Table des matières

Les packages

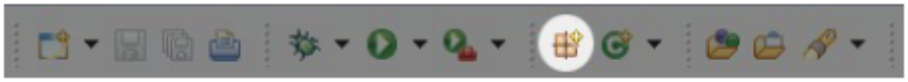
Lorsque nous avons été confrontés pour la première fois aux packages, c'était pour importer la classe Scanner via l'instruction ***import java.util.Scanner*** ;. Le fonctionnement des packages est simple à comprendre : ce sont comme des dossiers permettant de ranger nos classes. Charger un package nous permet d'utiliser les classes qu'il contient.

Il n'y aura rien de franchement compliqué dans ce chapitre si ce n'est que nous reparlerons un peu de la portée des classes Java.

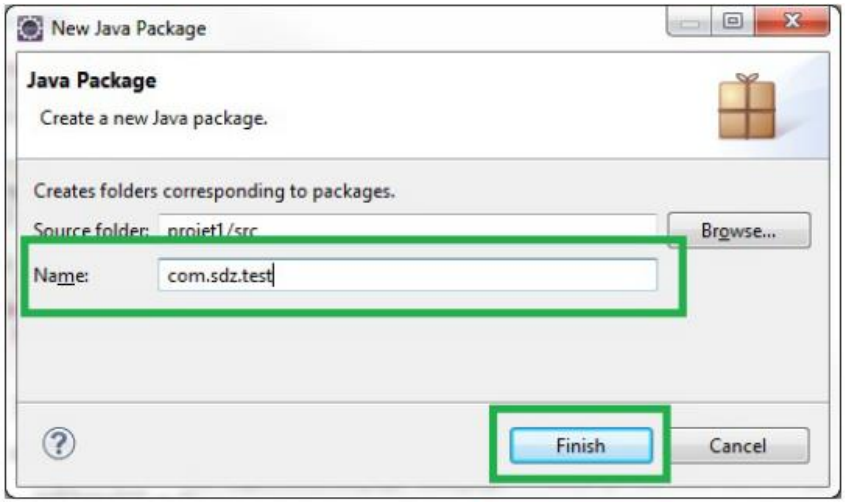
Création d'un package

L'un des avantages des packages est que nous allons y gagner en lisibilité dans notre package par défaut, mais aussi que les classes mises dans un package sont plus facilement transportables d'une application à l'autre. Pour cela, il vous suffit d'inclure le dossier de votre package dans un projet et d'y importer les classes qui vous intéressent !

Pour créer un nouveau package, cliquez simplement sur cette icône comme à la figure suivante (vous pouvez aussi effectuer un clic droit puis New > Package).



Une boîte de dialogue va s'ouvrir et vous demander le nom de votre package, comme à la figure suivante.



Il existe aussi une convention de nommage pour les packages :

- ceux-ci doivent être écrits entièrement en minuscules ;

- les caractères autorisés sont alphanumériques (de a à z, de 0 à 9) et peuvent contenir des points (.) ;
- tout package doit commencer par com, edu, gov, mil, net, org ou les deux lettres identifiant un pays (ISO Standard 3166, 1981) ; « fr » correspond à la France, « en » correspond à l'Angleterre (pour England)etc.
- aucun mot clé Java ne doit être présent dans le nom, sauf si vous le faites suivre d'un underscore (« _ »), comme ceci : com.sdz.package_.

Pour le cas qui nous occupe, appelons-le com.tri.test. Cliquez sur Finish pour créer le package. Et voilà : celui-ci est prêt à l'emploi.

Je vous invite à aller voir dans le dossier où se trouvent vos codes sources : vous constaterez qu'il y a l'arborescence du dossier com/tri/test dans votre dossier src.

Vous conviendrez que la création d'un package est très simple. Cependant, je ne peux pas vous laisser sans savoir que la portée de vos classes est affectée par les packages…

Droits d'accès entre les packages

Lorsque vous avez créé votre première classe, vous avez vu qu'Eclipse met systématiquement le mot clé public devant la déclaration de la classe. Je vous avais alors dit que ***public class Ville*** et ***class Ville*** étaient sensiblement différents et que le mot clé ***public*** influait sur la portée de notre classe. En fait, une classe déclarée avec le mot clé public sera visible même à l'extérieur de son package, les autres ne seront accessibles que depuis l'intérieur du package : on dit que leur portée est ***default***.

Afin de vous prouver mes dires, je vous invite à créer un second package : je l'ai appelé ***« com.tri.test2 »***. Dans le premier package, ***com.tri.test***, créez une classe A de portée public et une classe B de portée default, comme ceci (j'ai, volontairement déclaré les variables d'instance public afin d'alléger l'exemple) :

```
package com.tri.test;
class B {
  public String str ="";
}
```

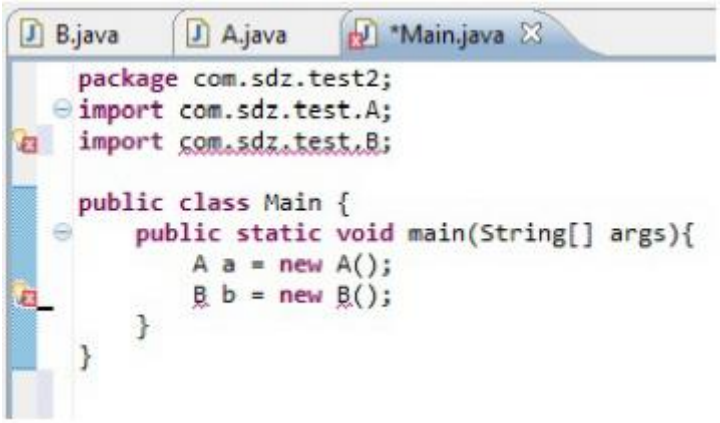
```
package com.tri.test;
public class A {
  public B b = new B();
}
```

Vous aurez remarqué que les classes contenues dans un package ont en toute première instruction la déclaration de ce package.

Maintenant que cela est fait, afin de faire le test, créez une classe contenant la méthode main, toujours dans le même package, comme ceci :

```
package com.tri.test;
public class Main {
  public static void main(String[] args){
    A a = new A();
    B b = new B();
    //Aucun problème ici
  }
}
```

Ce code, bien qu'il ne fasse rien, fonctionne très bien : aucun problème de compilation, entre autres. Maintenant, faites un copier-coller de la classe ci-dessus dans le package ***com.tri.test2***. Vous devriez avoir le résultat représenté à la figure suivante.



Vous pouvez constater qu'Eclipse n'aime ni l'instruction import com.sdz.test.B, ni l'instruction *B b = new B();*. Cela est dû à la déclaration de notre classe. J'irai même plus loin : si vous essayez de modifier la variable d'instance de l'objet A, vous aurez le même problème. Donc, ceci : *a.b.str = "toto";* n'est pas non plus autorisé dans ce package ! La seule façon de corriger le problème est de déclarer la classe B public. Rappelez-vous que seule la classe A avait été déclarée ainsi.

- Un package est un ensemble de dossiers et de sous-dossiers.
- Le nom du package est soumis à une convention de nommage.
- Si vous voulez utiliser un mot clé Java dans le nom de votre package, vous devez le faire suivre d'un underscore (« _ »).
- Les classes déclarées public sont visibles depuis l'extérieur du package qui les contient.
- Les classes n'ayant pas été déclarées public ne sont pas visibles depuis l'extérieur du package qui les contient.
- Si une classe déclarée public dans son package a une variable d'un type ayant une portée default, cette dernière ne pourra pas être modifiée depuis l'extérieur de son package.