RWTH AACHEN
Lehrstuhl für Informatik 10 (Medieninformatik)

Programmierung für Alle (Java)

Bonus–Übung 11

Abgabe der Lösungszettel: 06. Februar 2007 vor der Übung

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

Tragen Sie hier Ihre **Gruppenummer** 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 (8)	
2 (6)	
Total (14)	

Sie haben es geschafft! Diese Übung 11 ist eine Bonus-Übung, die letzte Java-Übung, die sie lösen können bzw. lösen müssen. Sie müssen 50% der Übungspunkte aus den Übungen 7-10 erreicht haben, um die Zulassung zur Teilklausur 2 zu erhalten. Falls Sie noch nicht 50% der Übungspunkte erreicht haben, sollten Sie diese Übung unbedingt lösen und abgeben, um genügend Punkte zu erreichen. Falls Sie bereits 50% der Übungspunkte erreicht haben, sollten Sie diese Übung trotzdem lösen, weil der Inhalt relevant für die Klausur ist :-)

1. **GUIAdressbuch** Teil 3 (Programmieraufgabe)

   (a) [3 Punkte] Um doppelte Kontakte zu vermeiden haben wir bisher selber im Code überprüft, ob ein Kontakt bereits im Adressbuch enthalten war. Benutzen Sie jetzt eine geeignete Collection, um doppelte Kontakte zu entfernen. Zwei Kontakte gelten als dabei als *gleich*, wenn die Vornamen, die Nachnamen und die Telefonnummern in beiden Kontakten gleich sind.

   Tipp: Um den Inhalt zweier Kontakte als "gleich" zu erkennen, muss die Bedingung für *Objektgleichheit* erfüllt sein. Kontakte in Groß-/Kleinbuchstaben können Sie als unterschiedlich behandeln.

   (b) [3 Punkte] Die Kontakte im Adressbuch sind nicht sortiert. Ändern Sie das Programm, so dass die Kontakte in der Liste immer nach dem Vornamen geordnet angezeigt werden.

   Tipp: Um die Kontakte im Adressbuch zu sortieren, müssen die Kontakte *vergleichbar* sein.

   (c) [2 Punkte] Veröffentlichen Sie ihren Code, indem Sie eine *JAR-Datei* für GUI-Adressbuch erstellen. Die Datei *GUIAdressbuch.jar* soll das lauffähige Programm enthalten und durch *java -jar GUIAdressbuch.jar* gestartet werden können. Starten Sie das JAR-Programm und tragen Sie alle Namen und Matrikelnummern (als Telefonnummern) ihrer Gruppenmitglieder in das Adressbuch ein. Speichern Sie die Kontakte, so dass beim nächsten Start der JAR-Datei die Kontakte ihrer Gruppenmitglieder geladen werden. Senden Sie die JAR-Datei und die Kontaktliste an Ihren Tutor.

   Tipp: Sie können die Kontakte entweder weiterhin in einer ArrayList speichern ODER eine geeignete Collection benutzen, die keine doppelten Elemente erlaubt und gleichzeitig auch die Elemente sortiert.

2. **Rekursion**: Binärbäume (Programmieraufgabe)

   Ein Binärbaume ist ein Baum, bei dem jeder Knoten höchstens zwei Nachfolger besitzt: einen linken und einen rechten Unterbaum. Der linke Unterbaum enthält nur Elemente, deren Werte kleiner als der Wert des Knotens sind. Der rechte Unterbaum enthält nur Elemente, deren Werte größer oder gleich dem Wert des Knotens sind. Diese Datenstruktur eignet sich für ein schnelles Auffinden von Elementen.

   (a) [2 Punkte] Die Klasse *Baum.java* implementiert einen solchen Binärbaum. Schreiben Sie eine Klasse *BinaerbaumAufbauen.java*, die den Binärbaum aus der Datei *Baum.pdf* erzeugt und in die Datei *binaerbaum.ser* serialisiert.

   Tipp: Der Baum kann in einer einzigen (langen) Anweisung erzeugt werden.

   (b) [3 Punkte] Implementieren Sie für die Klasse *Baum.java* eine *statische rekursive* Methode *inorderAusgeben()*, die einen Baum als Argument erwartet und alle im Baum gespeicherten Werte "inorder" (in sortierter Reihenfolge) ausgibt.

   Tipp: Stellen Sie sicher, dass ihre rekursive Methode endet, wenn ein Unterbaum leer ist.

   (c) [1 Punkt] Schreiben Sie eine Klasse *BinaerbaumDurchlaufen.java*, die den Baum aus der Datei *binaerbaum.ser* deserialisiert und die Werte "inorder" ausgibt, d.h. die Ausgabe muss 1 2 3 4 5 sein.

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.