



Bachelor-Studiengang Informatik
Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2009/2010

Übungsblatt 8: Interfaces, Klassen und Objekte Ausgabe am: 2.12.2009
Abgabe am: 16.12.2009

Auf diesem und den nächsten Übungsblättern sollen Sie eine Kontaktverwaltung (englisch: *Contact Manager*) erstellen, die inkrementell ausgebaut wird, indem schrittweise immer mehr Programmfunktionalität ergänzt wird. Mehrere Interfaces definieren einheitliche Schnittstellen zu Ihren Programmen.

Aufgabe 1 **8 Punkte**

Auf der Vorlesungshomepage finden Sie ein Interface *Address*, das definiert, welche Operationen konkrete Adressen (mit Straße, Hausnummer, Postleitzahl und Stadt) zur Verfügung stellen. Dazu gehören auch Operationen zum Speichern der *Address*-Daten in eine Datei und zum Laden der geschriebenen Daten aus einer Datei.

Implementieren Sie dieses Interface in einer Klasse mit einem Konstruktor, der alle angegebenen Daten als Parameter akzeptiert.

Aufgabe 2 **6 Punkte**

Auf der Vorlesungshomepage finden Sie ein Interface *Date*, das definiert, welche Operationen konkrete Datumsangaben (mit Tag, Monat und Jahr) zur Verfügung stellen. Dazu gehören auch Operationen zum Speichern der *Date*-Daten in eine Datei und zum Laden der geschriebenen Daten aus einer Datei.

Implementieren Sie dieses Interface in einer Klasse mit einem Konstruktor, der alle angegebenen Daten als Parameter akzeptiert.

Aufgabe 3 **16 Punkte**

Auf der Vorlesungshomepage finden Sie ein Interface *Contact*, das definiert, welche Operationen konkrete Datumsangaben zur Verfügung stellen. Zu einem Kontakt gehören:

- Vor- und Nachname, Titel (z.B. Dr., Prof. Dr.)
- Name der Firma, bei der die Person arbeitet
- Privat- und Firmenadresse (beide vom Typ *Address*, siehe Aufgabe 1)
- Telefonnummern: privat, dienstlich und mobil
- Email-Adresse und URL
- Geburtstag (beide vom Typ *Date*, siehe Aufgabe 2) und die Angabe, ob beim Start des Kontakt-Managers (siehe Aufgabe 4) in der Woche vor dem Geburtstag an diesen erinnert werden soll. Die Angabe des Geburtstags ist optional, weil das Geburtsdatum nicht von allen Kontakten bekannt ist.

Außerdem stellt jeder Kontakt Operationen zum Speichern der *Contact*-Daten in eine Datei und zum Laden der geschriebenen Daten aus einer Datei zur Verfügung (dabei werden für die *Address*- und das *Date*-Objekt deren Methoden verwendet).

Implementieren Sie dieses Interface in einer Klasse mit einem Konstruktor, der alle angegebenen Daten als Parameter akzeptiert.

Aufgabe 4 **15 Punkte**

Auf der Vorlesungshomepage finden Sie ein Interface *ContactManagement*, das die Operationen der Adressverwaltung definiert. Folgende Funktionalität soll die Adressverwaltung in dieser ersten Version bieten:

- einen Kontakt hinzufügen, einen Kontakt durch einen anderen ersetzen und einen Kontakt löschen,
- alle aktuell im Speicher vorliegenden Kontakte in eine Datei speichern bzw. die in einer Datei vorliegenden Kontakte in die Adressverwaltung im Speicher laden,
- alle aktuell vorliegenden Kontakte in einem (dafür erstellten) passend dimensionierten(!) Array liefern, so dass sie zum Beispiel auf dem Bildschirm ausgegeben werden können.

Implementieren Sie dieses Interface in einer Klasse.

Aufgabe 5 **25 Punkte**

Erstellen Sie eine Umgebung, mit der Sie Ihre Adressverwaltung aus Aufgabe 4 bedienen können, also z.B. neue Kontakte eingeben und anlegen, vorhandene Kontakte anzeigen, Kontaktdaten in eine Datei speichern oder von Datei laden. Ihre Umgebung darf nur Methoden aus den vier angegebenen Schnittstellen benutzen; diese dürfen nicht verändert oder erweitert werden. (In Ihren Klassen aus Aufgabe 1 bis 4 dürfen Sie selbstverständlich auch andere Methoden definieren und benutzen!)

Beispiel für einen typischen Arbeitsablauf mit dem Programm:

- Start der Umgebung
- Eingabe des Namens der Datei, von der Kontakte geladen werden sollen, laden der Kontakte
- Ausgabe aller geladenen Kontaktdaten
- Löschen eines der Kontakte
- Eingabe eines neuen Kontakts
- Speichern aller Kontaktdaten
- Ausgabe aller Kontaktdaten

Hinweise

- Sie dürfen keines der Interfaces verändern!
- Ihr Programm sollte aus fünf Klassen bestehen. Achten Sie auf eine gute Aufteilung der Aufgaben (wer ist für was zuständig?), damit Ihr Programm gut erweiterbar ist.
- Vergessen Sie nicht, Ihr Programm mit geeigneten Kommentaren zu versehen.
- Abgabe und Testate für diese Aufgaben finden erst in 14 Tagen statt, es wird aber nächste Woche ein neues Übungsblatt mit weiteren Aufgaben dazukommen. Sie dürfen uns gerne schon nächste Woche Ihre Lösungen zeigen!