

## Bachelor-Studiengang

## Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2007/2008

## Übungsblatt 9: Interfaces, Klassen und Objekte

Ausgabe am: 4.12.2007

Abgabe am: 13.12.2007

Auf diesem Übungsblatt ist die Adressverwaltung von Übungsblatt 8 zu erweitern. Es werden die beiden bereits bekannten Schnittstellen *Contact* und *ContactManager* um weitere Methoden und Konstanten ergänzt und es gibt einen neuen Aufzählungstyp *SortOrder*.

Zu Aufgabe 1 und 2:

Zu jedem Kontakt sind eine Reihe von "Blankotexten" zu verwalten, in denen Informationen gespeichert werden können, für die keine Rubrik wie "Adresse" oder "Telefonnummer" vorgesehen sind.

Diese bestehen jeweils aus einem für alle Kontakte einheitlichen Namen und einem freien Text pro Kontakt. Zum Beispiel könnte zu jedem Kontakt gespeichert werden, wann und wo man diese Person kennengelernt hat. Dafür könnte man den Namen eines der Blankotexte "Erstkontakt" nennen und die Texte zu einigen Kontakten könnten "Geburtstagsparty von Udo 2005" oder "XP-Konferenz 2006" lauten. Andere Beispiele für Verwendungsmöglichkeiten von Blankotexten könnten "2. Email-Adresse" oder "Nummer der Sekretärin" sein.

Die Konstante *ContactManager.CUSTOM\_LABEL\_NUMBER* legt fest, wieviele solcher Zusatztexte pro Kontakt gespeichert werden können.

**Aufgabe 1****15 Punkte**

Erweitern Sie Ihre Implementierung des *Contact*-Interface' um die beiden neuen Methoden für die Blankotexte.

Vergessen Sie nicht, die zusätzlichen Informationen pro Kontakt auch abzuspeichern und zu laden, also die *load*- und *save*-Methoden zu erweitern!

**Aufgabe 2****20 Punkte**

Erweitern Sie Ihre Implementierung des *ContactManager*-Interface' um die beiden Methoden, die nötig sind, um die Namen der Blankotexte zu setzen und abzufragen.

Vergessen Sie nicht, diese zusätzlichen Informationen auch abzuspeichern und zu laden, also die *load*- und *save*-Methoden zu erweitern!

**Aufgabe 3****10 + 10 + 10 + 10 = 40 Punkte**

Implementieren Sie Suchmethoden *searchByNameStart*, *searchByZip* und *searchByText* so, wie sie im Interface beschrieben sind. Wählen Sie einen beliebigen Suchalgorithmus; auch eine lineare Suche ist zulässig.

Trifft das angegebene Kriterium auf mehrere Kontakte zu (d.h. es wird ein Array mit einer Länge größer eins als Ergebnis geliefert), so ist für die Reihenfolge im Array das mit *setSortOrder* gesetzte Sortierkriterium zu berücksichtigen.

**Beispiel:**

Es gebe folgende Kontakte

- Guido Schmidt
- Alfred Schmitt
- Hans Schmidt
- Jens Müller

Die Suche *searchByNameStart("Schmi")* mit dem Sortierkriterium *SortOrder.LAST\_NAME\_THEN\_FIRST\_NAME* liefert

Guido Schmidt, Hans Schmidt, Alfred Schmitt.

Die Suche *searchByNameStart("Schmi")* mit dem Sortierkriterium *SortOrder.FIRST\_NAME\_THEN\_LAST\_NAME* liefert

Alfred Schmitt, Guido Schmidt, Hans Schmidt.

Achtung:

Die Regelung mit dem Sortierkriterium bezieht sich auch auf die Suchergebnisse der beiden anderen Suchmethoden.

**Aufgabe 4****20 Punkte**

Erweitern Sie Ihre Menü-Umgebung für die Adressverwaltung so, dass die neue Funktionalität aus den Aufgaben 1 bis 3 genutzt werden kann.