UNIVERSITÉ CÔTE D'AZUR

Programmation Orientée Objet

Tableaux

Frédéric Mallet

http://deptinfo.unice.fr/~fmallet/

UNIVERSITÉ CÔTE D'AZUR

Objectifs

- Syntaxe
  - Paramètres de la ligne de commande
  - Paramètres variables
- Structures de contrôle
  - for-each
- Structures de données
  - **Tableaux simples et multidimensionnels**
  - Classes utilitaires sur les tableaux

UNIVERSITÉ CÔTE D'AZUR

Tableaux

- Ensemble ordonnés d'éléments
  - Accès (lecture/écriture) en temps constant
  - Type unique choisi à la déclaration
    - Type primitif
    - Classe (polymorphisme)
  - **Longueur** statique choisie à la construction
  - Les tableaux sont des objets
    - Tableaux de tableaux (≠ matrice)

UNIVERSITÉ CÔTE D'AZUR

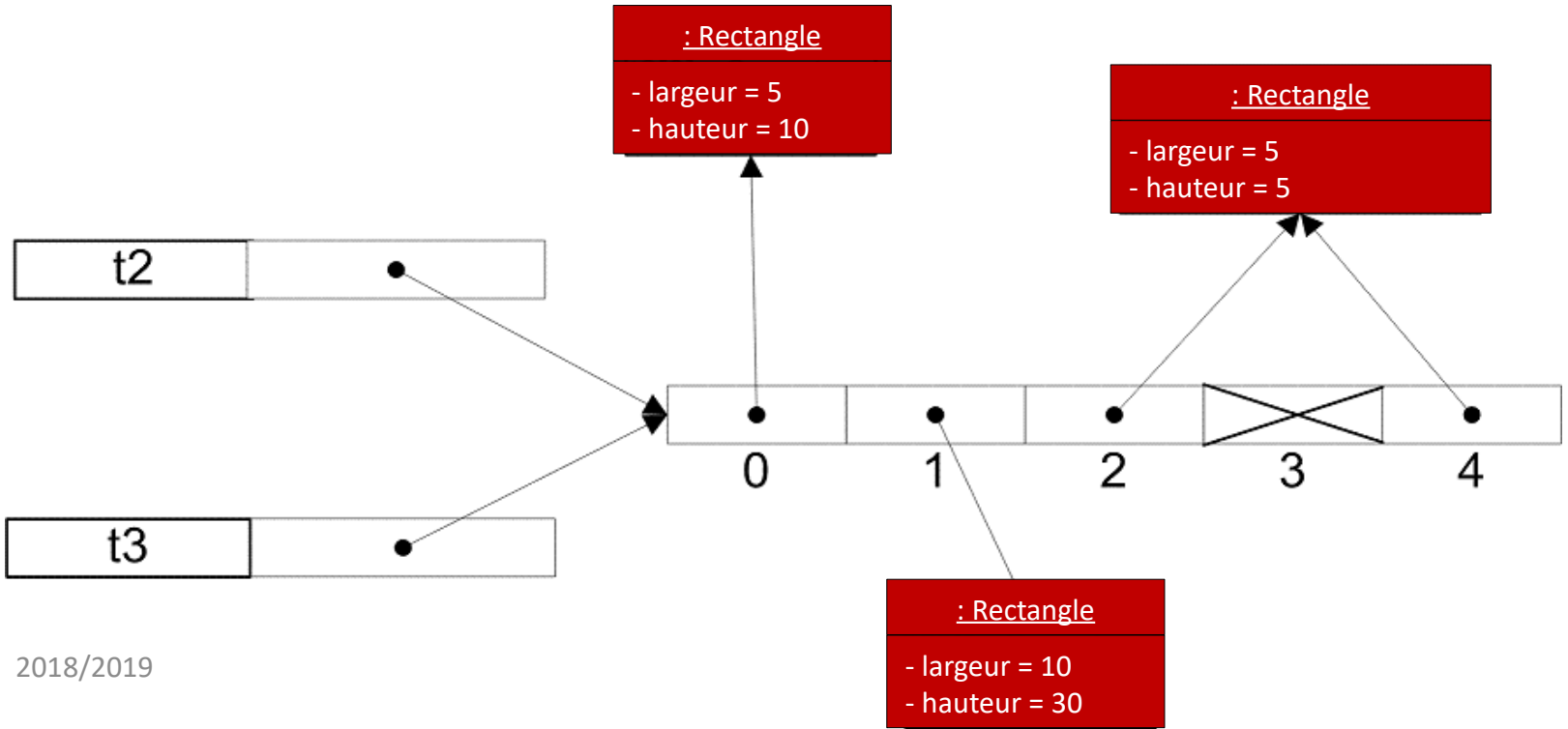
Tableaux : syntaxe

- Déclaration (référence)
  - `int[] t1; ou int t1[];`
  - `Rectangle[] t2;`
- Construction du tableau (pas de ses éléments)
  - `t1 = new int[10];`
  - `t2 = new Rectangle[5];`
- Affectation de tableaux (pas de copie globale)
  - `Rectangle[] t3 = t2;`
- Longueur (statique)
  - `t3.length` → 5

UNIVERSITÉ CÔTE D'AZUR

Indice: accès aux éléments

- Lecture ou écriture
