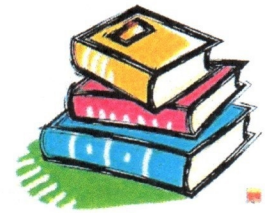

TpOBJET_04
Le livre comparable- L'utilisateur bavard
Projet Bookstore

Thèmes : utilisation de l'interface pré-existante Comparable<T>, un tableau d'objets, le tri d'un tableau d'objets, le codage d'une interface.

1 item: L'interface Comparable<T>

Si les livres sont comparables entre eux, ils pourront être triés. D'un point de vue fonctionnel, pour les livres, vous fixez votre critère de comparaison naturel (natural sort)(ou « par défaut ») sur le nombre de pages avec un ordre par défaut croissant (natural order).
Nous utilisons l'interface java.lang.Comparable<T>. Consultez la javadoc et les documents mis à votre disposition.
Travail à réaliser:
La classe Livre réalise (implémente) l'interface java.lang.Comparable<T> .
Adaptez la généricité à votre cas et écrivez ici la signature du Livre :
.....
L'interface est un contrat et, donc, vous devez redéfinir la méthode de l'interface. De plus, avec les facilités d'Eclipse, vous pouvez utiliser l'annotation @Override pour vous aider à générer/contrôler la signature de la méthode. Ecrivez-ici la signature de la méthode à développer :.....
Reprenez votre travail précédent avec l'item « Comparaison de deux livres en utilisant une méthode d'instance». Testez en comparant 2 livres.

2 item: Méthode sort. Critère de tri naturel d'un tableau de livres

Travail à réaliser : Vous placez quelques livres dans un tableau et appelez la méthode "java.util.Arrays.sort" . Consultez la javadoc de la classe Arrays : pour cette méthode sort, il existe plusieurs formes (surcharge). Voici la signature de celle qui nous intéresse:

```
public static void sort(Object[] a)
```

Quelle type de transtypage effectue la méthode sort sur vos objets livres ?: .....
Affichez vos livres triés.
Est-ce un tri ascendant ou descendant sur les valeurs numériques (nbPages)?
.....

3 item: affichez dans l'ordre descendant

Pour les rapides et en autonomie, toujours avec la méthode sort, vous affichez vos livres triés dans le sens descendant. Sur un moteur, faites une recherche « arrays.sort reverse order java »

4 Item : collection de livres

Pour les rapides et en autonomie, utilisez une collection. Regardez s'il y a possibilité de réduire le nombre ligne avec la programmation fonctionnelle Java8.

5 item: un autre critère de tri

Pour les rapides et en autonomie, utilisez un objet Comparator pour trier avec un autre critère de tri + sens. Dans la documentation pdf présente sur le serveur habituel, étudiez les exemples sur l'interface Comparator. Faites le tri des livres sur leur titre. Si vous ajoutiez le prix des livres, vous pourriez trier sur le prix.

6 item: un tri multi-critères

Pour les rapides et en autonomie, utilisez un objet Comparator qui fasse un tri d'abord sur le titre du Livre (1er niveau) puis, au cas où les titres sont égaux, un tri sur le nombres de pages (2eme niveau).

7 Item : votre interface IBavard

Les utilisateurs du TP précédent parlent beaucoup. Ils vont pouvoir entrer dans la grande famille des bavards. Implémentez ce comportement. Dessinez-ici la représentation UML de l'interface : .....

8 Item : votre interface Nommable

Avec les facilités d'Eclipse, on vous demande d'extraire (extract) les méthodes getNom et setNom pour créer une nouvelle interface que l'on appellera Nommable. Sur Eclipse, menu <refactor <extract interface ...

9 item: Etudes des interfaces de Java8

Pour les rapides et à propos des interfaces, lisez l'url suivante :
http://thecodersbreakfast.net/index.php?post/2014/01/20/Java8-du-neuf-dans-les-interfaces
Citez toutes les nouvelles possibilités avec les interfaces Java8 :
.....

10 item: Les interfaces fonctionnelles de Java8

Pour les rapides et à propos des interfaces fonctionnelles de Java8, lisez le blog d'Ippon technologie. Essayez les exemples proposés.
http://blog.ippon.fr/2014/03/18/java-8-interfaces-fonctionnelles/

Comment distingue-t-on une interface fonctionnelle d'une interface classique ? :
.....