

associé à `x` entraîne automatiquement une modification de celle associé à `y` et réciproquement.

4.2.2 L'égalité

Si `a` et `b` sont deux expressions de type `Point`, leur valeur est une référence et l'expression `a == b` vaut `true` uniquement quand ces deux références sont identiques. C'est-à-dire quand `a` et `b` partagent une même cellule. On appelle cette relation l'égalité *physique*. Ainsi, le programme

```
Point a = new Point(48.715,2.208,156.0);
Point b = new Point(48.715,2.208,156.0);
if (a == b) System.out.println("égal");
else System.out.println("différent");
```

affiche `différent`.

Il est cependant possible d'écrire une fonction qui teste l'égalité *structurelle* de deux enregistrements, c'est-à-dire l'égalité champ à champ

```
static boolean equal (final Point x, final Point y) {
    return (x.latitude == y.latitude)
        && (x.longitude == y.longitude)
        && (x.altitude == y.altitude);}

```

et l'instruction

```
if (equal(a,b)) System.out.println("égal");
else System.out.println("différent");
```

affiche `égal`.

4.2.3 Les types enveloppés

Un type *enveloppé* est un type enregistrement à un seul champ. Un exemple est le type `Integer`

```
class Integer {
    int c;

    Integer (int x) {
        this.c = x;}}

```