

Aufgabe: Command-Design-Pattern und Versionierung

1. Erweitern Sie die Adressverwaltung, deren GUI in der letzten Aufgabe realisiert wurde, mit den zugehörigen Funktionalitäten mit Hilfe des Command-Design-Patterns. Eine Undo-Funktionalität soll mindestens für die Aktion „Löschen“ realisiert werden. Nicht alle „undo“ sind sinnvoll - z.B. ein „undo“ für „Öffnen“ und „Speichern“ ist nicht sinnvoll. Zur Vereinfachung kann **das gegebene Model** verwendet und erweitert werden. Es kooperiert mit einer „JTable“. Beim verwendeten JTable-Objekt muss „unser“ Model via „setModel“ registriert werden.
2. Bei der Auswahl der Adressverwaltungsdatei für Öffnen und Speichern soll das zuletzt benutzte Verzeichnis angeboten werden. Dies kann man mit Hilfe der Klasse „Preferences“ realisieren.
3. Die Persistenz der Daten soll über Streams und Objektserialisierung (Klassen „ObjectInputStream“ und „ObjectOutputStream“) erfolgen.
4. Schreiben Sie „javadoc“-Kommentare für alle Klassen und Methoden und erzeugen Sie damit die zugehörige Dokumentation.
5. Stellen Sie Ihr Projekt (nur die Source-Files!) unter die Versionskontrolle von GIT.
6. Üben Sie „commit“, „diff“ usw.
7. Üben Sie über die FAU-Box, „Beehive“, Ihr ADS-Home-Verzeichnis oder über einen Git-Server die Zugriffe auf eine „remote repository“ („push“, „pull“).
8. Verwenden Sie GIT, Javadoc und ggf. CommandDP immer auch bei zukünftigen Applikationsentwicklungen.