



Etablissement Inter – Etats d'Enseignement Supérieur
CENTRE D'EXCELLENCE TECHNOLOGIQUE PAUL BIYA
BP: 13719 Yaoundé (Cameroun) Tel. (+237) 242 72 99 57/(+237) 242 72 99 58
Site web: www.lalcameroun.com E-mail: contact@lalcameroun.com

EXAMEN DE FIN DE SEMESTRE

Session de Février 2024

Epreuve de Programmation Orientée Objet

Durée : 2h Niveau 2 Filières : Génie Logiciel, Syst. et Réseaux, Software Eng. Année Aca. : 2023-2024

Evaluation des Connaissances (10pts)

Exercice 1 : QCM (0.5 * 5 = 2.5pts)

1. Java est un langage :

- a. Compiler
- b. Interprété
- c. Compiler et interprété
- d. Ni compiler ni interprété

2. Java est un langage développé par

- a. Hewlett-Packard
- b. Sun Microsystems
- c. Microsoft
- d. Oracle

3. Combien d'instances de la classe A, crée le code suivant ?

```
A x,u,v;  
x = new A();  
A y = x;  
A z = new A();
```

- a. Aucune
- b. Cinq
- c. Trois
- d. Deux

4. Laquelle des propositions suivantes est juste ?

- a. Une classe peut implémenter plusieurs interfaces mais doit étendre une seule classe;
- b. Une classe peut implémenter plusieurs classes mais doit étendre une seule

interface;

- c. Une classe peut implémenter plusieurs classes et peut étendre plusieurs interfaces;
- d. Une classe doit implémenter une seule interface et étendre une seule classe

5. Pour la classe C définie comme suit :

```
public class C {  
    public static int i;  
    public int j;  
    public C() {  
        i++;  
        j=i;  
    }  
    public static void main(String[]  
args) {  
        C x=new C();  
        C y=new C();  
        C z= x;  
        System.out.println(z.i + " et  
" + z.j);  
    }  
}
```

Qu'affichera le code ci-dessus :

- a. 2 et 2
- b. 1 et 1
- c. 2 et 1
- d. 1 et 3

Exercice 2 : QUESTION REPONSE (7.5pts)

1. Définir les termes suivants : Variable local, variable d'instance, variable statique. (1.5pt)
2. Citez et expliquez les rôles des modificateurs d'accès. (2pts)
3. Que sont le rôle des modificateurs de non accès suivants : final, abstract et synchronized. (1.5pt)
4. Quelle est la différence entre surcharge et redéfinition des méthodes ? (1pt)
5. Que se passe-t-il s'il y a plusieurs méthodes main() dans une classe en Java ? (0.5pt)
6. Est-il possible que le bloc « finally » ne soit pas exécuté ? (0.5pt)

Evaluation des compétences (10pts)

Exercice 1 : (2.5Pts)

1. Définir une classe *Salle* avec les attributs suivants : *id*, *code* et *libelle*. (0.5pt)
2. Définir les accesseurs aux différents attributs de la classe. (0.5pt)
3. Définir un constructeur permettant d'initialiser les attributs d'un objet *salle* par des valeurs saisies par l'utilisateur. Sachant que *Id* doit être auto-incrément. (0.5pt)
4. Définir la méthode *toString()* permettant d'afficher les informations de la salle en cours. (0.5pt)
5. Redéfinir la méthode *equals(Object O)* qui permet de comparer les valeurs de 2 Salles. (0.5pt)

Exercice 2 : (2,5 Pts)

Créer l'interface générique *FonctionsUtiles<T>* et la compléter avec les méthodes abstraites ci-dessous :

1. *boolean create (T Obj)* : Méthode permettant d'ajouter un objet *Obj* de type *T*. (0.5pt)
2. *boolean delete (T Obj)* : Méthode permettant de supprimer un objet *Obj* de type *T*. (0.5pt)
3. *boolean update (T Obj)* : Méthode permettant de modifier un objet *Obj* de type *T*. (0.5pt)
4. *T findById (int id)* : Méthode permettant de renvoyer un objet dont *id* est passé en paramètre. (0.5pt)
5. *List <T> findAll ()* : Méthode permettant de renvoyer la liste des objets de type *T*. (0.5pt)

Exercice 3 : (2.75Pts)

Créer la classe *SalleService* qui implémente cette interface *FonctionsUtiles*. Dans cette classe les données seront stockées dans une collection de type *List*, c'est-à-dire qu'elle :

1. A une donnée membre *private List<Salle> salles*; (0.25pt)
2. Redéfinit les méthodes de l'interface qu'elle implémente. (5*0.5 = 2.5 pts)

Exercice 4 : (2.25Pts)

Dans une classe de test *GestionSalle* : Créer cinq salles, Afficher la liste des salles, Supprimer une salle, Modifier les informations d'une salle, Afficher la liste des salles.