

Bachelor-Studiengang "Informatik"

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

Übungsblatt 10: Datumsrechnung

Ausgabe am: 8.1.2009

Abgabe am: Dienstag, 20.1.2009

Aufgabe 1**3 + 3 + 8 + 4 + 8 + 8 + 3 + 3 = 40 Punkte**

Auf der Vorlesungshomepage finden Sie ein Java-Interface *Date* und einen Aufzählungstyp *Weekday*, der in diesem Interface verwendet wird. Jedes Objekt einer *Date*-Implementierung repräsentiert ein bestimmtes Datum im Gregorianischen Kalender. Erstellen Sie eine Klasse *DateImpl* in Java, welche alle Methoden des Interface' wie nachfolgend beschrieben implementiert und einen Konstruktor wie unten angegeben zur Verfügung stellt.

Die Operationen von *Date* sind wie folgt definiert:

- *Date*: $\text{int} \times \text{int} \times \text{int} \rightarrow \text{Date}$
Dieser Konstruktor akzeptiert in der Reihenfolge Tag, Monat und Jahr als Parameter und liefert ein entsprechendes *Date*-Objekt als Ergebnis
- *getDay*: $\text{Date} \rightarrow \text{int}$
getMonth: $\text{Date} \rightarrow \text{int}$
getYear: $\text{Date} \rightarrow \text{int}$
Wenn es sich um ein gültiges Datum handelt, liefern diese Methoden den Tag, den Monat bzw. das Jahr des angegebenen Datums als *int*-Wert; im Falle eines ungültigen Datums liefern alle drei Methoden -1
- *isValid*: $\text{Date} \rightarrow \text{boolean}$
Liefert genau dann *true*, wenn es sich um ein gültiges Datum handelt, sonst *false*
- *isLaterThan*: $\text{Date} \times \text{Date} \rightarrow \text{boolean}$
Liefert genau dann *true*, wenn das 1. Datum später als das 2. Datum ist, sonst *false*
- *getWeekday*: $\text{Date} \rightarrow \text{Weekday}$
Liefert den Wochentag zu einem gültigen Datum bzw. *null* zu einem ungültigen Datum
- *subtract*: $\text{Date} \times \text{Date} \rightarrow \text{int}$
Subtrahiert das 2. Datum vom 1. Datum; liefert die Differenz der beiden Datumsangaben in Tagen als ganze Zahl
Voraussetzung: das 1. Datum ist größer als das 2. Datum, sonst ist das Ergebnis -1
- *saveToFile*: $\text{Object} \rightarrow \text{void}$
Speichert die Datumsangabe in die übergebene Datei
- *loadFromFile*: $\text{Object} \rightarrow \text{void}$
Lädt Datumsangaben aus der übergebenen Datei in das Datumsobjekt

Fügen Sie Interface, Aufzählungstyp und Klasse zum Package *contactManager* hinzu.

In Ihrer Klasse *DateImpl* dürfen Sie je nach Bedarf beliebige Felder und Hilfsmethoden sowie zusätzliche eigene Klassen verwenden, aber keinerlei Hilfsklassen oder -methoden aus der Java-Bibliothek.

Aufgabe 2

4 + 3 + 4 + 4 + 3 = 18 Punkte

Erweitern Sie das *Contact-Interface* und Ihre Implementierung davon um folgende neue Methoden:

- Mittels *setBirthDay* (Parameter: Tag, Monat und Jahr, in dieser Reihenfolge), *getDayOf-Birth*, *getMonthOfBirth* und *getYearOfBirth* kann für jeden Kontakt ein Geburtsdatum eingestellt bzw. abgefragt werden. Wurde kein (gültiges) Datum eingestellt, liefert die *get-Methoden* -1.
- Mittels *setReminderOneWeekBeforeBirthday* (Boole'scher Parameter: Erinnerung aktivieren oder nicht) kann eingestellt werden ob die Kontaktverwaltung für eine Kontaktperson an einen bevorstehenden Geburtstag erinnern soll oder nicht. Die Erinnerung kann nur eingeschaltet werden, wenn *vorher oder nachher* ein (gültiges) Geburtsdatum eingestellt wurde. Mit *isReminderOneWeekBeforeBirthdayActivated* kann abgefragt werden, ob ein Datum eingestellt und die Erinnerung aktiviert wurde.
- Mittels *isOneWeekBeforeBirthday* (Parameter: heutiger Tag, Monat und Jahr, in dieser Reihenfolge) kann abgefragt werden, ob der Geburtstag einer Kontaktperson innerhalb der sieben Tage liegt, die am angegebenen Datum beginnt (exklusive dem Geburtsdatum selbst). Wurde kein (gültiges) Datum eingestellt oder die Erinnerung nicht aktiviert, wird *false* geliefert.

Erweitern Sie die Bedienumgebung der Kontaktverwaltung wie folgt:

- Man kann zu jedem Kontakt den Geburtstag eingeben und abfragen sowie einstellen, ob an diesen Geburtstag erinnert werden soll. Diese Informationen werden bei der Ausgabe mit ausgegeben.
- Die Bedienumgebung (Achtung: *nicht* der *Kontaktmanager*!) prüft beim Start für alle Kontakte, ob ein Geburtstag eine Woche oder kürzer bevorsteht und ob an diesen erinnert werden soll und nennt diejenigen Kontakte, für die beides zutrifft.

Vergessen Sie nicht, den Geburtstag und die Information, ob daran erinnert werden soll, mit den anderen *Contact*-Daten zu speichern und zu laden.

Benutzen Sie für die Implementierung der Datumsoperationen Ihre Implementierung von Aufgabe 1. Die Interfaces *Address* und *ContactManager* dürfen nicht verändert werden, ihre Implementierung braucht nicht verändert zu werden!