



**RWTH AACHEN  
Lehrstuhl für Informatik 10 (Medieninformatik)  
Programmierung für Alle (Java)**

---

**Übung 10**

**Abgabe der Lösungszettel: 30. Januar 2007 vor der Übung**

**Deadline für die Abgabe der Programme per *E-Mail* beim Tutor:  
30. Januar 2007, 8:00 Uhr**

---

Tragen Sie hier Ihre **Gruppennummer** ein:

Tragen Sie hier den **Namen Ihres Tutors** ein:

Die Abgabe der Übung erfolgt in **Dreiergruppen**. Tragen Sie dazu alle Namen und Matrikelnummern Ihrer Gruppenmitglieder hier ein. Der / Die erste in der Tabelle gibt die Übung beim Tutor ab.

NAMEN	MATRIKELNUMMERN

Punktetabelle für die Korrektur:

	Punkte
1 (15)	
Total (15)	

## 1. GUIAdressbuch Teil 2 (Programmieraufgabe)

In dieser Aufgabe implementieren Sie die Funktionalität für das GUIAdressbuch. Den größten Teil des Codes können Sie aus der Musterlösung für die textbasierte Version des Adressbuches und für die GUI übernehmen. Zusätzlich soll ein in der Liste ausgewählter Kontakt gelöscht werden können:

- (a) [4 Punkte] In den Eingabefeldern soll ein Kontakt eingetragen und durch einen Klick auf “Kontakt hinzufügen” in das Adressbuch hinzugefügt werden. Ein Eintrag soll nur dann hinzugefügt werden, wenn keines der Felder leer UND der Eintrag noch nicht im Adressbuch vorhanden ist. Der Button “Felder löschen” löscht die Einträge in den Eingabefeldern.

*Hinweis:* Die Methode *kontaktHinzufuegen* (*String vname, String nname, String telnr*) in der Klasse *Adressbuch.java* soll überprüfen, ob ein Kontakt bereits in der Liste vorhanden ist. Benutzen Sie noch keine *Collection*-Klasse, die doppelte Kontakte automatisch entfernt.

- (b) [4 Punkte] Die Liste auf der linken Seite zeigt nicht mehr als 10 Kontakte auf einmal an. Es sollen entweder alle Kontakte oder nur die über das Suchfeld spezifizierten Kontakte angezeigt werden. Jedes im Suchfeld neu eingegebene Zeichen löst eine neue Suche und eine Aktualisierung der angezeigten Kontakte in der Liste aus.

*Tipp:* Ändern Sie in *Adressbuch.java* die Methode *kontakteSuchen* (*String suchstring*), so dass eine *ArrayList* mit den gefundenen Kontakten zurückgeliefert und diese an die *JList* weitergereicht wird. Benutzen Sie eine passende Methode aus dem *KeyListener-Interface*, um jeden Tastendruck im Suchfeld abzufangen.

- (c) [4 Punkte] Ein einzelner Kontakt soll in der Liste ausgewählt und durch einen Klick auf “Kontakt entfernen” aus dem Adressbuch entfernt werden. Das Entfernen eines Kontaktes muss auch dann funktionieren, wenn ein Text im Suchfeld eingegeben ist.

- (d) [3 Punkte] Beim Starten sollen die Kontakte aus der Datei *adressbuch.ser* deserialisiert und beim *Schließen des Fensters* automatisch serialisiert werden.

*Tipp:* Benutzen Sie eine passende Methode aus dem *WindowListener-Interface*, um das *Schließen des Fensters* abzufangen. Das Adressbuch braucht nicht gespeichert zu werden, wenn das Programm über das Menü oder einen Tastenkürzel beendet wird. Diese Funktionalität müssen Sie nicht implementieren.

**Tipps:** Jedes Mal wenn ein Kontakt hinzugefügt, gelöscht oder nach Kontakten gesucht wird, müssen die Einträge in der *JList* aktualisiert werden. Benutzen Sie dafür die *JList*-Methode *void setListData (Object[] listData)* und die *ArrayList*-Methode *Object[] toArray ()*.

Um die Kontakte in der *JList* im Format “Vorname Nachname:Telefonnummer” anzuzeigen, müssen Sie eine *Object*-Methode in der Klasse *Kontakt.java* überschreiben.

Die Datei *GUIAdressbuch.jar* enthält das lauffähige Programm. Sie können es durch *java -jar GUIAdressbuch.jar* (oder durch einen Doppelklick) starten und ausprobieren, falls ihnen die Funktionalität nicht klar ist.

## Hinweise zur Übungsabgabe

Bitte geben Sie zu jeder Übung das *Deckblatt*, angeheftet vor Ihren Lösungen, mit ab. *Programmieraufgaben* werden per E-Mail **und** der ausgedruckte Code vor Übungsbeginn beim Tutor abgegeben. *Verständnisaufgaben* werden **handschriftlich** beim Tutor abgegeben.

Sollten Sie Probleme oder Fragen haben, wenden Sie sich bitte an Ihren Tutor oder besuchen Sie die *Betreuungsstunden* im Lila Raum (4U15) im ZIP-Pool der Informatik: Donnerstag von 16:00 bis 18:00 und Freitag von 10:00 bis 12:00.