

## TP : LES COLLECTIONS

### Exercice 1 :

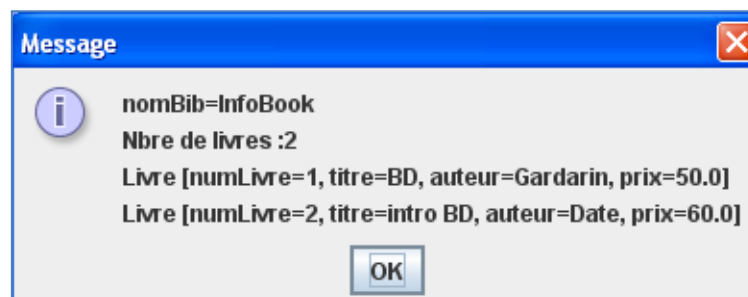
1. Créer une classe « Livre » qui comprend 4 attributs (numLivre, titre, auteur, prix) et 5 méthodes :

- Livre(numLivre, titre, auteur, prix) : constructeur
- int getNumLivre( ) : accesseur qui renvoie le numéro du livre courant
- double getPrix( ) : accesseur qui renvoie le prix du livre courant
- void setPrix(double prix) : mutateur qui modifie le prix du livre courant
- String toString( ) : renvoie une description textuelle complète du livre actuel

2. Créer une classe « Bibliothèque » qui comprend :

- Le nom de la bibliothèque « nomBib »
- une collection de type ArrayList appelé « listeLivres » capable de contenir un certain nombre de livres
- Un constructeur Bibliothèque(nomBib). La liste des livres est supposée initialement vide
- Une méthode ajouterLivre(livre)
- Une méthode rechercherLivre(numLivre) qui recherche un livre à partir de son numéro. S'il est trouvé, une description complète est retournée sinon on renvoie juste le message « Livre inexistant »
- Une méthode supprimerLivre(numLivre) qui supprime le livre dont le numéro est fourni en paramètre (s'il existe). Un message de succès/échec de l'opération est renvoyé.
- Une méthode modifierPrix(numLivre, prix) qui modifie le prix du livre dont le numéro est fourni en paramètre (s'il existe). Un message de succès/échec de l'opération est renvoyé.
- une méthode toString( ) qui retourne le nom de la bibliothèque, le nombre de livres disponibles ainsi que la liste complète de ces livres.

### Exemple



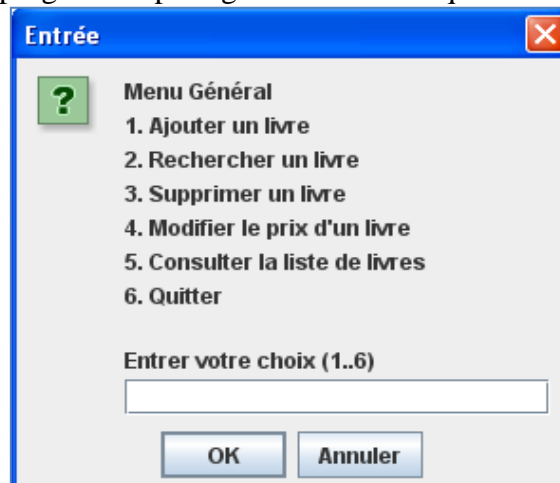
3. Créer une classe « TestBibliothèque » réduite à une méthode main() permettant de :

- Créer la bibliothèque B qui porte le nom « InfoBook »
- Ajouter les livres suivants à la bibliothèque B

numLivre	Titre	Auteur	Prix (TND)
1	Bases de données	G. Gardarin	50
2	Introduction aux bases de données	C. Date	60
3	Conception des bases de données avec UML	G. Roy	45
4	UML2 pour les bases de données	C. Soutou	35
5	De UML à SQL : Conception de bases de données	C. Soutou	30

- Chercher le livre N°4 puis le livre N°6
- Supprimer le livre N°3
- Rendre le prix du livre N°2 égal à 50 D
- Afficher une description de la bibliothèque avec le nombre et la liste complète des livres disponibles.

