



### **Exercice 1 : QCM (5pts)**

**Chaque bonne réponse est notée sur 1 point en cas de mauvaise réponse –0,5pt**

- Que signifie le mot clef "static" associé à un attribut ?
  1. Que la valeur de cet attribut est constante
  2. Que cet attribut n'est visible que dans la classe où il est défini
  3. Que cet attribut sera toujours passé par valeur
  4. Que cet attribut a une valeur unique pour toutes les instances de la classe
- Au sujet du mot clef "abstract", quelle assertion est fausse ?
  1. Une classe abstraite ne peut être instanciée
  2. Une méthode abstraite n'a pas d'implémentation
  3. Une classe abstraite n'a pas forcément de classe fille
  4. Une classe abstraite doit contenir au moins une méthode abstraite
- Une interface en Java peut contenir :
  1. Des déclarations de méthodes et de constantes
  2. Des déclarations de méthodes
  3. Des déclarations de méthodes et de méthodes statiques
  4. Des déclarations de méthodes et de constructeurs
- Quel est le droit d'accès le plus restrictif pour donner l'accès aux membres d'une autre classe de même package ?
  1. public
  2. default access

3. abstract
4. protected

- Dans quelle situation utilise-t-on la séquence : try-catch-finally ?

1. Pour traiter les exceptions
2. Pour sortir du programme
3. Pour déboguer le code

- Peut-on utiliser plusieurs blocs catch après un bloc try ?

1. Vrai
2. Faux

- Quelle instruction faut-il utiliser pour créer un nouveau tableau d'entiers de type int de 5 cases ?

1. `Array t = new Array ("int", 5);`
2. `int t = new int[5];`
3. `int[] t = new int[5];`
4. `int[] t = new int[4];`
5. `Array t = new int[5];`

## **Exercice 2: Classes et méthodes (6points)**

1. Définir une classe **Voiture** avec les attributs suivants : *Id*, *Marque*, *Vitesse*, *Puissance*.
2. Définir un **constructeur** permettant d'initialiser les attributs d'un objet voiture par des valeurs passées en paramètre. Sachant que l'Id doit être auto-incrément.
3. Définir les accesseurs aux différents attributs de la classe.
4. Définir la méthode **toString ( )** permettant d'afficher les informations d'une voiture.
5. Écrire un programme testant la classe Voiture.

## **Exercice 3 : Héritage en Java (9pts)**

1. Créez une classe « Personne » avec l'âge comme attribut.
2. Créez une classe « Elève » et une autre classe « Enseignant », les deux héritent de la classe « Personne ».
3. La classe « Elève » aura une méthode publique « Partir\_en\_classe », qui affichera à l'écran « Je pars en classe. ».

4. La classe « Enseignant » aura une méthode publique « Expliquer », qui affichera à l'écran « L'explication commence ». En plus, il aura un attribut privé « Sujet » de type string.
5. La classe « Personne » doit avoir une méthode « SetAge (int n) » qui indiquera la valeur de leur âge (par exemple, 15 ans).
6. La classe « Elève » aura une méthode publique « Affichage\_Age » qui écrira sur l'écran « Mon Age es: XX ans».
7. Vous devez créer une autre classe de test appelée « Test » qui contiendra « Main » et:  
Créez un objet Personne et faites-lui dire « Salut »
8. Créez un objet Elève, définir son âge à 15 ans, faites-lui dire « Salut », « Je pars en classe. » et afficher son âge.
9. Créez un objet Enseignant, 40 ans, demandez-lui de dire « Salut » puis commence l'explication.