Applications réflexives

TD 3 gestionnaire dynamique d'activités contrôlées

Introduction

Le but du programme que vous allez réaliser est de charger, d'utiliser et de surveiller des classes d'activité (Runnable) : un gestionnaire dynamique d'activités contrôlées. La mise en œuvre du système se fait en deux temps pour une activité : chargement de la classe, lancement d'une activité. Un bilan global complète le système.

Les problèmes de concurrence ne seront pas abordés ici, un menu séquentialisant les opérations sur les runnables.

Exercice 1. Chargement dynamique et lancement d'une activité

Une classe d'activité (implémentant Runnable) développée par un développeur indépendant est transmise à votre logiciel, pour ce tp, « manuellement » : le fichier compilé (dans son répertoire/package) doit être placé dans le répertoire standard ***myclasses*** dont vous choisirez l'emplacement (dans votre workspace par exemple). A partir de là, un URLClassLoader charge cette classe et elle est déclarée à `RunnableManager` (addRunnable).
Le lancement d'une activité au choix parmi celles chargées doit exécuter le run() par invocation dynamique.

L'application est donc lancée avec à priori un `RunnableManager` vide d'activité (on pourrait en option lire et charger les classes présentes dans ***myclasses***). Un URLClassLoader est créé pointant le répertoire ***myclasses***. Un menu permet ensuite de tester le fonctionnement. Le choix de l'activité à lancer se fera en listant dans un premier temps les activités présentes (toStringue) et en saisissant le numéro d'activité à lancer

Les cas d'erreurs à prendre en compte (affichage d'un message) sont : la classe demandée n'est pas trouvée ou n'implémente pas Runnable (option 1), le numéro de classe demandée n'est pas dans les bornes (option 2).

Exercice 2. Surveillance et proxy

RunnableManager, lorsqu'on lui déclare une activité, créé un `InvocationHandler` qui comptera le nombre d'invocation de la méthode run() de Runnable : `RunnableInvocationCpt` joue ce rôle . Le lancement dans un thread d'une activité doit, au passage par le run(), incrémenter le compteur d'activations que gère le `RunnableInvocationCpt`. On peut ensuite demander à RunnableManager un bilan d'activations des activités. La méthode *bilan*() joue ce rôle et renvoie une String directement affichable.

Vous trouverez sur le commun un « jeu d'essai » pour myclasses ; les classes *runnables.Coucou* et *runnables.Salut* affichent dans leur run() respectivement « Coucou » et « Salut ».