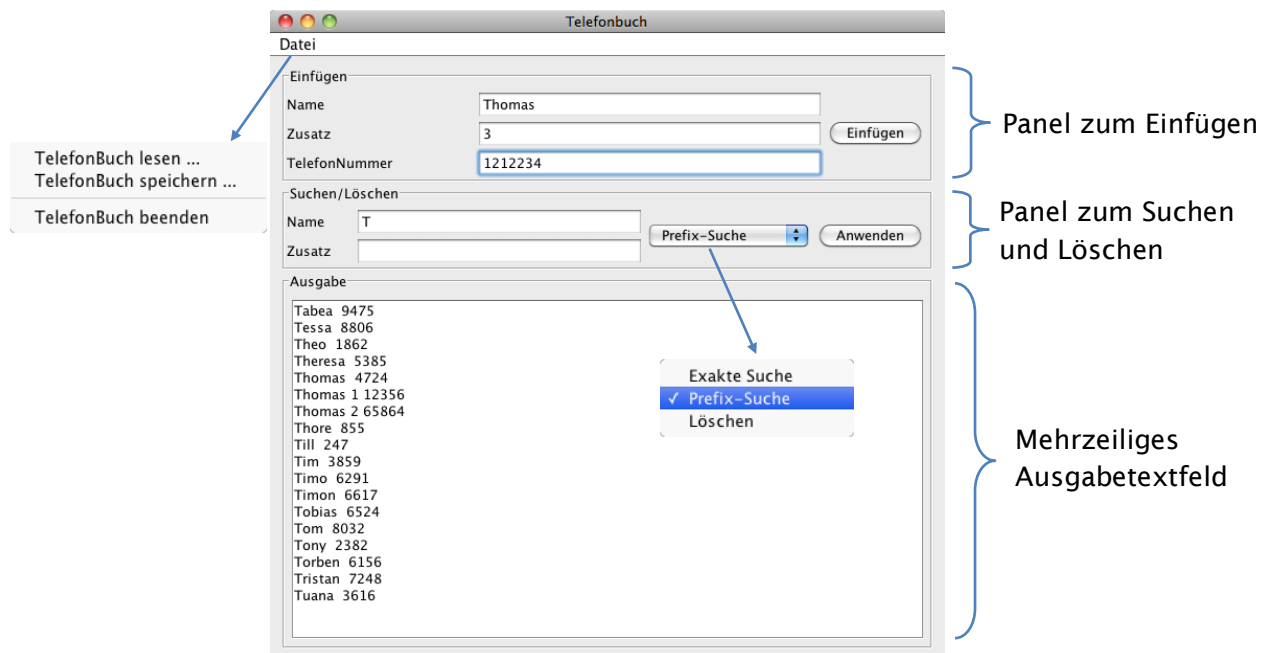


Aufgabenblatt 10 – Telefonbuch



Mit dieser Anwendung können Telefonbücher erstellt und benutzt werden. Ein Eintrag im Telefonbuch besteht aus Namen und Telefonnummer. Um eindeutige Namen zu erhalten, können die Namen mit einem Zusatz versehen werden (im Beispiel oben „Thomas“ + „3“).

Die Grafische Benutzeroberfläche bietet folgende Funktionalität an:

- Über das Menü kann ein Telefonbuch geladen bzw. das aktuelle Telefonbuch gespeichert werden (JFileChooser-Dialog).
- Über den Menüeintrag „TelefonBuch beenden“ kann die Applikation beendet werden. Der Benutzer muss über ein Dialog (confirm dialog) bestätigen, ob die Anwendung tatsächlich verlassen werden soll. Das veränderte Telefonbuch wird nicht abgespeichert.
- Freiwillig: durch Schließen des Fensters kann die Applikation ebenso beendet werden. Verwenden Sie als EventListener einen WindowAdapter (java.awt.event.WindowAdapter) und überschreiben Sie die Methode windowClosing.
- Im Einfügen-Panel können neue Einträge in das Telefonbuch erfolgen. Das Einfügen ist nur möglich, wenn ein Name mit dem angegebenen Zusatz noch nicht existiert. Ist das Einfügen nicht möglich, erhält der Benutzer eine entsprechende Mitteilung (message dialog).
- Im zweiten Panel kann über eine Combobox gesteuert werden, ob eine exakte Suche, eine Präfix-Suche oder Löschen durchgeführt werden soll. Bei der exakten Suche wird das Telefonbuch nach dem eingegebenen Namen mit Zusatz durchsucht. Das Ergebnis der Suche (Name, Zusatz und Telefonnummer) wird im Ausgabefeld ausgegeben. Falls der Eintrag nicht gefunden wird, erscheint eine entsprechende Mitteilung (message dialog).

- Bei der Präfixsuche werden alle Einträge, die mit dem eingegebenen String beginnen, im Ausgabefeld ausgegeben. Im Beispiel oben sind alle mit „T“ beginnenden Einträge aufgelistet.
- Beim Löschen muss der eingegebene Name und Zusatz mit dem zu löschenden Eintrag übereinstimmen. Der Benutzer wird benachrichtigt (message dialog), ob das Löschen erfolgreich war.

Klasse TelefonBuch

Ein Telefonbuch soll als Java Collection `TreeMap<Key,Value>` realisiert werden. Machen Sie sich zunächst mit der Klasse `TreeMap` aus der Java API anhand des beiliegenden Programm `TreeMapDemo.java` vertraut (siehe dazu auch im Skript die Seite 11-10 bis 11-16).

