

Exercice 1

Qu'affiche le code suivant ? Justifiez votre réponse.

```
class A {
    void f(A o) {
        System.out.println("f(A) dans A");
    }
}
class B extends A {
    void f(A o) {
        System.out.println("f(A) dans B");
    }
    void f(B o) {
        System.out.println("f(B) dans B");
    }
}
class exoAlgoSelect1 {
    public static void main(String[] args) {
        A a = new A();
        A ab = new B();
        B b = new B();
        a.f(a);
        a.f(ab);
        a.f(b);
        ab.f(a);
        ab.f(ab);
        ab.f(b);
        b.f(a);
        b.f(ab);
        b.f(b);
    }
}
```

Exercice 2

On considère l'interface Date

```
public interface Date {
    public int jour();
    public int mois();
    public int annee();
}
```

Question 1 Proposer une classe DateImpl qui implémente l'interface Date.

Exemple de code de test :

```
Date d = new DateImpl(25, 12, 2017);
System.out.println(d.jour());
System.out.println(d.mois());
System.out.println(d.annee());
```

affiche

```
25
12
2017
```

Question 2 Rajouter à la classe `DateImpl` une méthode pour que les instructions :

```
Date d = new DateImpl(14, 7, 1789);  
System.out.println(d);
```

affichent

```
14/7/1789
```

Question 3 Redéfinir la méthode `equals` héritée de la classe `java.lang.Object` pour que deux dates soient égales si elles ont les mêmes jour, mois et année. Quelle autre méthode héritée de la classe `java.lang.Object` est-il conseillé de redéfinir? Proposer une implémentation de cette méthode.

Question 4 Modifier le constructeur de `DateImpl` pour qu'il lève une exception `DateInvalideException` si les paramètres ne sont pas valides : mois non compris entre 1 et 12, jour non compris entre 1 et 31. Donner également le code de la classe `DateInvalideException`.

Exercice 3

On considère les classes `Voiture` et `Avion`.

La méthode `int nbPassagers()` peut être appelée sur toute instance de ces deux classes.

La méthode `int nbPortes()` peut être appelée sur toute instance de la classe `Voiture`.

La méthode `int nbReacteurs()` peut être appelée sur toute instance de la classe `Avion`.

Exemple de code de test :

```
Avion a380 = new Avion(600, 4);  
System.out.println(a380.nbPassagers()); // affiche 600
```

Écrire le code de ces deux classes. Veillez à factoriser votre code.