



Eclipse RCP (Rich Client Platform)

Développer un client riche basé sur la plate-forme Eclipse RCP

Objectifs

- Comprendre l'architecture de plugins d'Eclipse en profondeur
- Réaliser une application client riche basée sur Eclipse RCP avec SWT et JFace
- Connaître les best practices du développement de plugins
- Savoir tester, construire et packager le produit final

Contenu

Présentation

- Positionnement d'Eclipse RCP dans la mouvance Client Riche
- Distinctions entre SWT, JFace et Eclipse RCP
- Comparaison aux autres toolkits Java: Swing

Démarrer un projet RCP

- Présentation de PDE (Plugin Development Environment)
- Créer un projet de Plugin Eclipse
- Présentation et mise en oeuvre du debugging de plugins (Eclipse self-hosted)
- Apprendre à utiliser les vues spécifiques à PDE

Comprendre l'infrastructure Eclipse

- Présentation de la structure d'un plugin
- Définition d'une contribution à la plateforme et présentation des possibilités
- Présentation du socle dynamique d'Eclipse: OSGi, Equinox
- Best practices à respecter dans les graphes de dépendances
- Mise en oeuvre de tests unitaires par utilisation de PDE JUnit

Utiliser la bibliothèque de composants graphiques SWT

- Utilisation des widgets basiques
 - Boutons, labels
 - Champs de saisie: champs texte, cases à cocher, radios, widgets de type liste, ...
- Placement des composants: systèmes de Layouts
- Réaction aux événements graphiques: Events et Listeners

Comprendre et utiliser JFace

- Comprendre la philosophie de JFace et savoir quand l'utiliser
- Architecture interne de JFace : mécanisme de Viewer
- Utilisation des composants JFace
 - Dialogues, fenêtres
 - Menus, barres d'outils, mise en oeuvre du framework d'Actions
 - Assistants, fenêtres de préférences
 - Affichage de texte riche, assistance à la saisie

Packager et livrer l'application

- Présentation du mécanisme de Feature
- Mise en oeuvre d'un site de mise à jour (Update Site)
- Création d'un produit (Product)
- Industrialiser le processus de livraison

Contribuer à l'infrastructure Eclipse

- Contribuer sous forme d'entrées de menu ou de boutons: les Actions
- Créer et déclarer une Vue (View) pour représenter un état
- Lier la Vue à un état particulier du Workbench: SelectionProviders et SelectionListeners
- Créer et déclarer un Editeur (Editor)
- Lier l'éditeur à des Vues possiblement ouvertes (type Outline)
- Créer une Perspective, améliorer une Perspective existante
- Utilisation de fenêtres de dialogue (Dialogs) et d'assistants (Wizards)
- Création de pages de préférences

Techniques plus avancées

- Utilisation de méta-données (Properties) sur les ressources du Workbench et affichage de celles-ci dans les vues de type PropertiesView
- Définition de natures de ressource: mécanisme de Natures, incluant les Builders et Markers
- Ajout d'aide en ligne
- Gestion de l'internationalisation (i18n)
- Découverte de l'API Eclipse UIForms
- Améliorer la réactivité des interfaces: gestion des Threads, retours dans la tâche graphique

Intégration dans une architecture J2EE

- Présentation d'une architecture classique client riche
- Réalisation d'appels de services type SOA distants, utilisation de Spring Remoting
- Mise en oeuvre de best practices de découpage technique

Durée
3 jours (21h)

Répartition
40% théorie
60% pratique

Pré-requis
Connaissance de Java

Public
Architecte
Développeur
Chef de projet

Tarif (déjeuners inclus)
1 550 € (HT)

Sessions 2008 - Paris
4 au 6 février
7 au 9 avril
16 au 18 juin
4 au 6 août
6 au 8 octobre
1 au 3 décembre

Intra-entreprise sur demande

Renseignements et inscription
+33(0)1.45.26.19.15
training@zenika.com
www.zenika.com