**Exercice 1**

Après analyse des morceaux de code ci-dessous, remplissez le tableau ci-joint en mettant une croix (X) dans les cases correspondantes aux visibilités des attributs (a, b, c, d) dans les différentes classes.

| | |
|---|---|
| package p1;<br>class Clas1{<br>int a;<br>public int b;<br>private int c;<br>protected int d;<br>} | package p1;<br>class Clas2 extends Clas1{<br>} |
| package p1;<br>class Clas3 extends Clas1{<br>} | package p2;<br>class Clas4 extends Clas1{<br>} |
| package p2;<br>class Clas5{<br>} | class Clas6 extends Clas4{<br>} |

| Visibilité dans | Clas2 | Clas3 | Clas4 | Clas5 | Clas6 |
|---|---|---|---|---|---|
| a | X | X | | | |
| b | X | X | X | X | X |
| c | | | | | |
| d | X | X | X | | X |

**Exercice 2**

Répondez par Vrai ou par Faux aux questions suivantes :

Q1 Une classe ne peut être déclarée *abstract* et *final* **V**

Oui, car une classe finale ne peut être héritée. Hors on doit héritée d'une classe abstraitre : soit pour redéfinir les méthodes abstraitre , soit s'il n'y en a pas pour la rendre contrète.

Q2.1 Une méthode déclarée *final* est une méthode qui ne peut pas être surchargée **F**

**Une méthode surchargé n'a pas la meme signature on peut changer portée et finalité.**

Q2.2 Une méthode déclarée *final* est une méthode qui ne peut pas être redéfinie **V**

Oui C'est la la définition d'une méthode finale.

Q3.1 Il est possible de surcharger une méthode déclarée public en la déclarant private **V**

**Une méthode surchargé n'a pas la meme signature on peut changer portée et finalité.**

Q3.2 Il est possible de redéfinir une méthode déclarée public en la déclarant private **F**

Oui : on ne peut reduire la visibilité d'une methode redéfinie.

Q4 Une méthode déclarée private ne peut être déclarée abstract **V**

Oui contre sens : une methode private ne peut être redéfinie / une méthode abstract doit être redéfinie

Q5 Les méthodes d'une interface doivent être explicitement déclarées abstract et public **F**

Non : elles le sont implicitement. Il n'est donc pas « obligatoire » de le faire.

Q6 Les données membres déclarées dans le corps d'une interface sont implicitement public final et static **V**

Implicitement. Inutile de les déclarer public/final/statique (mais pas faux).

Q7 Un bloc try contient un ensemble d'instruction susceptibles de lever des exceptions durant leur exécution **V**

Il n'y a aucune utilité à mettre ces instructions dans un bloc try.