Ce TP porte sur la gestion d’objets planètes.   
On adopte la même règle de conception MVC.   
Mais on introduit l'utilisation des Collection(s) pour gérer plusieurs objets.   
La collection adoptée est HashSet.   
Cette structure est choisie parce qu'elle n'autorise pas la redondance d'objets (contrairement à la LinkedList ou ArrayList).   
De cette manière, on garantit que chaque planète existera une seule fois dans le système.   
  
La classe SolarSystem implémente le principe du singleton (un seul système existe dans le logiciel).   
  
On a aussi utilisé l'interface Comparator pour comparer les planètes.   
On a programmé trois comparateurs différents (chacun implémentant Comparator).   
  
La classe ComparateurPlaneteFactory permet de créer le bon comparateur en fonction du critère de l'utilisateur.   
  
Au moment du tri, le comparateur est utilisé pour créer un TreeSet. Le TreeSet est une structure de Set (donc sans redondance) qui est trié selon un comparateur qui définit l'ordre total de l'ensemble.   
On ajoute toutes les planètes au TreeSet pour obtenir un ensemble trié selon le critère que l'utilisateur veut.   
  
Il faut aussi signaler dans la vue VueAffichageCollectionPlanete qu'on reçoit un objet abstrait de type Collection.   
On n'a pas besoin de savoir son type concret (TreeSet ou HashSet ou autre comme les listes). On sait juste que c'est un objet Collection. Cela pour permettre d'utiliser cette vue quelque soit le type de collection.   
Pour afficher les planètes, on crée un itérateur sur la collection.   
On parcourt la collection à l'aide de l'itérateur et on affiche chaque planète.   
  
En résumé, il faut insister sur les points suivants :   
1- l'utilisation d'une structure de données (pré-programmée) pour stocker plusieurs objets.   
2- l'utilisation de HashSet pour éviter la redondance d'objets (ce qui est en accord avec la réalité des planètes).   
3- l'utilisation de plusieurs comparateurs qui implémente l’interface Comparator générique selon le critère   
4- l'utilisation du pattern Factory pour créer le bon comparateur   
5- utilisation de TreeSet (ensemble trié) créé à l'aide du comparateur pour trier les planètes   
6- l'utilisation dans la vue de l'interface Collection qui permet de recevoir un objet abstrait sans connaissance de sa classe concrète pour grantir la réutilisabilité de la vue   
7- l'utilisation du pattern Itertateur générique pour parcourir la collection   
  
  
PS. comme d'habitude, il faut exposer le sujet, faire le diagramme, et laisser programmer progressivement.