**Développement d’application - Bonnes pratiques**

**QCM - Modules 1 à 6**

Nom:

Prénom:

***Le QCM doit être envoyé sur l’adresse pierre.deruel@envoxoo.com***

**1/ Un objet représente un type de donnée?**

* Simple
* Complexe

**2/ La zone mémoire qu’on appelle “le tas” (heap en anglais) contient-elle la référence mémoire de l’objet auquel on veut accéder ?**

* Oui
* Non

**3/ Quelles sont les contraintes pour créer un constructeur?**

* Il doit avoir un type de retour
* Il doit avoir le même nom que la classe
* Il peut être déclaré static
* On ne peut pas lui déclarer des paramètres
* Il ne doit pas avoir de type de retour

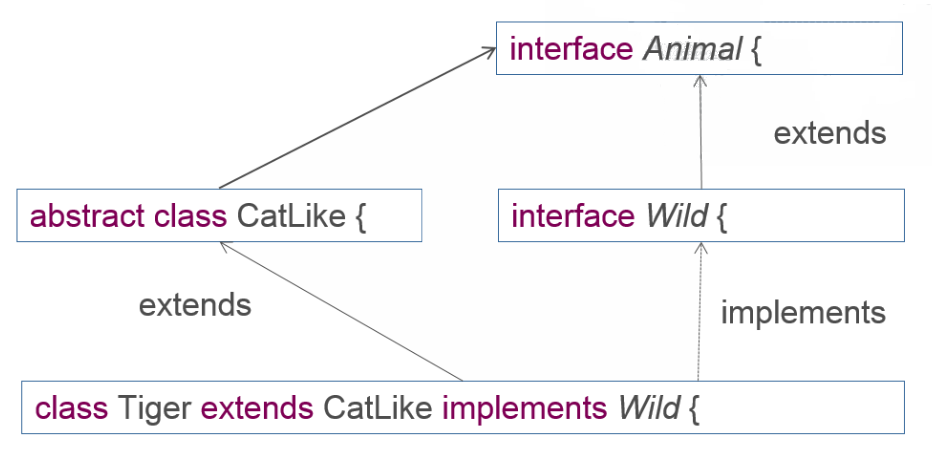
**4/ Combien existe-t-il de de niveaux de visibilité en Java?**

* aucun
* 2
* 4
* 5

**5/ On souhaite utiliser une collection ayant de bonnes performances lors de l’ajout/suppression d’un élément. Quelle implémentation doit-on choisir?**

* ArrayList<E>
* Vector<E>
* Set<E>
* LinkedList<E>

**6/ A partir du schéma ci-dessous, quelles affirmations sont correctes?**

****

* La classe CatLike peut être instanciée
* La classe Tiger peut être instanciée
* La classe CatLike possède le comportement Wild
* La classe Tiger possède le comportement Animal

**7/ Quelle est l’intérêt d’une interface?**

* Il n’y en a pas
* Elle permet de créer un comportement générique
* Elle permet d’améliorer les performances

**8/ On utilise un diagramme de cas d’utilisation en fin de projet?**

* Oui
* Non
* Ce diagramme n’existe pas

**9/ Un client lourd s’exécute dans un navigateur?**

* oui
* non

**10/ En UML, la généralisation peut-elle être comparée à l’héritage en Java?**

* Oui
* Non

**11/ Sélectionnez le(s) pattern(s) de création**

* Observer
* Factory
* Facade
* Singleton

**12/ En Java, peut-on hériter de plusieurs classes?**

* Oui
* Non

**13/ En Java, peut-on implémenter plusieurs interfaces?**

* Oui
* Non

**14/ La déclaration de méthode ci-dessous est-elle valide?**

freiner(){

}

* Oui
* Non

**15/ La déclaration de la constante ci-dessous est-elle valide?**

**public static final int *MA\_CONSTANTE*** = 1;

* Oui
* Non

**16/ En quelques lignes, expliquez ce qu’implique la déclaration d’une variable sans modificateur de visibilité (en Java)**

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

**17/ En quelques lignes, expliquez la notion de surcharge en POO**

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

**18/ En quelques lignes, expliquez la notion d’héritage en POO**

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

**19/ En quelques lignes, expliquez la différence entre gestion de projet classique et gestion de projet orientée agile**

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

**20/ En quelques lignes, expliquez à quoi sert un diagramme de cas d’utilisation**

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………