

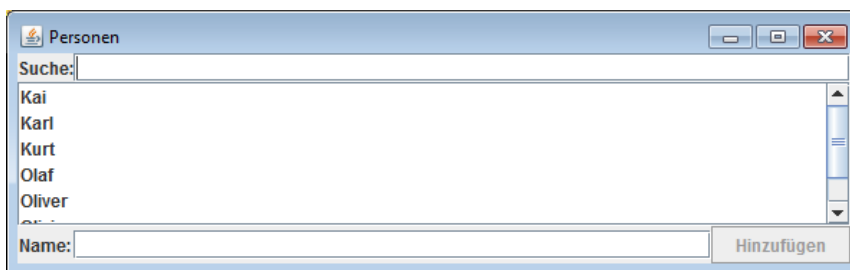
# Praktikum Programmieren II (Java)

Sommersemester 2010

18. Mai 2010

## Übung 4.1 Personenliste

Entwerfen Sie eine grafische Oberfläche, die analog zu folgendem Screenshot eine Liste mit den Vornamen „Kai“, „Karl“, „Kurt“, „Olaf“, „Olivia“, „Oliver“ und „Otto“ enthält:



Bezüglich der grafischen Realisierung sollen möglichst folgende Eigenschaften erfüllt sein, die sich am Besten über ein GridBagLayout erreichen lassen:

- Die Auswahlliste (innerhalb eines JScrollPane) wächst horizontal und vertikal;
- die beiden Textfelder wachsen horizontal;
- die beiden Textlabels und die Schaltfläche wachsen nicht.

Die Anwendungslogik soll wie folgt aussehen:

- Die Schaltfläche „Hinzufügen“ ist aktiv, wenn der Inhalt im Textfeld zu „Name:“ nicht leer ist.
- Beim Klick auf die aktive Schaltfläche oder beim Drücken von Enter im Textfeld zu „Name:“ wird dieser Name der Auswahlliste hinzugefügt. Hierbei wird das Suchfeld geleert und alle Vornamen werden angezeigt.
- Beim Tippen von Buchstaben im Textfeld zu „Suche:“ werden während des Tippens die Namen angezeigt, die mit dem aktuell eingegebenen Text beginnen, d. h. beim Tippen von „K“ erscheinen von den Namen aus dem Screenshot nur noch „Kai“, „Karl“ und „Kurt“, nach dem Tippen von „Ka“ nur noch „Kai“ und „Karl“.
- Bei leerem Suchfeld werden alle Namen angezeigt.
- Beim Klick auf einen Namen in der Auswahlliste soll dieser ins Suchfeld übernommen werden und die Auswahlliste auf diesen Namen reduziert werden.

### Tipps: (nicht zwingend, sondern beispielhaft)

- Erstellen Sie eine Klasse, die `ListModel` implementiert (z. B. durch Erben von `AbstractListModel`), mit der die Auswahlliste verknüpft wird.
- Diese kann als Attribute einen `SortedSet` mit allen Vornamen und eine `List` mit den auf den Suchstring passenden Namen enthalten.
- Implementieren Sie die zwingend zu implementierenden Methoden `getElementAt` und `getSize` durch Rückgriff auf die passenden Methoden der `List` der passenden Namen.
- Fügen Sie eine Methode `add` hinzu, die einen neuen Namen hinzufügt.
- Fügen Sie eine Methode `setFilter` hinzu, der ein String übergeben wird. Erzeugen Sie die Liste der passenden Namen so, dass sie alle Namen enthält, die mit diesem String anfangen, falls dieser nicht leer ist, bzw. alle vorhandenen Namen bei leerem String. *Tipps:* Für einen `SortedSet` erhalten Sie durch Aufruf der Methode `subset(anfang, anfang+"\u0037")` alle Namen, die mit dem String „anfang“ beginnen. Da `ArrayList` einen Konstruktor besitzt, der eine `Collection` als Parameter akzeptiert, können Sie aus einem solchen `Subset` direkt in eine Liste überführen. Übergeben Sie dem Konstruktor von `ArraySet` den kompletten `SortedSet`, so erhalten Sie eine Liste aller Namen.
- Durch Aufruf von `fireContentsChanged(this, 0, Integer.MAX_VALUE)` für ein `AbstractListModel` wird die gesamte Liste in der grafischen Oberfläche aktualisiert.
- Melden Sie beim Textfeld für die Suche einen `KeyListener` an, dessen Methode `keyReleased` für Ihr `ListModel` die Methode `setFilter` mit dem Inhalt des Suchfelds als Parameter aufruft.
- Melden Sie bei der Schaltfläche „Übernehmen“ und dem Textfeld zu „Name:“ einen `ActionListener` an, der die Methode `add` Ihres `ListModel` aufruft.
- Melden Sie beim Textfeld zu „Name:“ einen `KeyListener` an, dessen Methode `keyReleased` die Schaltfläche „Übernehmen“ aktiviert, wenn das Textfeld nicht leer ist.
- Beachten Sie, dass es i. A. einfacher ist, von `KeyAdapter` zu erben statt `KeyListener` komplett zu implementieren.