

Corrigé type Examen de rattrapage : programmation orientée objet

Documents non autorisés.

Exercice 1 (05 points) :

Pour chacune de ces déclarations, indiquez si elle est vraie ou fausse:

1. La méthode equals() :

- a) Faux 0,5 ptt
- b) Vrai 0,25 ptt
- d) Faux 0,25 ptt

2. Une classe abstraite est une classe qui:

- a) faux 0,25 ptt
- b) Faux 0,25 ptt
- c) Vrai 0,25 ptt
- d) Faux 0,25 ptt

3. Les chaînes de caractères en Java :

- a) Vrai 0,25 ptt
- b) Faux 0,25 ptt
- c) Faux 0,25 ptt
- d) Vrai 0,25 ptt

4. Les méthodes statiques en Java :

- a)Vrai 0,25 ptt
- a) Vrai 0,25 ptt
- b) Vrai0,25 ptt
- c) Vrai0,25 ptt

5. En java, quand deux méthodes ont la même signature:

- a) Faux 0,25 ptt
- b) Vrai 0,25 ptt
- c) Faux 0,25 ptt
- d) Faux 0,25 ptt

Exercice 2 : (05 points) :

I. Corrigez les erreurs et donnez le résultat produit par ce programme :

```
Les erreurs et la correction :
private int val =2;// private static int val =2; 0,25 ptt
    A objet3 = new A (1 ) ; 0,25 ptt // A
objet3 = new A ( ) ; 0,25 ptt
    System.out.println(A.val); 0,25 ptt //
System.out.println(objet3.val);0,25 ptt

Affichage :
2,3,3,3,3,0,4,-1,5,0,6,-1,7,0 0,25 ptt
FAUX 0,5 ptt
```

II. Considérer le code suivant :

Triangle T1 = new Triangle (10,6,9);

Triangle T2 = T1;

Triangle T3 = new Triangle(10, 6, 9);

1. Quel serait le résultat de la comparaison T1 == T2?

true 1 ptt

2. Quel serait le résultat de la comparaison T1 == T3?

false 1 ptt

3. Quel serait le résultat de la comparaison T2 == T3?

false 1 ptt.

Exercice 3 : (10 points)

I.

```
public class Complexe {
    private float a; // Partie réelle 0,25 ptt
    private float b; // Partie imaginaire 0,25 ptt
    private static int compteur = 0; // Compteur
de nombres complexes créés 0,5 ptt
```

1. // Constructeur pour initialiser les parties réelle et imaginaire

```
public Complexe(float a, float b) 0,5 ptt{
    this.a = a; 0,25 ptt
    this.b = b;0,25 ptt
    compteur++;0,5 ptt
```

```

    }

2. // Méthode pour afficher le nombre
   complexe
public void afficher() {
    System.out.println(a + " + i*" + b);
}0,5 ptt

3. // Méthode pour compter le nombre de
   nombres complexes créés
public static int compter() {
    return compteur; 1 ptt
}

4. // Méthode statique pour fournir la
   somme de deux nombres complexes
public static Complexe
somme(Complexe c1, Complexe c2) 0,5 ptt{
    return new Complexe(c1.a + c2.a,
c1.b + c2.b); 1ptt
}

5. // Méthode pour calculer le module du
   nombre complexe
public double module() 0,5 ptt {
    return Math.sqrt(a * a + b * b);
}1ptt

6. // Méthode pour comparer deux nombres
   complexes
public static boolean
comparer(Complexe c1, Complexe c2) 0,5 ptt
{
    return c1.a == c2.a && c1.b == c2.b;
}0,5 ptt
}

```

II.

```

public class TestComplexe 2 ptt sur tout {
    public static void main(String[] args) {
        // Créer deux nombres complexes
        Complexe c1 = new Complexe(3.0f,
4.0f);

```

```

        Complexe c2 = new Complexe(1.0f,
2.0f);

        // Afficher les nombres complexes
        System.out.print("c1: ");
        c1.afficher();
        System.out.print("c2: ");
        c2.afficher();

        // Compter le nombre de nombres
        complexes créés
        System.out.println("Nombre de
nombres complexes créés: " +
Complexe.compter());

        // Calculer et afficher la somme de c1 et
        c2
        Complexe somme =
Complexe.somme(c1, c2);
        System.out.print("Somme de c1 et c2:
");
        somme.afficher();

        // Calculer et afficher le module de c1 et
        c2
        System.out.println("Module de c1: " +
c1.module());
        System.out.println("Module de c2: " +
c2.module());
        // Comparer c1 et c2
        System.out.println("c1 est égal à c2: " +
Complexe.comparer(c1, c2));
        // Créer un autre nombre complexe
        pour la comparaison
        Complexe c3 = new Complexe(3.0f,
4.0f);
        System.out.print("c3: ");
        c3.afficher();
        // Comparer c1 et c3
        System.out.println("c1 est égal à c3: " +
Complexe.comparer(c1, c3));
    }
}

```