

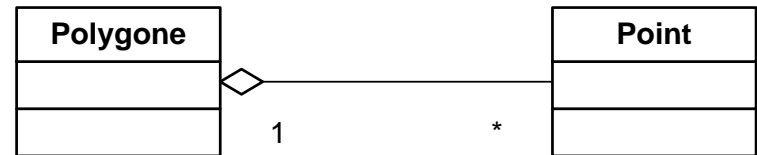
# Tableaux

Jean-François Berdjugin  
IUT1, Département SRC, L'Isle  
d'Abeau

# Pourquoi

Nous avons encapsuler dans nos objets les variables d'instance et les méthodes.

Comment faire pour regrouper plusieurs valeurs de même type : en utilisant des structures de données dont la plus connue les tableaux.



# Tableaux

En java un tableau est de taille fixe.

Il contient des type primitifs, ou des références d'objets ou des références de tableau.

Rem: en java il existe beaucoup d'autres structures de données (les collections).

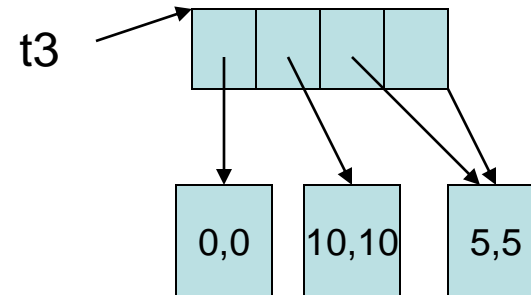
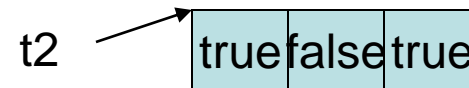
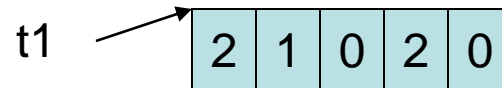


Tableau de points

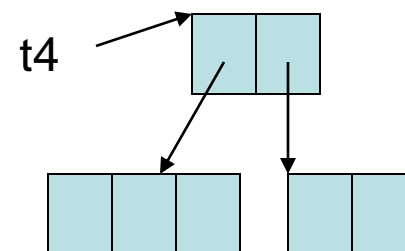
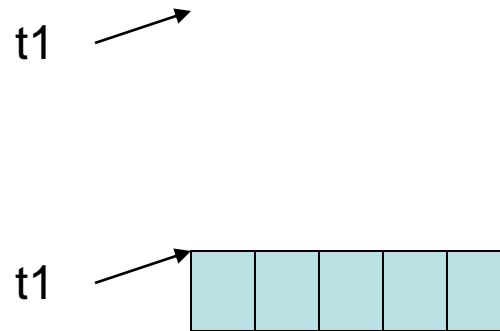


Tableau de tableau

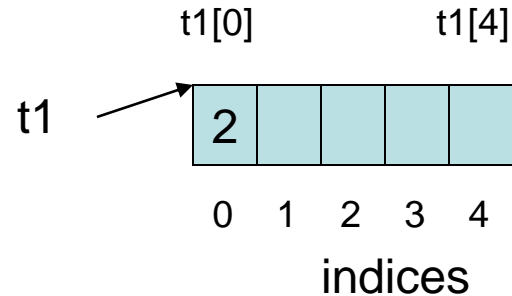
# Objets

- Les tableaux sont des objets particuliers dont l'instanciation et l'accès ont été réécrits.
- `int[] t1` //t1 est déclaré comme étant un tableau d'entiers.
- `t1 = new int[5]` //t1 est instancié et peut contenir 5 entiers.



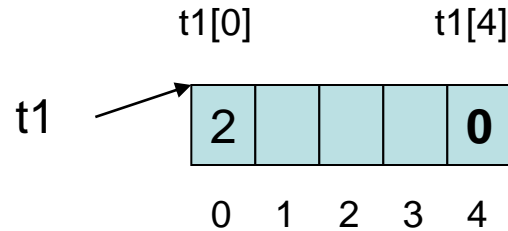
# « getters and setters »

- L'équivalent de la notion de getter et setter est l'utilisation de [] avec un indice.
- `t1[0]=2` //la case d'indice 0 reçoit 2 (set)
- `int x= t1[0]` //x reçoit la valeur de la case 0 (get)