  - `t1[0] = 24;` `t2[0] = new Rectangle(5,10);`
  - `t2[1] = new Rectangle(10,30);`
  - `t2[2] = new Rectangle(5,5);`
  - `t2[4] = t2[2];` // copie de référence



UNIVERSITÉ CÔTE D'AZUR

ArrayIndexOutOfBoundsException

- A chaque accès (lecture/écriture)
  - Les bornes sont vérifiées
  - => Pas d'accès non autorisé à la mémoire
- Example
  - `int[] t = new int[5];`
  - `t[-1] => ArrayIndexOutOfBoundsException -1<0`
  - `t[5] => ArrayIndexOutOfBoundsException 5>=5`

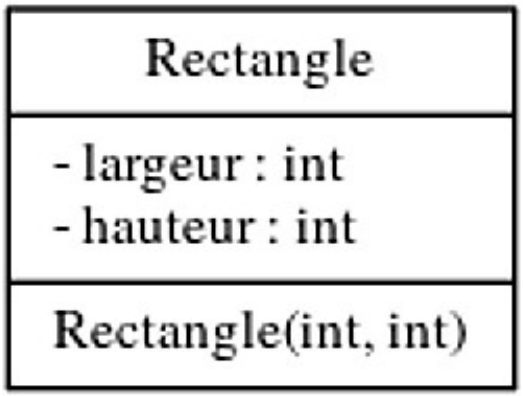
UNIVERSITÉ CÔTE D'AZUR

Tableaux: initialisation

- Lors de la déclaration

```
int[] t = {12, 25, -4, 3*5};
t.length → 4

Rectangle r = new Rectangle(5,10);
Rectangle[] tr = {new Rectangle(5,5), r};
tr.length → 2
```



- Tableau anonyme

```
t = new int[]{40, 52, 23, -4, 36, 4};
t.length → 6
```

La taille d'un tableau ne change JAMAIS, mais la même référence peut pointer successivement sur des tableaux de longueurs différentes

UNIVERSITÉ CÔTE D'AZUR

Parcourir les éléments

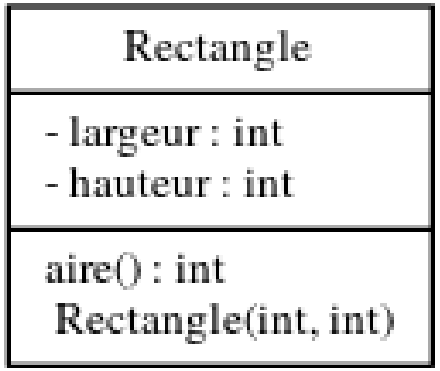
```
Rectangle[] tab;
```

- Itération: **for**

```
for (int i = 0; i<tab.length; i++)
    tab[i].aire();
```

