

Class path, packages, archives

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 3 février 2017

Packages : principe

- Chaque classe principale Java : dans son fichier
- Classe MyClass dans fichier ?

Packages : principe

- Chaque classe principale Java : dans son fichier
- Classe MyClass dans fichier ? `MyClass.java`
- Compilée dans fichier ?

Packages : principe

- Chaque classe principale Java : dans son fichier
- Classe MyClass dans fichier ? `MyClass.java`
- Compilée dans fichier ? `MyClass.class`
- Chaque fichier déclaré dans un *package* Sauf si package par défaut, non recommandé.
- Packages structurés hiérarchiquement, comme un arbre
- Structure indiquée par des points
- Exemples dans la JRE : `java.nio.file`,
`java.util.logging`
- Quand utilisation d'une classe, il faut l'importer en indiquant son package

Utilité des packages

- Recommandé : créez vos packages !
- Organiser par thème
- Unicité de nom
- Vous utilisez des classes créées par d'autres
- L'une d'elle s'appelle `MathUtils`
- Vous avez aussi une classe `MathUtils`
- Pas de problème si dans packages différents

Packages et répertoires

- Fichier dans un package `nom1.nom2.nom3`
- Se trouve dans sous-répertoire `nom1/nom2/nom3`
- Utilité de cette séparation en répertoires ?

Packages et répertoires

- Fichier dans un package `nom1.nom2.nom3`
- Se trouve dans sous-répertoire `nom1/nom2/nom3`
- Utilité de cette séparation en répertoires ? Organisation ;
Unicité de nom
- Chemin considéré à partir du répertoire source
- Dans Eclipse par défaut : `myproject/src/nom1/nom2/nom3`

Compilation

- Classes compilées : où ?

Compilation

- Classes compilées : où ?
- Souvent : dans répertoire bin ou target
- Structure des sources conservée !
- Exemple : classe dans `src/myutils/draft/MathUtils.java`
compilée dans `bin/myutils/draft/MathUtils.class`
- Package ?

Compilation

- Classes compilées : où ?
- Souvent : dans répertoire bin ou target
- Structure des sources conservée !
- Exemple : classe dans `src/myutils/draft/MathUtils.java`
compilée dans `bin/myutils/draft/MathUtils.class`
- Package ? `myutils.draft`

Archives

- Pour distribuer vos classes
- Ou partager entre vos propres projets
- JAR ?

Archives

- Pour distribuer vos classes
- Ou partager entre vos propres projets
- JAR ? Java ARchive
- Fichier compressé
- Contenu ?

Archives

- Pour distribuer vos classes
- Ou partager entre vos propres projets
- JAR ? Java ARchive
- Fichier compressé
- Contenu ? Collection de classes compilées (.class) et leur package
- Facultatif : code source
- Facultatif : `Main-Class` dans META-INF/MANIFEST.MF
- Et autres méta-données facultatives

Unicité de nommage

- Objectif du système de packages et de classes : assurer l'unicité de nom
- Pour éviter conflits lors utilisation d'un jar tiers (par exemple)
- Problème ?

Unicité de nommage

- Objectif du système de packages et de classes : assurer l'unicité de nom
- Pour éviter conflits lors utilisation d'un jar tiers (par exemple)
- Problème ? Il faut packages uniques !
- Convention : base = votre domaine internet inversé
- Exemple : `com.google.common.io.ByteSource`

Class path

- Comment lancer une application ?
- Il faut indiquer où aller chercher toutes les classes requises
- Dans de multiples répertoires ou fichiers JAR
- Class path : les chemins que la JVM utilise pendant son exécution pour charger les classes
- Class path fourni à la JVM au démarrage, exemple `java -cp path1:path2:path3 name1.name2.myclass`
- NB il doit contenir des classes compilées
- Classes dans la JRE : rien à préciser
- Class path par défaut ?

Class path

- Comment lancer une application ?
- Il faut indiquer où aller chercher toutes les classes requises
- Dans de multiples répertoires ou fichiers JAR
- Class path : les chemins que la JVM utilise pendant son exécution pour charger les classes
- Class path fourni à la JVM au démarrage, exemple `java -cp path1:path2:path3 name1.name2.myclass`
- NB il doit contenir des classes compilées
- Classes dans la JRE : rien à préciser
- Class path par défaut ? “.”
- Dans Eclipse : voir Java Build Path

Licence

Cette présentation, et le code LaTeX associé, sont sous [licence MIT](#). Vous êtes libres de réutiliser des éléments de cette présentation, sous réserve de citer l'auteur.
Le travail réutilisé est à attribuer à [Olivier Cailloux](#), Université Paris-Dauphine.