

## Préférences

Université de Nice - Sophia Antipolis  
Richard Grin  
Version 0.4 – 24/1/12

R. Grin

Préférences

page 1

## Description

- ❑ Sauvegarder et récupérer les préférences pour un utilisateur particulier ou pour tous les utilisateurs
- ❑ Cette API est apparue avec le JDK 1.4 (paquetage `java.util.prefs`)
- ❑ Auparavant on utilisait les propriétés
- ❑ La façon de sauvegarder les préférences entre 2 sessions d'utilisation dépend du système d'exploitation ; le développeur n'a pas besoin de savoir comment ça se passe

R. Grin

Préférences

page 2

## Exemples d'utilisation

- ❑ Conserver le chemin d'un répertoire où un utilisateur garde certaines informations ou récupère des fichiers
- ❑ Conserver le nom d'un fichier ou la couleur préférée d'un utilisateur

R. Grin

Préférences

page 3

## Concepts de base

- ❑ Les préférences sont des couples clé-valeur
- ❑ Elles sont rangées dans des nœuds de préférences
- ❑ La convention est de faire correspondre ces nœuds aux paquetages
- ❑ 2 types de préférences correspondant à 2 types de nœuds :
  - utilisateur (concerne un seul utilisateur)
  - système (partagé par tous les utilisateurs)

R. Grin

Préférences

page 4

## Nœud

- ❑ Un nœud correspond à un ensemble de préférences, qui est entreposé de façon persistante dans l'application
- ❑ Chaque nœud contient une *map* dont chaque entrée est le nom d'une propriété et la valeur est la valeur de la propriété
- ❑ Chaque propriété correspond à une préférence de l'utilisateur, par exemple un nom de fichier ou une couleur

R. Grin

Préférences

page 5

## Nom d'un nœud

- ❑ Structure hiérarchique
- ❑ Un nœud a un nom, un chemin absolu, un nom relatif (par rapport à un nœud ancêtre)
- ❑ Le nœud racine de l'arborescence a le nom « / »
- ❑ Les noms absolus sont du type des noms de fichiers Unix : `/nœud1/nœud2/...`
- ❑ Un nom relatif est du type `nœud1/nœud2/...`
- ❑ Le nom relatif d'un nœud par rapport à lui-même est la chaîne vide

R. Grin

Préférences

page 6

## Nom d'un nœud

- ❑ Pour éviter qu'un nom de nœud mal écrit ne provoque une erreur à l'exécution, l'usage est de donner à un nœud le nom du paquetage d'une classe (objet de la classe `Class<T>`)
- ❑ Le plus simple, si ça convient est d'associer au nœud le nom du paquetage de la classe qui utilise la préférence
- ❑ Si plusieurs applications utilisent le même paquetage, on peut distinguer le nom des propriétés en préfixant par le nom de l'application qui l'utilise (ou par un autre moyen)

R. Grin

Préférences

page 7

## Classe `Preferences`

- ❑ La classe `Preferences` correspond à un nœud de préférences
- ❑ On récupère un nœud avec des méthodes `static` (modèle de conception « fabrique ») de la classe `Preferences` ; elles renvoient une instance de `Preferences` :  
`userNodeForPackage(Class<?>)`  
`systemNodeForPackage(Class<?>)`
- ❑ Si le nœud n'existe pas, il est créé

R. Grin

Préférences

page 8

## Exemple

- ❑ `Preferences prefs = userNodeForPackage(this.getClass());`
- ❑ Si la classe est `fr.unice.Classe`, le nom du nœud de la préférence sera le nom du paquetage, en remplaçant les `."` par des `/"` : `/fr/unice`

R. Grin

Préférences

page 9

## Propriétés

- ❑ Quand on a un nœud, on peut ajouter ou lire des préférences comme dans une *map*
- ❑ Les clés qui identifient les préférences sont des `String`
- ❑ Les valeurs des préférences sont de type `String`, `boolean`, `byte[]`, `double`, `float`, `int`, `long`
- ❑ Les entrées de type `String` sont limitées à 8192 caractères ; la taille des tableaux d'octets ne doit pas dépasser 6144

R. Grin

Préférences

page 10

## Lire et écrire une propriété

- ❑ Des méthodes `getXXX` permettent de récupérer la valeur d'une préférence
- ❑ Des méthodes `putXXX` permettent de ranger la valeur d'une préférence
- ❑ Les méthodes sont `putInt`, `putBoolean`, `putFloat`, `putLong`, `putByteArray`, et les méthodes `get` correspondantes
- ❑ `get()` et `put(String)` correspondent à une préférence de type `String`

R. Grin

Préférences

page 11

## Sauvegarder une préférence

```
// Récupère le nœud
Preferences prefs =
    Preferences.userNodeForPackage(
        getClass());
// Mets la valeur de la préférence pref1
prefs.put("pref1", "v11");

// Pour les nœuds « système » :
Preferences prefs =
    Preferences.systemNodeForPackage(
        getClass());
```

R. Grin

Préférences

page 12

## Lire une préférence

- ❑ On doit donner une valeur par défaut ; cette valeur sera retournée si la préférence ne peut être lue :

```
String valeurPropriété =
    prefs.get("pref1", "defaut");
```

R. Grin

Préférences

page 13

## Supprimer une préférence

- ❑ La méthode `remove(String)` supprime la préférence du nom passé en paramètre
- ❑ `clear()` supprime toutes les préférences du nœud ; elle peut lancer une exception contrôlée par le compilateur

R. Grin

Préférences

page 14

## Exemple

```
try {
    prefs.clear();
}
catch (BackingStoreException e) {
    // Problème avec le système
}
```

R. Grin

Préférences

page 15

## Sauvegarde des préférences

- ❑ La sauvegarde des préférences entre les sessions de travail est automatique et dépend du système dans lequel on travaille
- ❑ Sous Windows les préférences sont sauvegardées dans le registre
- ❑ Sous Unix, dans le système de fichiers
- ❑ On peut aussi importer ou exporter des préférences dans un fichier XML avec des `{In|Out}putStream`

R. Grin

Préférences

page 16