

# Déploiement

Comment déployer des applications Java





Déploiement

# Objectifs du cours

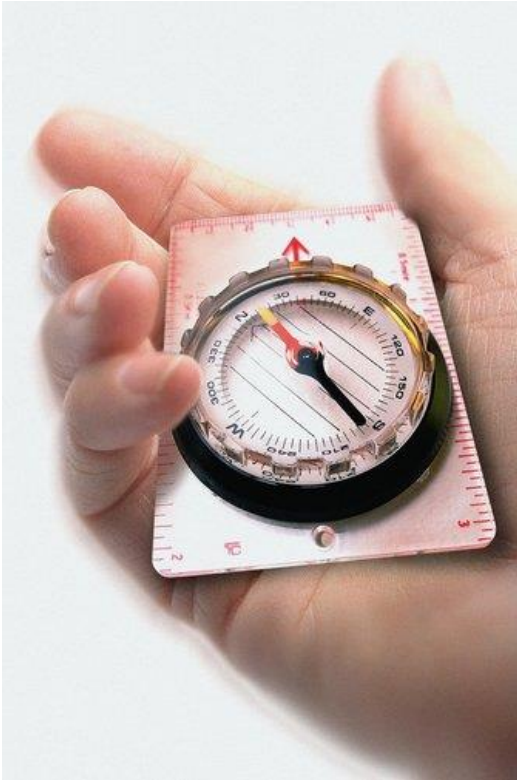
En complétant ce cours, vous serez en mesure de:

- Empaqueter des programmes dans des fichiers JAR



Préparation

# Course plan



- Introduction
- JAR Files

Déploiement

# INTRODUCTION



# Pourquoi déployer ?

- De nos jours, les logiciels ont besoin de beaucoup de fichiers :
  - Images
  - Icônes
  - Des sons
  - Fichiers de configuration
- Le déploiement évite aux utilisateurs de gérer ces fichiers
- L'utilisateur a juste à :
  - Mettre en place le programme
  - L'utiliser



# Méthodes de déploiement déjà existantes

- Sur Windows:
  - EXE files
  - MSI files
  - ...
- Sur Linux:
  - Debian packages (.deb)
  - Script files
- Sur Mac:
  - APP files

