

## Eclipse 4 RCP training agenda

- Introduction to Eclipse and Eclipse 4
  - Components of the Eclipse platform
  - Eclipse 3.x in comparison with Eclipse 4.x
  - Eclipse license
  - Internet information sources
- Eclipse architecture
  - Software components
  - Configuration files (plugin.xml, MANIFEST.MF)
  - Extensions and extension points
  - Important user interface components
- Deployment of an Eclipse product
  - Product configuration file
  - Feature projects
  - Branding and product export
  - Run configuration
  - Problem analysis during export
- Eclipse 4 application model
  - Application model and model components
  - Model editor
  - Naming schema for ID's
- Dependency injection and annotations
  - Overview dependency injection
  - Dependency injection framework in Eclipse
  - Field, method and constructor dependency injection
  - Behavior annotations
  - Application lifecycle annotations
- Commands, Handlers, Menus and Toolbars
  - Contributing to the menu and the toolbar
  - Handling of popup menus
  - Scope of handlers and core expressions
  - Defining keybindings
- Scope of injection
  - IEclipseContext
  - Injection search strategy
  - Creation of injectable objects

- Model elements and dependency injection
- Modularity of the Eclipse platform with OSGi
  - Plug-ins and bundles
  - Definition of dependencies between plug-ins
  - Fragment projects
  - OSGi framework start configuration and usage of the OSGi console
- OSGi services
  - Services and the OSGi service registry
  - Publishing services via OSGi declarative services
  - Usage of services in Eclipse 4
  - OSGi declarative service definition with annotations
- User interface development with SWT
  - Overview Standard Widget Toolkit
  - SWT event handling
  - SWT layout manager: FillLayout, RowLayout and GridLayout
  - User interface builder: SWT Designer
  - Custom widgets and Nebula widgets
- Introduction JFace
  - Overview Jface components
  - SWT resource management
  - Control decorations for user feedback
  - Introduction into the Viewer framework (LabelProvider, ContentProvider, ComboViewer)
  - Handling Viewer selection
- Jface TableViewer and TreeViewer
  - ColumnLabelProvider and CellLabelProvider
  - Editable tables
  - Sorting, filtering, layouts and own label provider
- Dialogs and Wizards
  - SWT standard Dialogs
  - JFace Dialogs
  - Jface Wizards
- Declarative styling with CSS
  - Introduction into CSS
  - Definition of styles and themes, colors and gradients
  - Styling specific widgets
  - Dynamic style switching at runtime
  - Using the CSS Spy tooling

- Platform services and interaction of components
  - Service overview
  - Part service
  - Model service
  - Selection service
  - Command and Handler service
- Editor handling in Eclipse 4
  - Comparison Views and Editors
  - Getting parts which behave as editors
  - Using services to interact with parts
- Accessing and extending the Eclipse context
  - Accessing the context
  - Extending the Eclipse context with own objects
  - Using dependency injection to create own objects
- Settings and preferences
  - Configuration area and workspace
  - Persistence of the Eclipse application
  - Part persistence
  - Dependency injection for preference values
- Modularity for Eclipse 4 applications
  - Contributing to the application model
  - Static model contributions with fragments
  - Dynamic model contributions with processors
- Internationalization (i18n)
  - Adding support for multiple languages
  - Usage of fragment projects
  - Outlook: translation services in Eclipse 4
- Concurrent UIs
  - SWT threading
  - Avoiding invalid thread access
  - Asynchronous processing with the Eclipse API
- Jface Data Binding
  - Introduction into databinding
  - Observing properties
  - Conversion, validation and update strategies
  - Databinding for Jface Viewers
  - Master / Detail bindings

- Target Platform
  - Definition of development components
  - Creation of target platform definitions
- 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
  
- Definition of own annotations for dependency injection
  - Definition of new annotations
  - Evaluation of new annotations
  - Use cases
  
- Creating and evaluating extension points
  - Eclipse extensions and extension points
  - Accessing existing extensions
  - Creating and evaluating a new extension point
  
- The Renderer framework
  - Purpose of the Renderer framework
  - Define your own renderer
  - Outlook: Using an alternative renderer
  - Outlook: Extending the application model
  
- Building Eclipse applications with Maven \*
  - High level overview of Tycho
  - Building plug-ins, features, products and update sites
  - Executing plug-in unit tests with Tycho

## Best practices and tips & tricks

\* Topics only included in the online training material.