GUI Vorlesung 2019

Übung 8 - Architektur

Beschreibung

Ziel der Übung ist es, die bekannte Anwendung in eine Schichtenarchitektur zu überführen. Die Schichten werden dabei als eigene Komponenten umgesetzt. Technisch erfolgt dies als Multi-Projekt-Gradle-Build

Hinweis: Eine typische Projektstruktur für Gradle sieht wie folgt aus:

```
<Wurzelprojekt>
   | - build.gradle
   | - settings.gradle
   | - <Subprojekt 1>
      | - build.gradle
      | - src
         | - main
            | - java            //Java Klasse
            | - resources       //sonstige Dateien, z.B.: FXML, Icons…
   | - <Subprojekt 2>
      | - build.gradle
      | - src
         | - main
            | - java
            | - resources
```

Aufgabe 1

Holen Sie sich das Starter-Projekt von https://github.com/dominikhaas/Vorlesung-GUI-2019/tree/master/codebase/u08-architecture-starter und öffnen Sie in IntelliJ die Datei build.gradle (als Projekt).

Starten Sie im Gradle-Fenster auf der rechten Seite einen build.

Führen Sie den Gradle-Task run aus.

Ergebnis: Die Anwendung startet und die bekannte Oberfläche wird angezeigt.

Aufgabe 2

Zerlegen Sie das existierende Projekt in Schichten. Gehen Sie dabei wie folgt vor:

1. Bestimmen Sie die drei typischen Schichten aus der Vorlesung. Legen Sie für jede noch fehlende Schicht ein entsprechend benanntes Modul vom Typ „Gradle -> Java“ an. Das sind ihre Subprojekte.
2. Leeren Sie die neuen gradle.build Dateien bis auf den Abschnitt dependencies.
3. Verschieben Sie die Klassen aus dem Ordner unsorted in die jeweilige Schicht.
   Hinweis: Wenn Sie auf Klassen aus einem anderen Projekt zugreifen wollen, so müssen Sie die Abhängigkeit (dependency) hinzufügen. Sie können dies über den Eintrag
   „`compile project(':<Name des anderen Subprojekt>')`“ im Abschnitt dependencies tun.
4. Bauen Sie das Projekt und führen Sie es aus.

Aufgabe 3

Erzeugen Sie in Ihrer Schicht „Anwendungskern“ eine neue saubere Komponente zur Protokollierung mit folgendem Interface:

```
public interface ProtocolService {
    void writeProtocol(ProtocolEntry entry);
}
```

Gehen Sie dabei wie folgt vor:

1. Erzeugen Sie eine neues package.
2. Legen Sie das Interface an.
3. Denken Sie sich eine Klasse ProtocolEntry aus, mit mindestens zwei Feldern.
4. Schreiben Sie eine Dummy-Implementierung für den Service
5. Teilen Sie die Komponente in Schnittstelle und Implementierung auf.
6. Verwenden Sie den Service im Service DataSeriesServiceImpl
7. Prüfen Sie, mit den Folien der Vorlesung, ob Sie alle Regeln zum Thema „Legale und illegale Abhängigkeiten“ einhalten.

Aufgabe 4 – optional

Verwenden Sie das Java Modul System um Sichtbarkeiten zu steuern und für den Service-Lookup. Gehen Sie dabei wie folgt vor:

1. Ergänzen Sie die Java-Module (nicht JavaFX) um den unten angeführten Abschnitt.
2. Fügen Sie in jedes Subprojekt unter src/main/java eine Datei module-info.java ein.
3. Steuern Sie die Sichtbarkeiten so, dass die API sichtbar ist und die Implementierung nicht
4. Registrieren Sie die Implementierung des Services
5. Entfernen Sie die direkte Verwendung des Services DataSeriesServiceImpl in der GUI. Verwenden Sie stattdessen:
   ```
   DataSeriesService service =
   ServiceLoader.load(DataSeriesService.class).findFirst().get();
   ```
6. Passen Sie die modul-info.java der GUI so an, dass alle Pakete erreichbar sind (open) und signalisieren Sie die Verwendung des Services

Ergänzung für den Modul-Build (nur in Java-Modulen, nicht JavaFX):

```
ext.moduleName = '<module name>'

compileJava {
    inputs.property("moduleName", moduleName)
    doFirst {
        options.compilerArgs = [
                '--module-path', classpath.asPath,
        ]
        classpath = files()
    }
}
```

Module-info.java für das GUI-Projekt

```
module de.throsenheim.gui.architecture.gui {
    requires <Module busines>;
    requires  javafx.controls;
    requires  javafx.fxml;
    requires java.logging;

    opens de.throsenheim.gui.u08.chart;
    opens de.throsenheim.gui.u08.header;
    opens de.throsenheim.gui.u08.status;
    opens de.throsenheim.gui.u08;

    uses DataSeriesService;
}
```