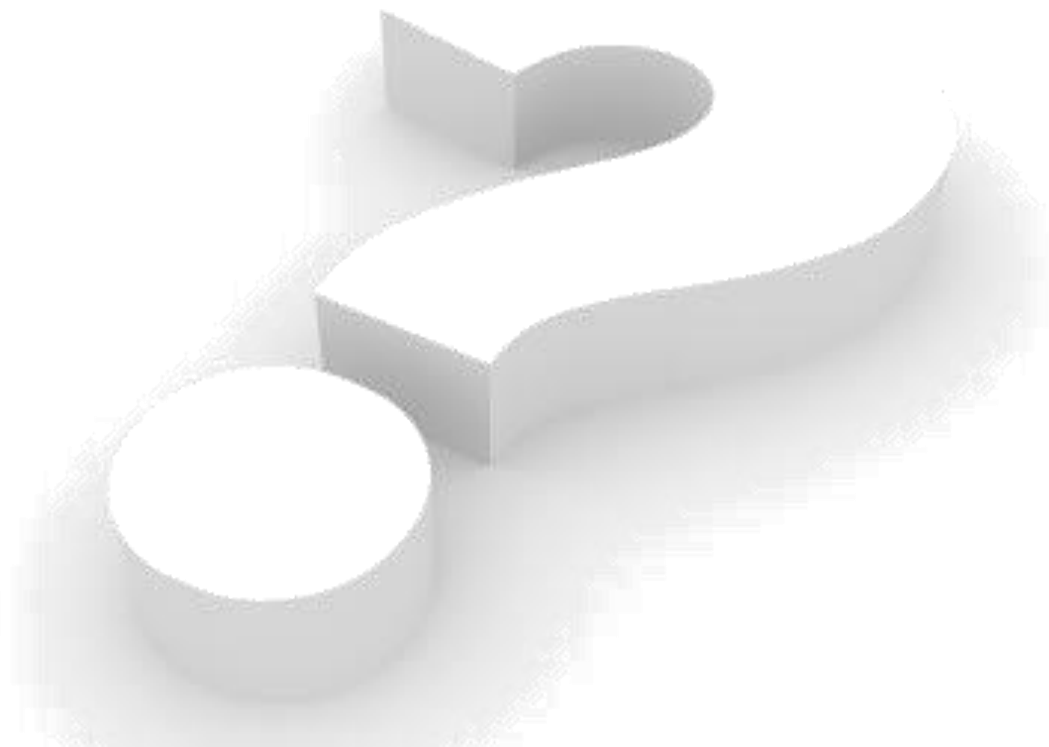


# Solutions Java

- Java est multiplateforme :
  - Généralement, vous n'avez pas à vous soucier de :
    - Exécutable Windows
    - Exécutable Linux
    - Exécutable Mac
- Java fournit :
  - Fichiers JAR exécutables
  - Java Web Start (et JNLP)
  - Fichiers WAR (vous les verrez si on a le temps)



# Questions ?





Déploiement

**JAR**



Jar

# Introduction

- Java Archive
- Permettre de rassembler :
  - fichiers .class
  - Images
  - Des sons
  - ...
- Peut être exécutable !





Jar

# La commande jar

- Disponible avec le JDK
- Autoriser la création de fichiers JAR à l'aide de la ligne de commande
- Syntaxe :

```
jar [options] [manifest] destination.jar input-file [input-file]
```

- Les fichiers JAR exécutables nécessitent un fichier manifeste.



# Les options de la commande jar

Options	Description
c	Créer une nouvelle archive
f	Permet de spécifier le nom de l'archive
v	Mode détaillé
t	Lister le contenu du jar
x file	Extraire le contenu de l'archive, vous pouvez extraire un fichier si vous spécifiez son nom
u	Permet la mise à jour de l'archive
m	Permet de spécifier un fichier manifeste afin de faire un jar exécutable



Jar

# Exemple

- Créer un rassemblement JAR :
  - MyClass.class
  - java.jpg
- Et nommé MyCoolJar.jar :

```
jar cf MyCoolJar.jar MyClass.class java.jpg
```



Jar

# Le fichier manifeste

- Nécessaire pour créer des fichiers JAR exécutables
- Généralement nommé manifest.mf
- Contenu :

```
Manifest-version: 1.0
```

```
Main-Class: class.containing.the.main.method.to.Launch
```

- Créez un JAR exécutable :

```
jar cmf manifest.mf MyCoolJar.jar MyClass.class java.jpg
```



Jar

# Signer un JAR

- Vous pouvez apposer une signature numérique à un JAR
- Vous avez besoin de deux outils :
- keytool :
  - Créer une clé dans un keystore
- jarsigner :
  - Signez votre JAR à l'aide d'une clé dans un keystore



Jar

# Signer un JAR

- Exemple dans la ligne de commande:

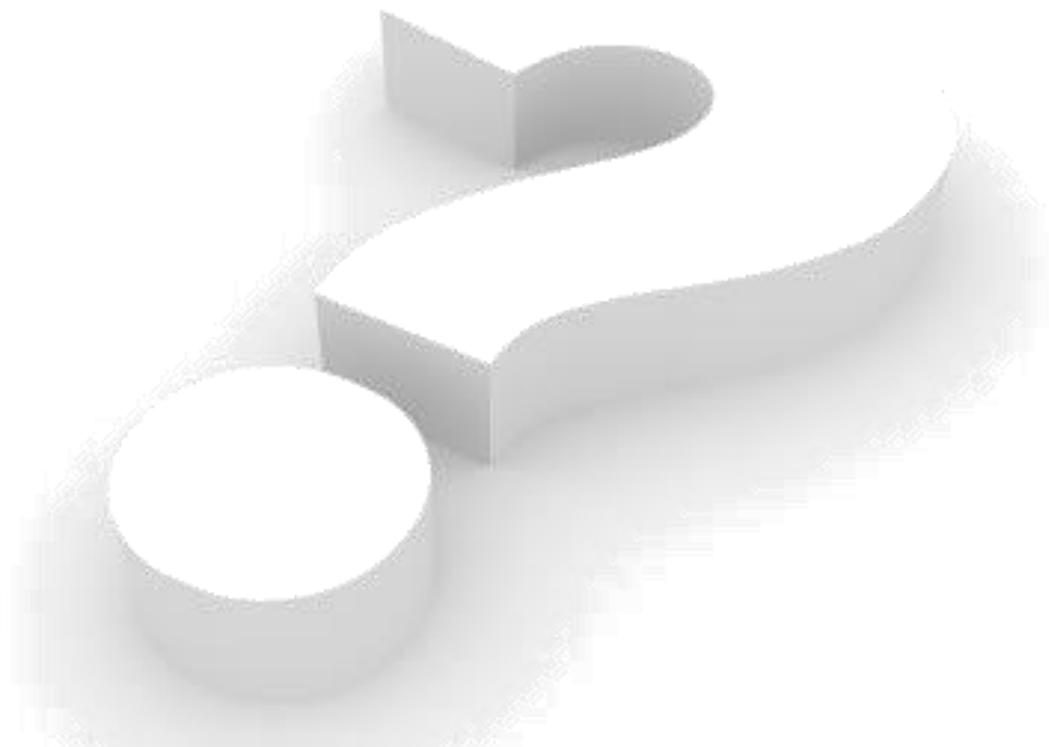
```
keytool -genkey -keystore myKeys -alias aKey  
jarsigner -keystore myKeys MyCoolJar.jar aKey
```

- Votre JAR est maintenant signé.
- En créant un KeyStore d'informations comme ...
  - Nom
  - État
  - Organisme
  - Etc





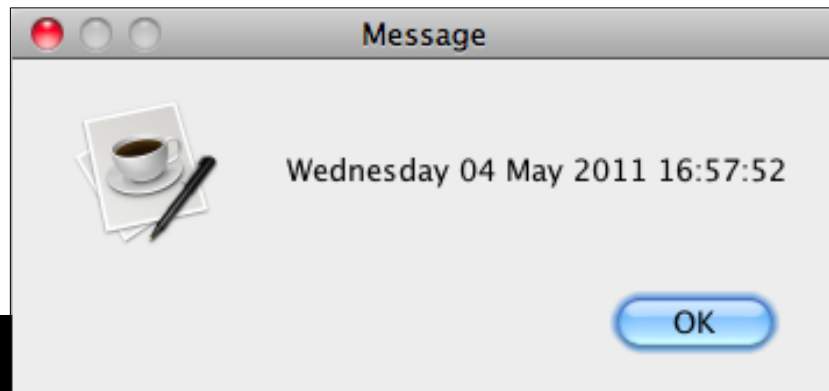
# Questions ?





# Exercises (1/2)

- Créez un nouveau projet Java et nommez-le JWSDemo
  - Créer une classe JWSDemo avec une méthode principale
    - Afficher la date et l'heure actuelles dans une fenêtre contextuelle
      - Utilisez la classe **JOptionPane** pour le faire (voir Javadoc)
      - Formatez la date avant de l'afficher
    - Si tout est correct, le lancement du programme fera apparaître une fenêtre comme celle-ci :





## Exercises (2/2)

- Créez un JAR exécutable avec eclipse pour JWSDemo :
  - Nommez-le : JWSDemo.jar
  - Créer le bon fichier manifeste
- Faites la même chose en utilisant la ligne de commande
- Essayez de le signer !
  - Utilisez l'option de vérification de jarsigner pour vérifier si votre jar est bien signé



Préparation

**Fin**

*Merci de votre attention*