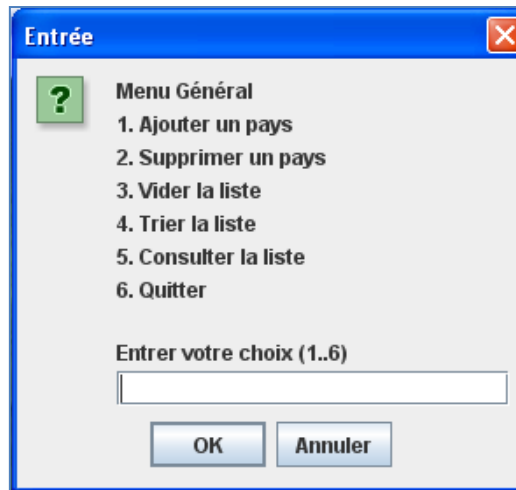
4. Améliorer votre programme pour gérer la bibliothèque à travers le menu suivant:



NB : Utiliser les utilitaires de la bibliothèque « swing » pour effectuer les opérations d'entrée/sortie en mode graphique.

## Exercice 2 :

1. Créer une classe « CollectionPays » réduite à une méthode main( ) qui permet de créer une collection nommée « collPays » de type ArrayList qui contient des noms de pays (sous forme de chaînes de caractères) et de gérer cette liste à travers un menu qui offre les options suivantes :



N.B :

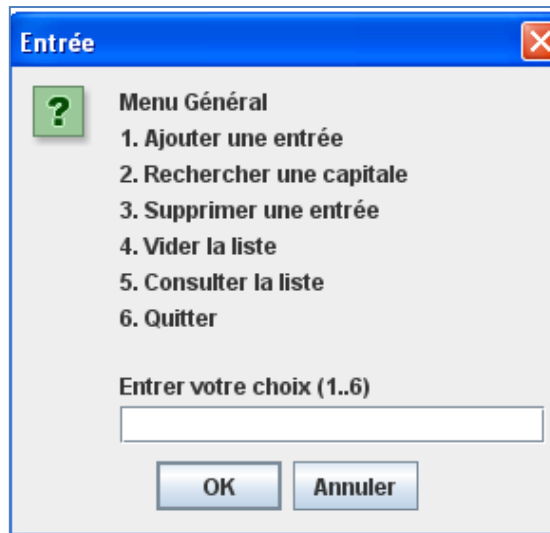
- Après chaque opération le programme doit aviser l'utilisateur si l'opération s'est bien déroulée ou non.
- Si l'utilisateur tente d'ajouter un pays déjà existant, le programme doit afficher un message qui le signale
- De même, si l'utilisateur tente de supprimer un pays inexistant, le programme doit le signaler.

Vous pouvez éventuellement utiliser les méthodes suivantes de traitement de collections :

```
- int size()
- boolean contains(Object o)
- boolean add(Object o)
- void clear()
- boolean isEmpty()
- Object remove(int index)
- collections.sort(liste)
```

2. Modifier la structure de collection utilisée pour que la liste soit triée en permanence (sans avoir besoin d'appeler la méthode *sort*) et n'accepte pas les doublons (sans avoir besoin de faire le contrôle lors de la saisie).

3. Créer une classe « Capitales » réduite à une méthode *main( )* qui permet de créer une collection nommée « collPays » sous forme d'une table associative de type « *HashMap* » qui contient des couples (nomPays, nomCapitale) et de gérer cette liste à travers un menu qui offre les options suivantes :



Vous pouvez éventuellement utiliser les méthodes suivantes de traitement de « HashMap » :

```
- int size()
- void clear()
- boolean containsKey(Object key)
- boolean containsValue(Object value)
- V put(K key, V value)
- V get(Object key)
- boolean isEmpty()
- V remove(Object key)
- Collection<V> values()
```

4. Modifier la structure de collection utilisée pour que la liste soit triée en permanence selon la clé (nomPays).

### **Exercice 3 :**

Créer un programme permettant de gérer un annuaire qui contient un ensemble de contacts. Pour chaque personne dont on connaît le nom, on crée une fiche comportant le numéro de téléphone et l'adresse.