

Cours de POO avec Java (Partie 3)

Module GL et POO

2LMD,

2014

Kahloul Laid

Avant de commencer

Questions?

1: c'est quoi 'string[] args'

2: dans la solution de exo 2 série 2 tu as fait `public static void main()` et après tu as créé une instance 'eq'? mais dans le cours on dit que si on fait 'public static' avant le 'main' on ne pas besoin de créer des instances.

Réponses

```
public static void main(String [] args) {
```

```
String one = args[0]; //=="one"
```

```
String two = args[1]; //=="two"
```

```
}
```

```
C:\ java MyProgram one two
```

Mes Questions:

- Que signifie le mot clé **super**?
- Que signifie le mot clé **static**?
- Est-il possible d'utiliser une classe **sans créer des objets** de cette classe?
- Que signifie le mot clé **protected**?

Objectif

Avancer dans la programmation OO avec Java :
organisation de programmes sous forme d'un
ensemble de **packages**.

Plan

- L'opérateur: **this**
- La **récurtivité** dans la définition des classes
- Les paquetages (**packages**): c'est quoi un package? Comment créer un package? Comment utiliser un package?

Le mot clé **this** (1)

- Le mot clé **this** est une référence à l'objet courant lors de l'instanciation de la classe;

- Exemple:

```
class C{
```

```
    int x, y;
```

```
    C(int x, int y){
```

```
        this.x=x;
```

```
        this.y=y;
```

```
    }
```

```
    ...
```

```
 }
```

this réfère à l'objet courant



La récursivité (1)

- Il est possible d'utiliser **le nom de la classe** **durant sa définition** (définition récursive);
- Le nom de la classe peut être le **type de certains arguments** ou de **certaines variables locales** des méthodes de cette classe;
- Le nom de la classe peut être le **type de certains attributs** de cette classe elle-même.

La récursivité (2)

exemple

- Exemple:

```
class Personne {  
    String nom;  
    String prénom;  
    Personne père;  
    Personne mère;
```

....

```
}
```

La récursivité (2)

exemple

```
class Complex{  
    float x, y;  
  
    void add(Complex z){  
    }  
  
    ....  
}
```

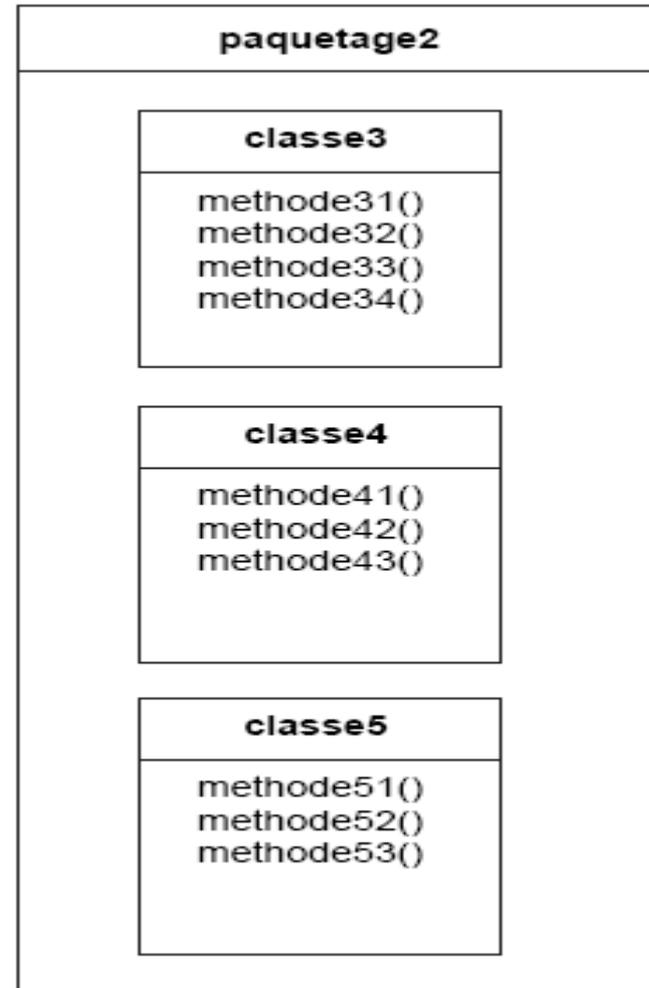
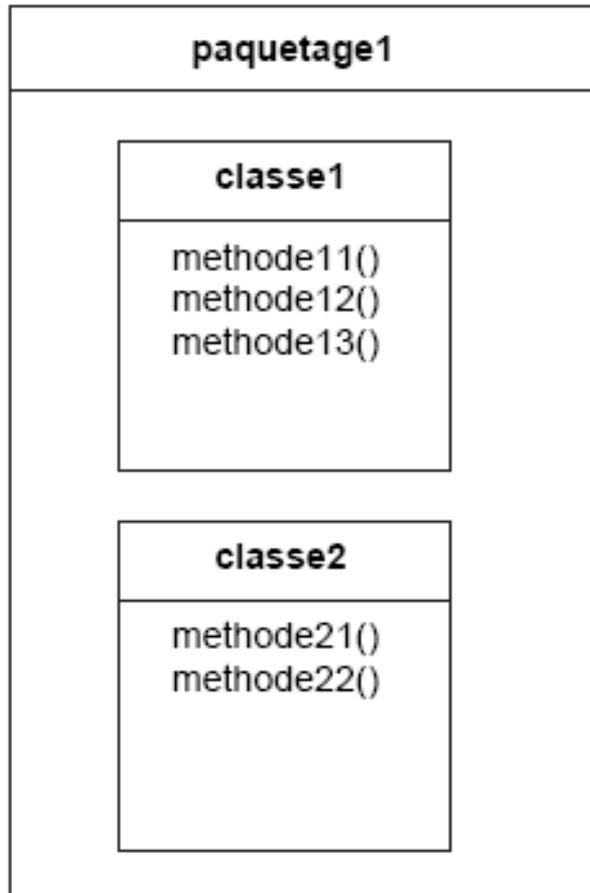
Packages (1):

