

4.5.1 Les types tableaux

En Java, un tableau dont les éléments sont de type `T` est de type `T []`.

4.5.2 L'allocation d'un tableau

On peut, par exemple, donner le type `int []` à une variable.

```
int [] t;
```

Comme pour une toute déclaration de variable, cela ajoute un couple à l'environnement associant une référence `r` à la variable `t` et un couple à la mémoire associant une valeur à la référence `r`. Si l'on déclare cette variable sans lui donner de valeur, la valeur par défaut est la valeur `null`. En Java, la référence `r` n'est jamais associée directement à un tableau de valeurs dans la mémoire. La case `r` est toujours une petite case qui ne peut contenir que `null` ou une autre référence.

Pour associer un tableau à la variable `t`, il faut donc commencer par créer une case suffisamment grande pour contenir plusieurs entiers. Cela se fait avec l'instruction `new`

```
new int [10]
```

L'évaluation de l'expression `new int [u]`, où `u` est une expression dont la valeur est un entier `n`, crée une nouvelle référence `r'` et associe cette référence à un `n`-uplet ne contenant que des valeurs par défaut : 0 dans ce cas. Les champs sont numérotés de 0 à `n - 1`.

L'instruction

```
t = new int [10];
```

associe la référence `r'` à la référence `r` dans la mémoire. L'environnement est alors [`t = r`] et la mémoire [`r = r'`, `r' = [0,0,0,0,0,0,0,0,0,0]`]. Il est, bien entendu, possible de déclarer la variable `t` en lui donnant une valeur initiale. Ainsi, au lieu d'écrire

```
int [] t;  
t = new int [10];
```

on peut écrire

```
int [] t = new int [10];
```