numéro du vol. **On ne connaît pas le nombre d'enregistrements qu'il y a dans les fichiers.**

Dans votre projet, la classe qui contiendra le « main » devra faire afficher un menu : par exemple, pour la compagnie « Cie Air Relax » nous aurons le menu suivant, à l'aide de `JOptionPane.showInputDialog`

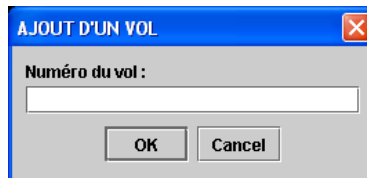


N'oubliez pas de produire un message d'erreur si le choix n'est pas entre 0 et 5

Travail à faire

Votre projet va comporter 3 classes soient :

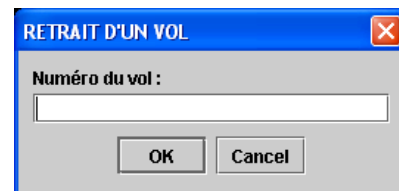
1. La classe **Date** ayant les attributs entiers **jour**, **mois**, **an**, le constructeur par défaut et celui avec 3 paramètres, les méthodes « **get** » et la méthode **toString** qui retourne la chaîne contenant les attributs séparés par "/"
2. La classe **Vol** dont les caractéristiques sont :
 - attributs **d'instances**: numéro du vol, sa destination, la date de départ (objet de la classe Date), et le nombre total de réservations à date
 - le constructeur permettant d'initialiser les 4 attributs d'instances avec les valeurs reçues en paramètres
 - attribut de classe (static) **nombreVols=0**;
 - les méthodes « **set** » pour seulement le total des réservations et la date de départ
 - les méthodes « **get** » pour tous les attributs
 - la méthode **toString()** qui retourne une chaîne contenant les attributs séparés par des "\t" (un tab)
3. La classe **GestionVols** qui contiendra le main et d'autres méthodes illustrées plus loin. Dans la classe vous allez créer un tableau **tabVols** que contient les vols pour la compagnie **CieAirRelax** :
 - la constante de classe publique **MAX_PLACES = 340** (le nombre de places que contient un avion)
 - une méthode **chargerVols** pour faire l'ouverture et la lecture du fichier **CieAirRelax.txt** (le fichier est dans LEA, section Documents) de façon à remplir le tableau **tabVols** d'objets Vol. Ensuite, il ferme le fichier. Le séparateur des données ce sont des espaces)
 - la méthode d'instance **insérerVol** doit s'assurer qu'il reste de la place dans le tableau des vols, sinon prévoir un message d'erreur adéquat avec **JOptionPane.showMessageDialog**. S'il reste de la place, elle permet de saisir le numéro du nouveau vol à l'aide de **JOptionPane.showInputDialog** sous la forme :



elle fera appel à la méthode **rechercherVol** (retourne true si le vol existe déjà sinon retourne false), car pour pouvoir ajouter un nouveau vol, le numéro du vol ne doit pas déjà exister (prévoir un message d'erreur approprié si ce n'est pas le cas). Si le numéro est accepté, saisir alors la destination, la date de départ (**JOptionPane.showInputDialog**) et insérer le vol.

- la méthode **retirerVol** permet de saisir le numéro du vol à l'aide **JOptionPane.showInputDialog** sous la forme :

Elle fera appel à la **rechercherVol** car pour pouvoir retirer un vol, le du vol doit être existant (prévoir un message d'erreur approprié si ce n'est pas le cas). Si le numéro existe, à l'aide d'une seule fenêtre **JOptionPane.showInputDialog**, faire afficher sa destination, sa date et le total des réservations et demander la confirmation du retrait : « **Désirez-vous vraiment retirer ce vol (O/N)?** » Si oui, retirer le vol du tableau et ne pas oublier de diminuer le nombre de vols actifs.

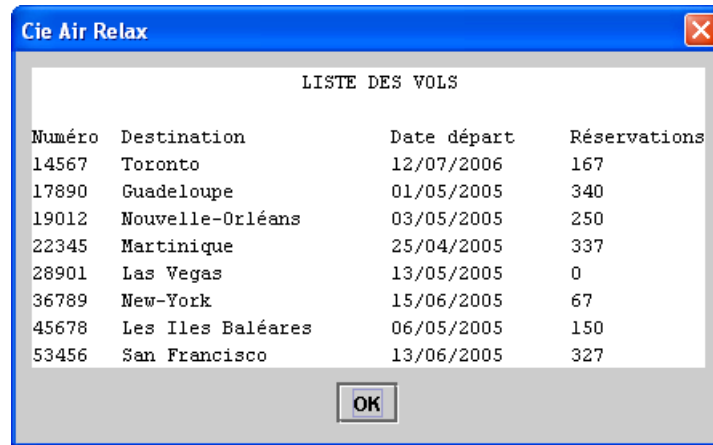


numéro
pas le

- la méthode **modifierDate** permet de saisir le numéro du vol dont on veut modifier la date, à l'aide de **JOptionPane.showInputDialog** (la barre de titre contiendra le titre **MODIFICATION DE LA DATE DE DÉPART**). Elle fera appel à **rechercherVol**. Si le numéro n'existe pas prévoir le message d'erreur approprié, sinon à l'aide d'une seule fenêtre **JOptionPane.showInputDialog** faire afficher sa destination, sa date de départ actuelle et demander l'entrée de la nouvelle date (dans la forme JJ/MM/AAAA).
- la méthode **reserverVol** permet de saisir le numéro du vol que le client désire réserver, à l'aide de **JOptionPane.showInputDialog** (la barre de titre contiendra le titre **RÉSERVATION D'UN VOL**). Elle fera appel à la **rechercherVol**. Si le numéro n'existe pas prévoir le message d'erreur approprié, sinon s'il reste de la place dans l'avion, à l'aide d'une seule fenêtre **JOptionPane.showInputDialog** faire afficher sa destination, sa date de départ, le nombre de places

restantes dans l'avion et demander l'entrée du nombre de places que le client désire réserver et faites la réservation. S'il n'y a plus de place, faire afficher à l'aide de **JOptionPane.showMessageDialog** le message adéquat.

- la méthode **listeVols** qui fait afficher la liste des vols existants à l'aide de **JTextArea** [fonte courier, de style **Font.PLAIN** et de taille 12] à transmettre à **JOptionPane.showMessageDialog**, sous la forme (la barre de titre contiendra le nom de la compagnie):



Numéro	Destination	Date départ	Réservations
14567	Toronto	12/07/2006	167
17890	Guadeloupe	01/05/2005	340
19012	Nouvelle-Orléans	03/05/2005	250
22345	Martinique	25/04/2005	337
28901	Las Vegas	13/05/2005	0
36789	New-York	15/06/2005	67
45678	Les Iles Baléares	06/05/2005	150
53456	San Francisco	13/06/2005	327

- la méthode d'instance **ecrireFichier** qui ouvre le fichier de la compagnie en écriture (pas en append mais le récréer) et écrit le tableau d'objets mis à jour dans le fichier.

NOTA : Ce travail peut être fait par des équipes de maximum trois étudiants.

Déposer votre travail compressé dans LEA.