Définitions

- Une **bibliothèque** de programme;
- Un **répertoire** de fichiers;
- Un **ensemble** de classes;

Et donc c'est une manière pour **organiser** le programme et d'avoir une certaine hiérarchie

Packages (2): exemple



Packages (3): création et usage

- Lors de la création: À l'entête du code:

package < nom_de_package >

- Lors de l'usage: À l'entête du code:

import paquetage1.classe1;

- Il est possible d'importer tous les classes dans un package **import** paquetage1.*;

Packages (5): sous packages

- Il est possible d'avoir une hiérarchie de packages: des packages englobés dans d'autres;

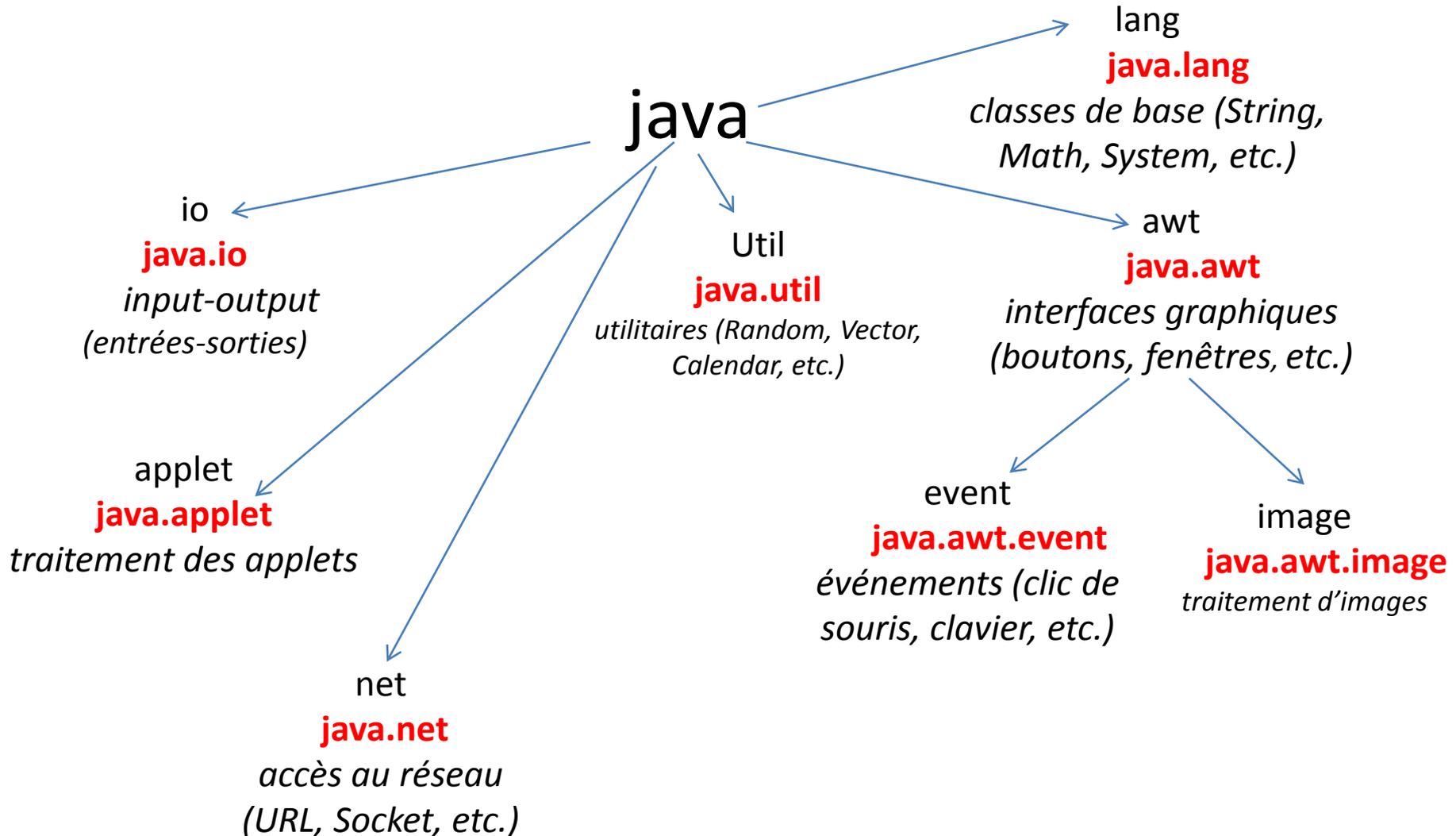
Dans l'usage:

```
import pack1.pack11.pack111...
```

- C'est le cas de tout le système **Java**: une hiérarchie de package.

Les packages (6) :

exemples de quelques packages java



Les packages (7) :

rappel : droit d'accès ou portée

- Les attributs et méthodes non précédés par aucun modificateurs (ni **private**, ni **public**, ...) sont accessibles dans le package.

Les packages (8) :

Exemple: Création d'un package

- Le package= un répertoire. Donc on doit suivre les étapes suivants:
 - 1) Créer un répertoire qui sera le package. Exemple **pack1**;
 - 2) Création du fichier java dans ce répertoire. Ce fichier java doit avoir dans son entête le nom du package (répertoire);
 - 3) Le nom du fichier java doit être le **nom de l'une des classes**; celle là doit être déclarée public

Les packages (9) :

Exemple: Création d'un package

Exemple de code pour créer un package pack1 contenant une seule classe Classe1:

```
package pack1 ;  
public class Classe1 {  
    private int d11 = 1;  
    public int d12 = 2;  
    int d13 = 3;  
  
    public methode() {  
        return d13;  
    }  
}
```

Nom du package

Cette classe public portera le même nom du fichier

Les packages (10) :

Exemple: usage du package

On peut utiliser la classe Classe1, dans une classe Classe2 créée dans un autre package pack2

```
package pack2;  
import pack1.*;  
public class classe2{  
  Classe1 C1;  
  ...  
}
```

Un autre package

Importation de toutes les classes du package pack1

la classe Classe1 est définie dans pack1

Les packages (10) :

Exemple: compilation et exécution

- Compilation:

```
javac -cp . nom_pack/classe.java
```

- Exécution:

```
java -cp . nom_pack/classe
```