

Exercice 1

- 1) Parmi les méthodes de la classe suivante, lesquelles peuvent être statiques et lesquelles ne peuvent en aucun cas être statiques ?

```
class VerificationStatic{
    int x, y;
    String nom;
    static float g ;

    VerificationStatic (String n, int a, int b){
        nom=n;
        x=a;
        y=b;
    }
    void afficher(){
        system.out.println(nom + " " + x + " " + y);
    }
    void ajouter(VerificationStatic obj){
        x = x + obj.x;
        y = y + obj.y;
        nom = nom + obj.nom;
        g=7 ;
    }
    VerificationStatic nouveau(int n){
        VerificationStatic res = new VerificationStatic ("aaa",3,5);
        res.x = n+g;
        res.y = n*2;
        res.nom = "Auto_" +n;
        return res;
    }
    boolean plusGrand(VerificationStatic obj){
        if (obj.x == x){
            return y>obj.y;
        } else{
            return x>obj.x;
        }
    }
    boolean comparer(VerificationStatic obj1, VerificationStatic obj2){
        if (obj1.x == obj2.x){
            return obj1.y>obj2.y;
        }else{
            return obj1.x>obj2.x;
        }
    }
}
```

- 2) Modifier le programme en insérant le mot static devant les méthodes appropriées. Ecrire la méthode main permettant de :
- Créer deux objets v1 et v2 avec des valeurs de votre choix
 - Appeler les méthodes `VerificationStatic`, `plusGrand`
 - Appeler les autres méthodes de la classe

Exercice 2

Quelles sont les instructions qui provoquent des erreurs de compilation ? (indiquer la cause de l'erreur en commentaire)

```
1. class Essai{.....
2. int y, x; .....
3. final static float z=3.0 ; .....
4. int m1(){.....
5. int f ; .....
6. return f+z; .....
7. }
8. static void m2(final int h){.....
9. h=5; .....
10. System.out.println(h+x); .....
11.}
12.public static void main(String [] a){ .....
13.Essai e=new Essai();.....
14.e.m2();.....
15.m1();.....
16.x=7; .....
17.}
```

Application 2

Soit la classe suivante :

```
package packA ;
public class A {
public int i ;
private int a;
protected int b;
int c;
public void afficher() { System.out.println(i+a+b); }
}
```

Indiquer si les classes suivantes compilent. Indiquer les erreurs en cas de non compilation.

```
package packA ;
public class B extends A {
public void traiter() { System.out.println(i+a+b+c); }
}
```

```
.....
.....
package packA ;
public class E {
public B b = new B();
public void traiter() { b.afficher();}
}
```

```
.....
.....
package packD ;
import packA.A;
public class D {
public A k = new A();
public void traiter() { System.out.println(k.c + k.i + k.b); }
}
```

```
.....
.....
package packF;
import packA.A;
```

```
public class F extends B{
public void traiter() { System.out.println(i+b+c); }
}
```

```
.....
.....
package packF;
import packA.A;
public class F extends A{
public void traiter() {
System.out.println(i+b);
A k = new A();
System.out.println(k.i+k.b);
}
}
```

Correction
Exercice 1