

Java Lab 2008/11/17

Client

Prof. Dr. Bernhard Humm

Version: 1.0
Date: 2007/11/11

1 Preparation

1. Open my homepage www.fbi.h-da.de/~b.humm and all relevant documents
2. Update the project from the Subversion server

In this lab session, you will implement the dialogs of the library system:

1. Identify the dialogs of the system with respect to the use cases.
2. Organize in your team as to who is implementing which dialog.

2 Presentation

Implement the visual representation of the dialog:

1. Draft the dialog statics on a sheet of paper.
2. Implement a class `<Dialog>Panel` that extends `javax.swing.JPanel` in package `de.h_da.library.<dialog>` of project `Library*-app-client` which contains the visual representation of the dialog
Tipp: use the wizard *New* → *File, Folder* → *Java GUI forms* → *JPanel Form*
3. Design the dialog statics using the NetBeans GUI builder
4. Modify the auto-generated widget names (e.g., `jButton1`) to self-explanatory names (e.g., `createButton`)
5. Generate getters and setters

3 DialogKernel

Implement the dialog kernel:

1. Create a new Java class `<Dialog>Dialog` that extends `com.sdm.quasar.client.core.dialog.AbstractFeature`
2. Implement an instance variable `visualRepresentation` which refers to the `<Dialog>Panel`
3. Implement business methods that implement actions to be performed by the dialog. Implement necessary instance variables, e.g., for use case proxies and for transfer objects (entity classes).
4. Implement data binding methods

5. Implement `AbstractFeature.init` to perform the name server look-up
6. Implement `AbstractFeature.prepare`:
 - create the visual representation
 - prepare the dialog for embedding
 - register listeners
 - initialize presentation
7. Implement `AbstractFeature.dispose` to remove the visual representation
8. Tipp: refer to the implementation of `de.h_da.library.usecase1.UseCase1Dialog`

4 Portal

Embed your dialog into the `LibraryPortal`:

1. In class `LibraryPortalFrame` of package `de.h_da.library.portal`, add a constant for your dialog with the dialog menu entry.
2. In `LibraryPortalFrame.newNavigationTreeModel`, add a new `DefaultMutableTreeNode`
3. In `LibraryPortalDialog.prepare`, add a `TreeSelectionListener`
4. In class `LibraryClientApplication` of package `de.h_da.library`, instantiate a configuration for your dialog (method `start`).
5. Add your dialog to the `dialogDefinitionService`
6. Tipp: Organize in your team as to avoid conflicts

5 Test

Test your dialog by running `LibraryClientApplication` in package `de.h_da.library`

6 Optional Tasks

Extend the client and the uses cases with more sophisticated functionality.