TD2 Java

# Exercice 1. Redéfinition

1. Le main affiche : 42 42 24 24 24 24 car la méthode meth est redéfinie dans la fonction Fille. Et dans le dernier exemple, c’est un squelette de type mère mais qui implemente quand même un objet de type fille.  
   Ainsi elle affiche 42 quand l’objet est un objet Mère et 24 quand c’est un objet Fille.
2. S’il est dans la classe Fille, un objet de type Fille a accès à 2 méthodes meth(), celle de la classe Fille et celle de la classe Mère héritée s’il est dans le main il a accès à seulement la méthode meth() de la classe Fille par défaut;
3. Si les méthodes meth sont statiques, le main affichera : 42 42 24 24 42 42 parce qu’on ne peut pas appeler une instance de la classe, il faut définir la méthode sur la classe elle-même. Il faudra donc pour le troisième exemple écrire Mere.meth ;
4. Si les meth sont des champs, le main affichera : 42 42 24 24 42 42 parce que le champ meth de la classe mère est protected, ainsi la troisième instance qui a un squelette de classe Mère accèdera au champ de la classe Mère.

# Exercice 2 : Redéfinition – Surcharges

1. - Il y a une erreur avec la méthode e, elle est définie comme private dans la classe Fille alors qu’elle est définie comme protected dans la classe Mère.  
   - Il y a une erreur avec la fonction char h dans la classe Fille, elle retourne un char alors qu’elle est définie dans la classe Mère comme étant une méthode de type int.  
   - La méthode i de la classe Fille retourne un nombre alors qu’elle ne devrait rien retourner sachant qu’elle a été définie comme étant void dans la classe Mère.  
   - La méthode k lance une exception de type Exception dans la classe Fille alors que dans la classe Mère elle est de type IOException, ça ne fonctionne pas car IOException est une sous-classe de Exception .  
   -mere et mereFille ne peuvent compiler une méthode miage car elle est définie seulement dans la classe Fille et l’héritage ne fonctionne que dans un sens.
2. Une redéfinition c’est une méthode qui a le même nom et les mêmes paramètres (même profil) que la classe héritée. Une méthode surchargée c'est une méthode avec le même nom mais des paramètres différents (profil différents).  
   Les surcharges : void c, void d (Mère , Mère),   
   Les redéfinitions : void a , void b,void d()  
     
     
     
   Exercice 3 Expressions arithmétiques