

Exercice 4.2

Dans quel état s'exécute le corps de la fonction `swap` si `a` et `b` sont finales, mais `x` et `y` mutables? Et si `a` et `b` sont mutables, mais `x` et `y` finales?

Exercice 4.3

Qu'affiche le programme ci-dessus, si l'on remplace la fonction `swap` par la fonction `swap1` définie de la manière suivante?

```
static void swap1 (Integer x, Integer y) {  
    Integer z = x;  
    x = y;  
    y = z;}
```

4.3 Caml

Voyons maintenant les constructions qui permettent, en Caml, de définir un type, d'allouer une cellule, d'accéder à un champ et d'affecter un champ.

4.3.1 La définition d'un type enregistrement

En Caml, comme en Java, on définit un type enregistrement en indiquant l'étiquette et le type de chacun de ses champs

```
type point = {  
    latitude : float;  
    longitude : float;  
    altitude : float;}
```

4.3.2 La construction d'un enregistrement

On construit une valeur en donnant chacun de ses champs {latitude = 90.0; longitude = 0.0; altitude = 0.0;}. Il y a pas de mot clé `new` et c'est la paire d'accolades qui alloue la cellule.

La déclaration `let x = ref {latitude = 90.0; longitude = 0.0; altitude = 0.0;}` crée l'état