

## Bachelor-Studiengang

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

## Freiwilliges Zusatzblatt: Interfaces, ADT

Ausgabe am: 18.12.2007

Abgabe am: Dienstag, 8.1.2008

Betrachtet wird ein abstrakter Datentyp *Date*:

Der ADT *Date* repräsentiert ein bestimmtes Datum im Gregorianischen Kalender und bietet verschiedene Operationen auf Kalenderdaten an.

Die Operationen 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 die angegebenen Zahlen in der Reihenfolge Tag, Monat, Jahr ein gültiges Datum repräsentieren, 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

Die Operationen *isLaterThan*, *getWeekday* und *subtract* (nicht jedoch der Konstruktor und die anderen Operationen) dürfen sich darauf verlassen, dass sie nur mit gültigen Datumsangaben aufgerufen werden.

**Aufgabe 1****5 Punkte**

Informieren Sie sich über den Gregorianischen Kalender. Beschreiben Sie kurz, wie dieser definiert ist.

**Aufgabe 2****5 Punkte**

Erstellen Sie einen Aufzählungstyp *Weekday* mit den Elementen *MONDAY*, *TUESDAY*, ..., *SUNDAY*.

**Aufgabe 3****12 + 44 + 7 + 14 + 12 + 6 = 95 Punkte**

Auf der Vorlesungshomepage finden Sie eine Definition für ein Java-Interface *Date*. Erstellen Sie eine Klasse *DateImpl* in Java, welche alle Methoden dieses Interface' wie oben beschrieben implementiert und einen Konstruktor wie oben angegeben zur Verfügung stellt.

Sie dürfen je nach Bedarf beliebige Attribute und Hilfsmethoden sowie zusätzliche eigene Klassen verwenden, aber keinerlei Hilfsklassen oder -methoden aus der Java-Bibliothek.

Erstellen Sie das Interface *Date*, Ihre Klasse *DateImpl* und den Aufzählungstyp *Weekday* in einem Paket namens *date*.

**Aufgabe 4****50 Punkte**

Erstellen Sie ein Programm, das seinen eigenen Quelltext auf der Konsole ausgibt.

Sie dürfen keinerlei Methoden benutzen, die auf Dateien zugreifen und keinerlei Methoden anderer Bibliotheken als *MakeItSimple* und *String*.

Eine gruppenübergreifende Zusammenarbeit ist für diese Aufgabe nicht erlaubt und alle Gruppenmitglieder müssen das Programm wirklich gut erklären können!

**Hinweis**

- Es wird ein automatisches Testprogramm für Aufgabe 3 geben.