

Paquetages

J. Sam, J.-C. Chappelier

version 1.0 de mars 2014

1 Notion de paquetage

En Java, il est possible de regrouper les classes en paquetages (bibliothèques). Les paquetages ont une incidence sur la gestion des droits d'accès. On met usuellement dans un même paquetage des classes relatives à un concept commun.

Pour les programmes de taille importante auxquels plusieurs programmeurs contribuent, la notion de paquetage permet notamment :

- une meilleure organisation ;
- d'éviter des conflits de noms.

Si l'on ne fait rien de particulier, les classes que l'on programme se trouvent *de facto* dans un paquetage : le paquetage par défaut.

Les trois étapes à suivre pour définir/utiliser un paquetage sont :

1. déclarer le paquetage courant en début de chaque fichier ;
2. indiquer les classes *publiques* du paquetage (une par fichier) ;
3. accéder aux classes publiques au moyen du nom du paquetage.

Ces trois éléments sont brièvement présentés dans ce qui suit et illustrés au moyen de l'exemple de la figure 1.

2 Spécification d'un paquetage

Pour spécifier un paquetage, il faut ajouter son nom à chaque fichier concerné, comme première instruction du fichier :

```
package nomDuPaquetage;
```

Toutes les classes déclarées dans le fichier appartiendront au paquetage spécifié. Si aucun paquetage n'est spécifié, les classes appartiendront au paquetage par défaut.

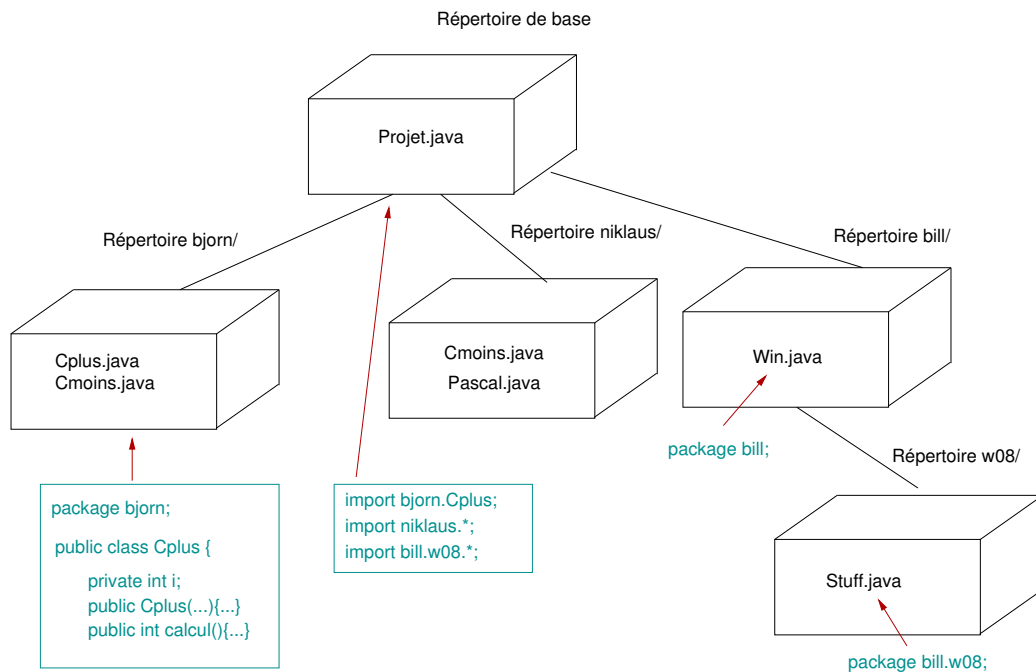


FIGURE 1 – Exemple illustratif pour les paquets.

Si les classes sont stockées dans un sous-répertoire, le nom doit correspondre au chemin pour retrouver le paquetage depuis le répertoire de base.

Chaque sous-répertoire correspond à un sous-paquetage.

Exemples (cf Fig. 1) :

Pour le fichier `bjorn/Cplus.java` (à gauche de la figure), on aurait :

```
package bjorn;
class Cplus { ... }
```

et pour le fichier `bill/wz08/Stuff.java` (à droite) :

```
package bill.wz08;
class Stuff { ... }
```

3 Composants publiques

Quand une classe appartient à un paquetage, elle est visible *seulement* par les autres classes du même paquetage. Pour la rendre visible ailleurs, il faut ajouter le modificateur `public` à la classe. Notez qu'il ne peut y avoir qu'une classe publique par fichier. La seule classe pouvant être publique doit porter le même nom que le fichier dans laquelle elle est définie.

Exemple :

```
package bjorn;
public class Cplus{
    private int i;
    public Cplus () {
        ...
    }
    public int calcul () {
        ...
    }
}
```

4 Accès aux composants d'un paquetage

Les classes publiques d'un paquetage sont accessibles *seulement* à travers le nom du paquetage. On peut soit :

1. Indiquer le nom complet de la classe :

```
class MonProjet {
    public static void main (String[] args) {
        bjorn.Cplus cp  = new bjorn.Cplus();
        bjorn.Cmoins cm1  = new bjorn.Cmoins();
        niklaus.Cmoins cm2 = new niklaus.Cmoins();
        bill.w08.Stuff b = new bill.w08.Stuff();
        ...
    }
}
```

2. Importer (au début du fichier) les classes désirées :

```
import bjorn.Cplus;
import bjorn.Cmoins;
// import niklaus.Cmoins; //Ambiguïté : il y a déjà un
//autre Cmoins !!!

import bill.w08.Stuff;

class AutreProjet {
    public static void main (String[] args) {
        Cplus cp = new Cplus();
        Cmoins cm1 = new Cmoins();          // Classe Cmoins de Bjorn

        // pour éviter l'ambiguïté :
        niklaus.Cmoins cm2 = new niklaus.Cmoins();
        Stuff b = new Stuff();
        ...
    }
}
```

Mais notez bien qu'on ne peut pas importer de la sorte plusieurs classes ayant le même nom.

3. Importer toutes les classes d'un paquetage :

```
import bjorn.*;
// import niklaus.*;          Ambiguïté !!!
import bill.*;
import bill.w08.*;
...
```

Attention cependant, il faut importer chaque sous-paquetage séparément ! L'étoile ne représente *que* les classes du paquetage, mais *pas* celles des sous-paquetages.