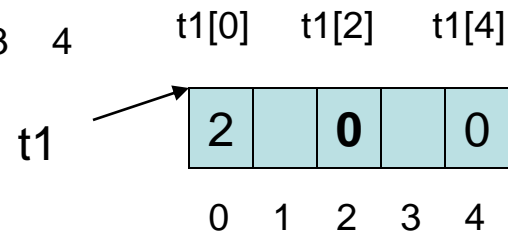


# « getters and setters »

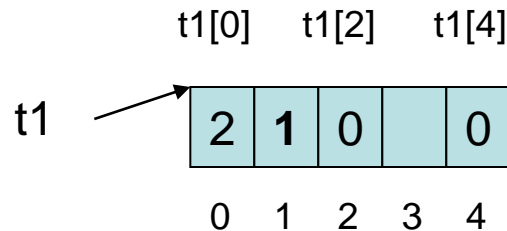
- $t1[4] = 0$



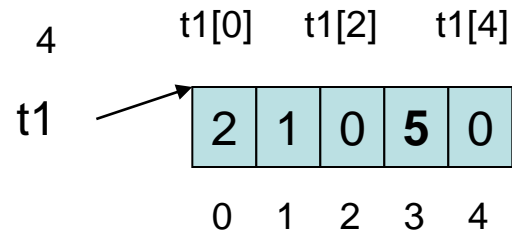
- $t1[2] = t1[4]$



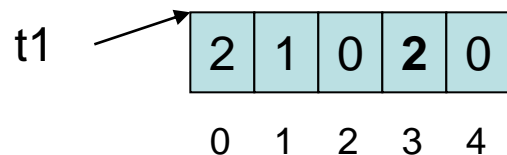
- $t1[1] = t[0]/2$



- $t1[3]=5$



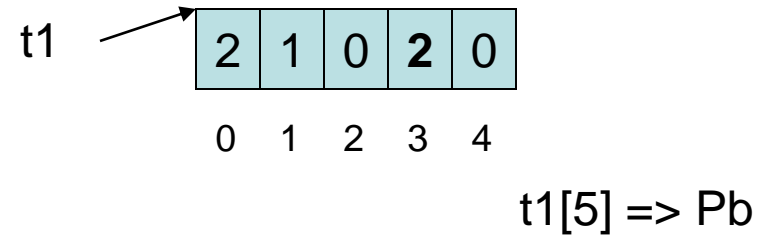
- $t1[3] = 2$



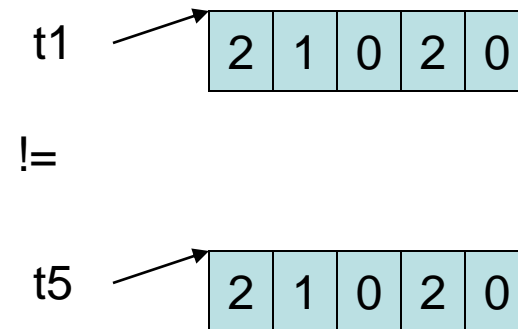
# « getters and setters »

- Petits problèmes

Les indices en dehors  
des bornes



- Liés aux objets  
égalité, perte de  
référence, ...



# Pièges

- Déplacement de capacité
  - `int[] t = new int[2]; t[4]=0 //pb`
- Alias involontaires
  - `int[] t1; int[] t2;`
  - `t1 = new int[2];`
  - `t2 =t1;`
  - `t1[0]=4;`
  - `t2[1]=6; //modifie t1[1]`



# Une variable d'instance intéressante

- La longueur : length 

```
int t[] = new int[5];  
t[0]=1;  
for (int i = 1; i <t. length;  
    i++) {  
    t[i] = t[i-1] + 2;}
```

# Un raccourci de notation

Initialisation avec un ensemble de valeurs.

```
int[] notes = {1, 19, 2, 18, 3};
```

# Exercice

- Les éléments supérieur à la moyenne.

```
//lecture de la longueur
```

```
System.out.println(« nb elt ?»);
```

```
Scanner c = new Scanner();
```

```
Int l =c.readInt();
```

```
int t[] = new int[l] //instanciation du tableau
```

```
double m = 0; //la moyenne
```

```
//lecture des éléments et calcul de la moyenne
```

```
for(int i=0; i<t.length;i++) {
```

```
    t[i] = c.nextInt();
```

```
    m = m + t[i];
```

```
}
```

```
m=m/t.length;
```

```
//Affiche du resultat
```

```
for(int i=0; i<t.length;i++) {
```

```
    if (t[i] > m)
```

```
        System.out.println(t[i]);
```

```
}
```

# Méthode et tableaux

- Les tableaux sont des objets, la valeur de leur référence est passée en paramètre => le tableau peut-être modifié par la méthode.  
`public static void tri(int t[])`
- Les tableaux sont des objets, une référence vers un tableau peut-être retournée  
`public static int[] tri(int t[])`