- Itération: **for-each**

```
for (Rectangle r : tab)
    r.aire();
```



UNIVERSITÉ CÔTE D'AZUR

La ligne de commande

- Afficher les paramètres

```
class Arguments {
  static public void main(String[] args) {
    for(String arg : args) {
      System.out.println(arg);
    }
  }
}
```

- Usage :

```
>java Arguments programmation Java 2012
programmation
Java
2012
```

UNIVERSITÉ CÔTE D'AZUR

Paramètres variables

Operation binaire

```
class Addition {
  int calcule(int v1, int v2){
    return v1 + v2;
  }
}


new Addition().calcule(5, 12);
```

Opération N-aire

```
class Add {
  int calcule(int[] valeurs){
    int somme = 0;
    for(int v : valeurs)
      somme += v;
    return somme;
  }
}

new Add().calcule(new int[]{5,12});
```

UNIVERSITÉ CÔTE D'AZUR

Paramètres variables

Operation binaire

```
class Addition {
  int calcule(int v1, int v2){
    return v1 + v2;
  }
}
```

```
new Addition().calcule(5, 12);
```

Opération N-aire

```
class Add {
  int calcule(int … valeurs){
    int somme = 0;
    for(int v : valeurs)
      somme += v;
    return somme;
  }
}
```

```
new Add().calcule(5,12,40,-5);
```

UNIVERSITÉ CÔTE D'AZUR

Classe utilitaire : `java.util.Arrays`

- Manipulations courantes de tableaux (opérations statiques)
  - Recherche: `int binarySearch(T[] tab, T val);`
  - Copie: `T[] copyOf(T[] tab, int newLength);`
  - Comparaison: `boolean equals(T[] t1, T[] t2);`
  - Remplissage: `void fill(T[] tab, T val);`
  - Tri: `void sort(T[] tab)`
  - Affichage: `String toString(T[] tab);`
- Remplacer T selon le type
  - `boolean, char, byte, short, int, long, float, double, Object`

UNIVERSITÉ CÔTE D'AZUR

Affichage

- Affiche la référence

```
int[] t = {2012, 2013};
System.out.println(t);
```

[I@addbf1

- A la main

```
for(int i : t) System.out.print(i+" ");
System.out.println();
```

2012 2013

- Avec la classe Arrays

```
System.out.println(Arrays.toString(t));
```

[2012, 2013]

UNIVERSITÉ CÔTE D'AZUR

Tableaux et énumérations

- Les valeurs et les noms des énumérés

```
enum Couleur {
   TREFLE, CARREAU, CŒUR, PIQUE;
}
```

  - Accéder aux littéraux
    - Couleur[] Couleur.values()
    - String[] Couleur.valueNames()

UNIVERSITÉ CÔTE D'AZUR

Tableaux à plusieurs dimensions

- Déclaration
  - `int[][] t4;` // tableau de tableaux d'entiers
- Construction du tableau
  - `t4 = new int[5][];` // 5 tableaux d'entiers
- Utilisation
  - `t4[0] = t1;` // copie de références
  - `t4[1] = new int[3];`
- Remarque
  - `t4.length` → 5
  - `t4[0].length` → 10 `t4[1].length` → 3

UNIVERSITÉ CÔTE D'AZUR

Initialisation en bloc

Lors de la déclaration

```
int[][] t = {{2012, 2013, 2014, 2015},
             {2000, 2004, 2008}};
```

`t.length` → 2

`t[0].length` → 4

`t[1].length` → 3

Tableau anonyme

```
int[] t2 = new int[10];
t = new int[][]{new int[]{2012, 2013}, t2};
```

`t.length` → 2

UNIVERSITÉ CÔTE D'AZUR

Affichage

- Affiche la référence

```
int[][] t = {{2012, 2013, 2014, 2015},
             {2000, 2004, 2008}};
System.out.println(t);
```

[[I@42e816

- Avec la classe Arrays

```
System.out.println(Arrays.toString(t));
```

[[I@9304b1, [I@190d11]

```
System.out.println(Arrays.deepToString(t));
```

[[2012, 2013, 2014, 2015], [2000, 2004, 2008]]