

Ne rien inscrire dans ce cadre

Prénom

Nom

Promotion

Groupe

Promotion : L3 LSI

Module : Programmation en JAVA

ALSI54

DE - 1h45 min

Horaire : 10H20-12H10

Examen sur table

Sujet rédigé par : **Imen Cheikh Rouhou**

Supports autorisés :

Calculatrice autorisée : non

Documents autorisés : non

Traducteur électronique : non

Dictionnaire : non

Consigne :

Merci de restituer : **votre copie quadrillée**

Rappel :

- Tous les appareils électroniques (téléphones portables, ordinateurs, tablettes, montres connectées, accès à internet ...) doivent être éteints et rangés.
- Il est interdit de communiquer.
- Toute fraude ou tentative de fraude fera l'objet d'un rapport de la part du surveillant et sera sanctionnée par la note zéro, assortie d'une convocation devant le conseil de discipline. Aucune contestation ne sera possible. Tous les documents et supports utilisés frauduleusement devront être remis au surveillant.
- Aucune sortie de la salle d'examen ne sera autorisée avant la moitié de la durée de l'épreuve.

- 1) Complétez le tableau suivant pour illustrer la visibilité des modificateurs (*reproduisez le tableau sur votre feuille*):

Modificateur	Classe	Classe-fille	Package	Autre
private	✓			
protected				
public				
néant		✓		

- 2) Quelle est la différence entre la redéfinition (*overriding*) et la surcharge (*overloading*)?
- 3) Qu'est-ce qu'une classe abstraite?
- 4) Qu'est-ce qu'une interface?
- 5) Indiquez deux cas d'utilisation du mot clé *this*
- 6) Indiquez deux cas d'utilisation du mot clé *super*
- 7) Une classe peut-elle hériter de plusieurs classes? Implémenter plusieurs interfaces?
- 8) Quelle différence existe-t-il entre la comparaison avec `==` et avec `equals()` ?
- 9) En Java le développeur peut-il détruire lui-même les objets?
- 10) Quels sont les impacts de l'utilisation du mot *static* sur une méthode?
- 11) Que se passe-t-il lorsqu'on exécute ce code ?

```

1. int Output = 10;
2. boolean b1 = false;
3. if((b1 == true) && ((Output += 10) == 20))
4. {
5.     System.out.println("Nous sommes égaux " + Output);
6. }
7. else
8. {
9.     System.out.println("Pas égaux! " + Output);}

```

- a) Erreur de compilation à la ligne 3.
- b) Compilation OK et affichage de "Nous sommes égaux 10".
- c) Compilation OK et affichage de "Pas égaux ! 20".
- d) Compilation OK et affichage de "Pas égaux! 10".

- 12) Soit :

```

1. class X { void do1() { } }
2. class Y extends X { void do2() { } }
3.
4. class Chrome {
5.     public static void main(String [] args) {
6.         X x1 = new X();
7.         X x2 = new Y();
8.         Y y1 = new Y();
9.         // insérez code ici
10. } }

```

Laquelle(s) de ces instructions peut-on insérer à la ligne 9 pour que le code compile ? **Merci d'expliquer votre réponse!**

- a) x2.do2();
- b) ((Y)x2).do2();
- c) Aucune des instructions ci-dessus ne permet une compilation.

13) Soit :

```
1. public class Outer
2. {
3.     public int a = 1;
4.     private int b = 2;
5.     public void method(final int c)
6.     {
7.         class Inner
8.         {
9.             private void iMethod(int d)
10.            {
11.
12.            }
13.        }
14.    }
15. }
```

A la ligne 11, quelle(s) variable(s) peut-on référencer de façon licite (sans erreur de compilation) ? **Merci d'expliquer votre réponse!**

- a) a
- b) b
- c) c
- d) d
- e) Aucune des variables ci-dessus
- f) Toutes les variables ci-dessus

14) A l'exécution, qu'affiche ce code ?

```
class Value{
    public int i = 15;}
public class Test{
    public static void main(String argv[]){
        Test t = new Test();
        t.first(); }
    public void first() {
        int i = 5;
        Value v = new Value();
        v.i = 25;
        second(v, i);}
    public void second(Value v, int i){
        i = 0;
        v.i = 20;
        Value val = new Value();
        v = val;
        System.out.println(v.i + " " + i);}}
```

- a) 15 0
- b) 20 0
- c) 0 15
- d) Aucun des résultats ci-dessus

15) Soit le code suivant :

```
1. class EFREI
2. {
3.     void maMethode() {}
4. }
5.
6. class L3A extends EFREI
7. {
8.     void maMethode() {}
9. }
```

Si le code reste tel quel, quel(s) modificateur(s) (public, protected ou private) peut-on légalement placer devant maMethode() la ligne 3?

Et suite à votre changement à la ligne 3, quel(s) modificateur(s) peut être légalement placé avant maMethode à la ligne 8?

Merci d'expliquer votre réponse!

- a) private ou rien (i.e. il reste tel quel) à la ligne 3. Et rien (i.e. laisser tel quel) ou protected ou public sur la ligne 8.
- b) public ou protected à la ligne 3. Et private ou alors rien du tout (i.e. laisser tel quel) sur la ligne 8.
- c) rien (i.e. laisser tel quel) ou protected ou public à la ligne 3. Et private ou rien du tout (i.e. laisser tel quel) sur la ligne 8.
- d) Aucun des cas de figure ci-dessus.

16) Soit le code suivant :

```
class Personne {
    Personne(String s, int i) {
        ++pid;
        nom = s;
        age = i;
    }
    static int pid;
    int age;
    String nom;
}

class Test {
    public static void main(String args[]) {
        Personne p1 = new Personne("Jacques", 22);
        Test te = new Test();
        Personne p2 = te.changer(p1);
        System.out.println(p2.pid + " " + p2.nom + " " + p2.age);
        System.out.print(p1.pid + " " + p1.nom + " " + p1.age);
    }

    private Personne changer(Object o) {
        Personne p2 = (Personne) o;
        p2.age = 25;
        return p2;
    }
}
```

Quel est le résultat attendu?

- a) Affichage de
1 Jacques 25
1 Jacques 25
- b) Affichage de
1 Jacques 25
1 Jacques 22
- c) Affichage de
1 Jacques 22
1 Jacques 22
- d) Lancement de ClassCastException
- e) Erreur de compilation