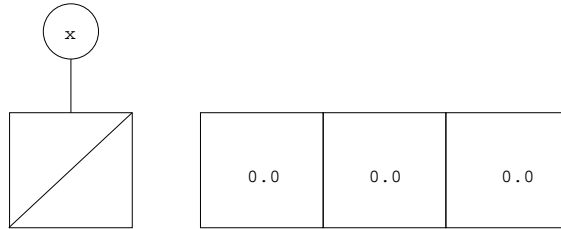


L'évaluation de l'expression `new Point()` crée une nouvelle référence `r'` et associe cette référence à un enregistrement, par défaut `{latitude = 0.0, longitude = 0.0, altitude = 0.0}`. La valeur de cette expression est la référence `r'`.



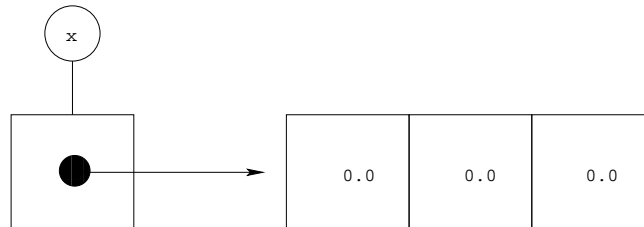
Dans ce dessin, l'objet de droite est une case unique, qui peut contenir trois nombres à virgule, et non une collection de trois cases. Autrement dit, il y a une seule référence `r'`.

Une référence qui a été ajoutée à la mémoire par la construction `new` s'appelle une *cellule*. L'ensemble des cellules de la mémoire s'appelle le *tas*. L'opération consistant à ajouter une nouvelle cellule à la mémoire s'appelle l'*allocation* d'une cellule.

Quand on exécute l'instruction

```
x = new Point();
```

on associe la référence `r'` à la référence `r` dans la mémoire. L'environnement est alors `[x = r]` et la mémoire `[r = r', r' = {latitude = 0.0, longitude = 0.0, altitude = 0.0}]`.



Ainsi, en Java, comme en Caml et en C, les références sont des valeurs et il est possible pour une référence d'être associée dans la mémoire à une autre référence. Mais contrairement à Caml et C, cela ne peut se faire qu'en utilisant cette construction des enregistrements.

Il est, bien entendu, possible de déclarer la variable `x` en lui donnant une valeur initiale. Ainsi, au lieu d'écrire

```
Point x;  
x = new Point();
```

on peut écrire