Eclipse 4 RCP application Development
COURSE OUTLINE

Description

The Eclipse 4 RCP application development course will help you understand how to implement your own application based on the Eclipse 4 platform. The Eclipse 4 release significantly changes the programming model for Eclipse RCP development. The course explains the existing core frameworks in Eclipse, e.g. SWT, databinding, plug-in architecture, extension point, etc. as well as the new Eclipse 4 platform concepts, e.g. dependency injection, declarative styling, service consumption, the application model, the rendering framework, etc. This course also covers application lifecycle questions, like application updates and headless builds. Participants will create a complete standalone application to familiarizing themselves with the different concepts.

Audience

Software developers and architects

Prerequisites

Experience with Eclipse and strong knowledge of Java

Duration and Format

Five-day instructor-led class with at least 50% hands-on labs and tutorials

Outline

1. **Introduction into Eclipse and Eclipse 4**
   - Components of the Eclipse platform
   - Eclipse license
   - Internet information sources
2. **Eclipse architecture**
   - Software components and configuration files
   - Important user interface components
3. **Deployment of an Eclipse product**
   - Product configuration file
   - Feature projects
   - Branding and product export
   - Run configuration
4. **Eclipse 4 application model**
   - Application model and model components
   - Model editor
   - Naming schema for ID's

5. **Dependency injection and annotations**
    - Overview dependency injection
    - Dependency injection framework in Eclipse
    - Field, method and constructor dependency injection
    - Behavior annotations
    - Application lifecycle annotations
6. **Scope of injection**
    - IEclipseContext
    - Injection search strategy
    - Creation of injectable objects
    - Model elements and dependency injection
7. **Modularity of the Eclipse platform with OSGi**
    - Plug-ins and bundles
    - Definition of dependencies between plug-ins
    - Fragment projects
    - Overview Equinox
    - OSGi framework start configuration
    - OSGi console
8. **OSGi services**
    - Services and service registry
    - Publishing services via OSGi declarative services
    - Usage of services in Eclipse 4
    - Auto-start and start level
9. **User interface development with SWT**
    - SWT widgets and event handling
    - SWT layout manage
    - User interface builder: SWT Designer
    - Custom widgets and Nebula widgets
10. **Introduction JFace**
    - Overview JFace components
    - SWT resource management
    - Control decorations for user feedback
    - Introduction into the Viewer framework (LabelProvider, ContentProvider, ComboViewer)
    - Handling Viewer selection

11. **JFace TableViewer and TreeViewer**
    - ColumnLabelProvider and CellLabelProvider
    - Editable tables
    - Sorting, filtering, layouts and own label provider

12. **Commands, Handlers, Menus and Toolbars**
    - Contributing to the menu and the toolbar
    - Handling of popup menus
    - Scope of handlers and core expressions
    - Defining keybindings

13. **Dialog and Wizards**
    - SWT standard dialogs
    - JFace dialogs
    - JFace wizards

14. **JFace Data Binding**
    - Introduction into databinding
    - Observing properties
    - Conversion, validation and update strategies
    - Databinding for JFace Viewers
    - Master / detail bindings

15. **Platform services and interaction of components**
    - Service overview
    - Part service
    - Model service
    - Selection service
    - Command and Handler service

16. **Editor handling in Eclipse 4**
    - Definition of parts which behave as editors
    - Using services to interact with parts

17. **Accessing and extending the Eclipse context**
    - Accessing the context
    - Extending the Eclipse context with own objects
    - Using dependency injection to create own objects

18. **Concurrent UIs**
    - SWT threading
    - Avoiding invalid thread access
    - Asynchronous processing with the Eclipse API

19. **Event communication**
    - Usage of the event services
    - Sending and receiving events

20. **Application lifecycle**
    - Application lifecycle registration
    - Lifecycle annotations

21. **Context functions**
    - Usage of context functions
    - Using dependency injection with context functions

22. **Modularity for Eclipse 4 applications**
    - Contributing to the application model
    - Static model contributions with fragments
    - Dynamic model contributions with processors

23. **Settings and preferences**
    - Configuration area and workspace
    - Persistence of the Eclipse application
    - Part persistence
    - Dependency injection for preference values

24. **Declarative styling with CSS**
    - Definition of styles and themes, colors and gradients
    - Styling specific widgets
    - Dynamic style switching at runtime
    - Using the CSS Spy tooling

25. **Internationalization (i18n)**
    - Adding support for multiple languages
    - Usage of fragment projects
    - Outlook: translation services in Eclipse 4

26. **Updates with Eclipse p2**
    - Overview: the p2 provisioning platform
    - Creating p2 update sides
    - Updating products and features via the p2 API

27. **Target Platform**
    - Definition of development components
    - Creation of target platform definitions

28. **Migrating Eclipse 3.x applications**
    - Running Eclipse 3.x applications on top of Eclipse 4
    - Mixing Eclipse 3.x and Eclipse 4.x components
    - Discussion: Migration path for existing applications

29. **Definition of own annotations for dependency injection**
    - Definition of new annotations
    - Evaluation of new annotations

30. **Creating and evaluating extension points**
    - Eclipse extensions and extension points
    - Accessing existing extensions
    - Creating and evaluating a new extension point

31. **The Renderer framework**
    - Purpose of the Renderer framework
    - Define your own renderer
    - Outlook: Using an alternative renderer
    - Outlook: Extending the application model

32. **Best practices and tips & tricks**