

# Bases de la POO / Java

1

## Compléments packages et tableaux

# Package

2

## → Définition et structure

Ensemble fonctionnel cohérent de :

- Classes Java (fichiers .class)
- Interfaces Java (fichiers .class)
- Sous packages

Un package = un répertoire physique

Un sous package = un sous répertoire

Structure récursive

- Fichier **src.jar** installé par défaut, avec désignation dans la variable d'environnement CLASSPATH

# Exemples de packages standards

3

- **Packages prédéfinis**

- **java.lang** (Object, System, Class, String, ...)
- **java.util** (Calendar, ArrayList, HashMap, ...)
- **java.applet**
- **java.awt**
- **java.awt.event**
- ...

# Packages et accès

4

→ Pour utiliser une classe C d'un package P, il faut que :

- C soit définie **public** dans P
- le code exploitant C doit la désigner correctement
- Deux désignations possibles :
  - individuelle via chemin complet d'accès à la cible
    - ✦ `java.util.Date d;`
  - collective via une instruction **import**
    - ✦ `import java.util.*;`
    - ✦ `Date d;`

# Classe **java.util.Calendar**

5

## → Permet d'obtenir le point courant du temps

- Classe abstraite / Méthode *getInstance*
- Analogue à la méthode *currentTimeMillis* de la classe **System**
- Méthode *toString* peu exploitable
- Accès aux attributs publics YEAR, MONTH et DAY\_OF\_MONTH (Cf. exemple en annexe)
- Formatage manuel pour obtenir aaaa/mm/jj

[http://jmdoudoux.developpez.com/cours/developpons/java/chap-utilisation\\_dates.php](http://jmdoudoux.developpez.com/cours/developpons/java/chap-utilisation_dates.php)

# Tableaux

6

- `int [] t;`
- `t = new int[10];`
- `int taille = t.length;`

→ Arrays utilities

→ tri, remplissage, égalité...

→

<http://docs.oracle.com/javase/1.5.0/docs/api/java/util/Arrays.html>

# Exemple 1

7

```
import java.util.Arrays;
```

```
int[]    T1={0, 6, 2, -4, 3, 8, -11, 0, 1};
```

```
String[] T2={"bleu","rouge","blanc","vert","mauve","indigo"};
```

```
Arrays.sort(T1);
```

```
for (int i=0; i<T1.length; i++) System.out.print(T1[i] + " ");
```

```
Arrays.sort(T2);
```

```
for (int i=0; i<T2.length; i++) System.out.print(T2[i] + " ");
```

→ Exécution

-11 -4 0 0 1 2 3 6 8

blanc bleu indigo mauve rouge vert

# Exemple 2

8

```
int[] T1 = new int[10];  
String[] T2 = new String[7];  
  
Arrays.fill(T1, 5);  
  
Arrays.fill(T2, 1, 3, "bleu");
```

→ Exécution

5 5 5 5 5 5 5 5 5 5

null bleu bleu null null null null



# Exemple 3

9

```
int[] T1 = {0, 6, 2, 4, 3};
```

```
int[] T2 = {0, 6, 2, 4, 3};
```

```
Tests.Unit(false, T2==T1);
```

```
Tests.Unit(false, T2.equals(T1));
```

```
Tests.Unit(true, Arrays.equals(T1, T2));
```

## → Exécution

Valeur attendue : false

Valeur obtenue : false

Valeur attendue : false

Valeur obtenue : false

Valeur attendue : true

Valeur obtenue : true