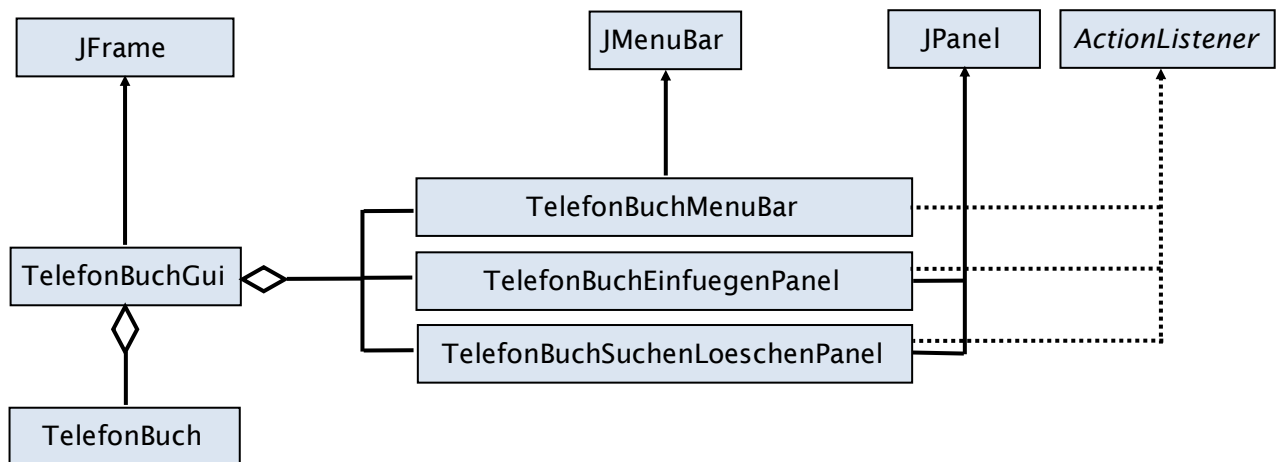
Implementieren Sie dann die Klasse `TelefonBuch` mit den folgenden Methoden:

- `boolean insert(String name, String zusatz, String telNr);`
fügt in das Telefonbuch einen neuen Eintrag mit Name, Zusatz und Telefonnummer ein.
Wichtiger Hinweis: Name und Zusatz werden mit einem Leerzeichen getrennt zu einem String verknüpft, der dann als Schlüssel für `TreeMap` verwendet wird. Beispiel: `insert("Peter", "KN", "1234")` fügt in die `TreeMap` den Eintrag mit dem Schlüssel (key) "Peter KN" und der Telefonnummer (value) "1234" ein. Existiert der Eintrag bereits, dann kann nicht eingefügt werden, und es wird `false` zurückgeliefert. Sonst kann eingefügt werden und es wird `true` zurückgeliefert.
- `boolean remove(String name, String zusatz);`
löscht im Telefonbuch den entsprechenden Eintrag. Konnte der Eintrag gelöscht werden, dann wird `true` und sonst `false` zurückgeliefert.
- `String exactSearch(String name, String zusatz);`
sucht im Telefonbuch den entsprechenden Eintrag. Konnte der Eintrag gefunden werden, dann wird die Telefonnummer und sonst `null` zurückgeliefert.
- `List<String> prefixSearch(String str);`
liefert alle Einträge, die mit dem String `str` beginnen, als Liste von Strings zurück. Jeder String besteht dabei aus Name, Zusatz und Telefonnummer (durch Leerzeichen getrennt).
- `void read(File f);`
liest ein Telefonbuch von der Datei `f`. Auf der Webseite finden Sie ein Beispiel-Telefonbuch mit 420 Einträgen.
- `void save(File f);`
speichert das Telefonbuch in die Datei `f` ab.

Verwenden Sie die auf der Web-Seite vorhandene Datei `TelefonBuch.java`, die bereits einen Teil des Codes enthält. Insbesondere sind die Methoden `read` und `save` bereits implementiert. Die Klasse besitzt auch eine `main`-Methode, die einige Testfälle durchspielt.

Vervollständigen Sie die Klasse und führen Sie ausgiebige Tests durch.

Klasse TelefonBuchGUI und ihre Hilfsklassen



Damit die Klasse TelefonBuchGui nicht zu umfangreich und damit zu unübersichtlich wird, wird sie aus kleineren Hilfsklassen aufgebaut.

Die Klasse TelefonBuchEinfügenPanel ist für den Aufbau des Panels zum Einfügen zuständig. Außerdem enthält sie einen Beobachter (ActionListener), der bei Klicken auf den Einfügen-Button das Einfügen der eingegebenen Daten in das Telefonbuch vornimmt.

Die Klasse TelefonBuchSuchenLoeschenPanel ist für den Aufbau des Panels zum Löschen und Suchen zuständig. Diese Klasse verwaltet auch das mehrzeilige Ausgabefeld. Sie enthält einen Beobachter, der für das Suchen und das Löschen zuständig ist und der die Suchergebnisse in das Ausgabefeld ausgibt.

Die Klasse TelefonBuchMenuBar baut das Menü auf und enthält einen Beobachter, der das Lesen oder Speichern des Telefonbuchs bzw. das Beenden des Programms übernimmt. Verzeichnis und Datei wird dabei über einen JFileChooser-Dialog gewählt.

Die Klasse TelefonBuchGui baut ihr Hauptfenster mit Hilfe dieser Klasse zusammen. Dazu werden die Konstruktoren der drei Hilfsklassen aufgerufen. Diese bekommen jeweils eine Referenz auf das Telefonbuch übergeben, damit die Beobachter auf das Telefonbuch zugreifen können.

Auf der Web-Seite finden Sie für diese vier Klassen Java-Dateien, die bereits einen kleinen Teil des Codes enthalten.