UML Diagramme de Package (DPKG)

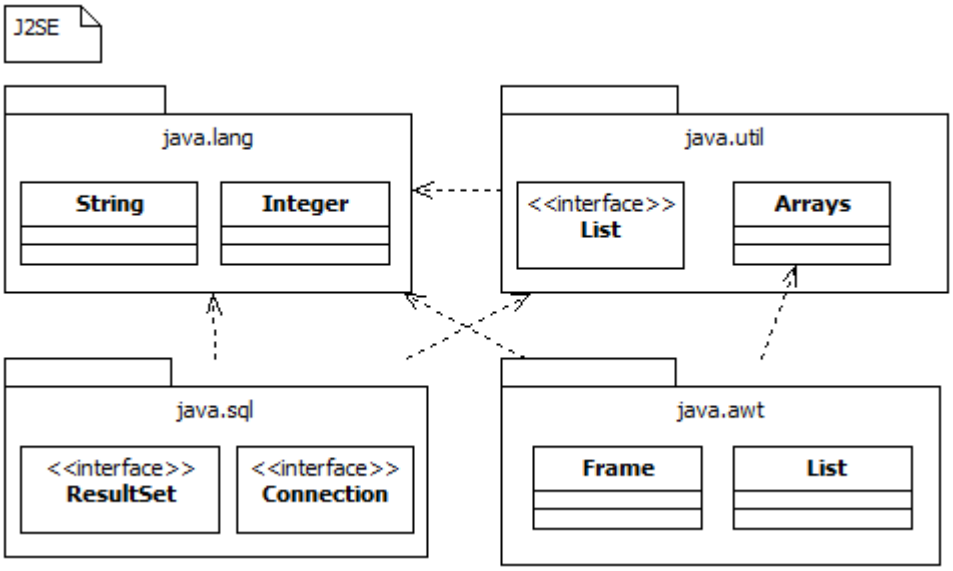
Table des matières

1.1 - Première définition et représentation graphique

Un paquetage (package) est un élément de modélisation qui contient d'autres éléments de modélisation (classes, cas d'utilisation, autres paquetages, …).

Un package est représenté graphiquement par un rectangle conteneur d'autres composants surmonté en haut à gauche d'un rectangle vide. Le nom du package est dans la partie supérieure du rectangle inférieur.

Packages de Java SE.



Note : la notion de package permet d'avoir plusieurs éléments de même nom dans des packages différents (cf List dans le package java.util et dans le package java.awt).

1.2 - DÉFINITION DÉTAILLÉE

Un paquetage (package) est un élément de modélisation qui contient d'autres éléments de modélisation (classes, autres paquetages, …).

Un paquetage définit un espace de nom (namespace).

Deux composants de deux paquetages différents sont différents, quel que soit leur nom.
Le nom complet d'un composant est composé du nom du package (préfixe) et du nom du composant.
java.util.List et java.awt.List sont deux composants différents. Le premier est un composant non graphique qui peut contenir un groupe de valeurs ou d'objets avec éventuellement des doublons alors que le deuxième est un composant graphique qui permet d'afficher dans une IHM des objets ou des chaînes de caractères.

Un élément d'un package peut être visible seulement dans le package ou par d'autres packages.

Si un composant d'un package dépend d'un composant d'un autre package il existe une dépendance entre les packages.
Il y a une dépendance lorsque un composant A possède un attribut ou une méthode de type B ou un paramètre de type B ou encore lorsqu'il existe une relation navigable de A vers B.
Une dépendance est formalisée par un trait discontinu et une flèche ouverte dirigée du composant dépendant vers le composant « maître ».

Les sous-types de relations inter-packages sont <<access>>, <<import>>.
A accède à B lorsqu'un composant de A accède par son nom complet à un élément de B. cf en java à :
public class JListModifiableStrings extends javax.swing.JFrame {

A importe B et ainsi A accède à un composant de B par son nom. cf en java :
import java.sql.*;
et plus loin :
Connection lcn;
Statement lstSQL;
ResultSet lrs;

La relation d'importation est représentée en UML par une relation de dépendance.

Un diagramme de packages est une représentation graphique d'un package ou plusieurs.

1.3 - ARCHITECTURE LOGIQUE

Une architecture logique est un regroupement de classes logicielles en paquetages.

L'objectif d'une architecture logique est d'encapsuler et de décomposer la complexité, de faciliter le travail en équipes, de faciliter la réutilisation et l'évolutivité.

Certains critères permettent de déterminer les composants et frontières du package :

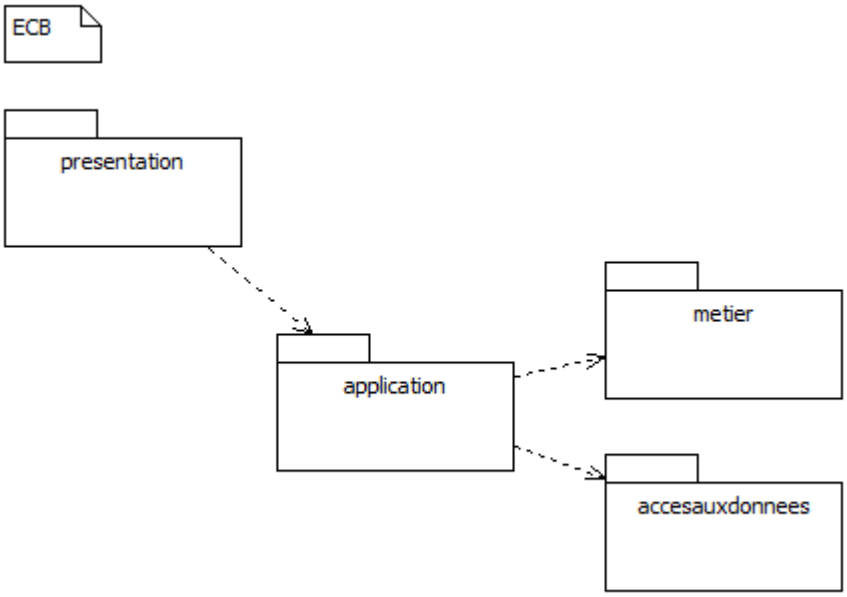
appartenance au même domaine,
hiérarchie,
composition,
fort couplage entre classes,
sémantique.

Exemples d'architectures logiques :

Le package java.util, le package javax.swing, ...
Architecture en couches (ECB),
Architecture Modèle - Vue - Contrôleur (MVC),
Architecture multi-tiers.

1.4 - Packages, couches et dépendances

Pattern ECB (Entity, Control, Boundary – Entité, Contrôleur, « Frontière » ou IHM).



1.5 - EXERCICES

1) Modifiez le diagramme de packages de J2SE en ajoutant les associations de dépendance.

2) Organisez ces classes en paquetages.