

Class path, packages, archives

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 17 mars 2019

Packages : principe

- Une classe Java a un nom simple
- Exemple : `Math.java`
- Une classe a aussi un nom complet, « fully qualified name »
- Exemple : `java.lang.Math`, `java.util.Scanner`
- Utilité du nom complet ?
- Chaque classe déclarée dans un *package* Sauf si package par défaut, non recommandé.
- Nom complet = nom du package point nom de la classe
- Référence à la classe par son nom complet ou nom simple avec `import`
- Packages structurés hiérarchiquement, comme un arbre
- Structure indiquée par des points
- Organiser par thème

Packages : principe

- Une classe Java a un nom simple
- Exemple : `Math.java`
- Une classe a aussi un nom complet, « fully qualified name »
- Exemple : `java.lang.Math`, `java.util.Scanner`
- Utilité du nom complet ? Assurer unicité ! (`MathUtils`)
- Chaque classe déclarée dans un *package* Sauf si package par défaut, non recommandé.
- Nom complet = nom du package point nom de la classe
- Référence à la classe par son nom complet ou nom simple avec `import`
- Packages structurés hiérarchiquement, comme un arbre
- Structure indiquée par des points
- Organiser par thème

Packages et fichiers

- Classe de nom complet `MyClass`
 - Fichier source ?
 - Compilée dans fichier ?
- Classe de nom complet `nom1.nom2.nom3.MyClass`
 - Fichier source ?
 - Compilée dans ?
- Dans Eclipse par défaut : `MyProject/src`, `MyProject/bin`

Packages et fichiers

- Classe de nom complet `MyClass`
 - Fichier source ? `<src>/MyClass.java` (`<src>` = rép. source)
 - Compilée dans fichier ?
- Classe de nom complet `nom1.nom2.nom3.MyClass`
 - Fichier source ?
 - Compilée dans ?
- Dans Eclipse par défaut : `MyProject/src`, `MyProject/bin`

Packages et fichiers

- Classe de nom complet MyClass
 - Fichier source ? `<src>/MyClass.java` (`<src>` = rép. source)
 - Compilée dans fichier ? `<dst>/MyClass.class`
- Classe de nom complet `nom1.nom2.nom3.MyClass`
 - Fichier source ?
 - Compilée dans ?
- Dans Eclipse par défaut : `MyProject/src`, `MyProject/bin`

Packages et fichiers

- Classe de nom complet MyClass
 - Fichier source ? `<src>/MyClass.java` (`<src>` = rép. source)
 - Compilée dans fichier ? `<dst>/MyClass.class`
- Classe de nom complet `nom1.nom2.nom3.MyClass`
 - Fichier source ? `<src>/nom1/nom2/nom3/MyClass.java`
 - Séparation en répertoires pour organisation et unicité de nom
 - Compilée dans ?
- Dans Eclipse par défaut : `MyProject/src`, `MyProject/bin`

Packages et fichiers

- Classe de nom complet `MyClass`
 - Fichier source ? `<src>/MyClass.java` (`<src>` = rép. source)
 - Compilée dans fichier ? `<dst>/MyClass.class`
- Classe de nom complet `nom1.nom2.nom3.MyClass`
 - Fichier source ? `<src>/nom1/nom2/nom3/MyClass.java`
 - Séparation en répertoires pour organisation et unicité de nom
 - Compilée dans ? `<dst>/nom1/nom2/nom3/MyClass.class`
- Dans Eclipse par défaut : `MyProject/src`, `MyProject/bin`

Compilation

- Structure des sources conservée !
- Exemple : rép. source = `src/main/java`, source dans `src/main/java/myutils/draft/MathUtils.java`
- Nom complet ?
- Compilée dans ?

Compilation

- Structure des sources conservée !
- Exemple : rép. source = `src/main/java`, source dans `src/main/java/myutils/draft/MathUtils.java`
- Nom complet ? `myutils.draft.MathUtils`
- Compilée dans ?

Compilation

- Structure des sources conservée !
- Exemple : rép. source = `src/main/java`, source dans `src/main/java/myutils/draft/MathUtils.java`
- Nom complet ? `myutils.draft.MathUtils`
- Compilée dans ? `bin/myutils/draft/MathUtils.class`

Exécution

Pour exécuter une classe, de quelles informations la JVM a-t-elle besoin ?

Exécution

Pour exécuter une classe, de quelles informations la JVM a-t-elle besoin ?

- Nom complet de la classe à exécuter (expl : `myutils.draft.MathUtils`)
- Répertoire de la classe compilée (expl : `bin`)
- Si dépendance ?

Exécution

Pour exécuter une classe, de quelles informations la JVM a-t-elle besoin ?

- Nom complet de la classe à exécuter (expl : `myutils.draft.MathUtils`)
- Répertoire de la classe compilée (expl : `bin`)
- Si dépendance ?
- Répertoires où trouver les classes dont la classe dépend (expl : `bin` et `myotherproject/bin`)

Archives

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

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 ?
- Convention : base = votre domaine internet inversé
- Exemple : `com.google.common.math.Quantiles`,
`org.apache.crunch.lib.Quantiles`

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.math.Quantiles`,
`org.apache.crunch.lib.Quantiles`

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 (sauf celles de l'API Java)
- 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 l'API Java : rien à préciser
- Class path par défaut ?
- Dans Eclipse : voir Java Build Path

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 (sauf celles de l'API Java)
- 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 l'API Java : 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.