

Bachelor-Studiengang

Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2006/2007

Übungsblatt 10: Interfaces, ADT

Ausgabe am: 8.1.2007

Abgabe am: 17.1.2007

Betrachtet wird ein abstrakter Datentyp *Date*:

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

Die jeweiligen Operationen sind wie folgt definiert:

- *Date*: $\text{int} \times \text{int} \times \text{int} \rightarrow \text{Date}$
Dieser Konstruktor 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*
- *equals*: $\text{Date} \times \text{Date} \rightarrow \text{boolean}$
Liefert genau dann *true*, wenn die beiden Kalenderdaten gleich sind, sonst *false*
- *laterThan*: $\text{Date} \times \text{Date} \rightarrow \text{boolean}$
Liefert genau dann *true*, wenn das 1. Datum später als das 2. Datum ist, sonst *false*
- *earlierThan*: $\text{Date} \times \text{Date} \rightarrow \text{boolean}$
Liefert genau dann *true*, wenn das 1. Datum früher als das 2. Datum ist, sonst *false*
- *getWeekday*: $\text{Date} \rightarrow \text{Weekday}$
Liefert den Wochentag zu einem 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

Die Operationen *equals*, *laterThan*, *earlierThan*, *getWeekday* und *subtract* (nicht jedoch der Konstruktor, die *get*-Methoden und *isValid*) dürfen sich darauf verlassen, dass sie nur mit gültigen Datumsangaben aufgerufen werden.

Aufgabe 1**10 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 + 7 + 7 + 14 + 12 = 103 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.

Testen Sie Ihre Klasse gründlich, zum Beispiel mit Hilfe einer zusätzlichen Testklasse, von der aus Ihre Methoden aufgerufen werden.

Hinweise

- Es wird ein automatisches Testprogramm für Aufgabe 1 geben; dafür müssen auf jeden Fall der geforderte Konstruktor und das Interface *Date* implementiert werden.
- Überlegen Sie sich vor der Bearbeitung dieser Übungsaufgaben, wie viele Punkte Sie noch benötigen, um die 75%-Klausurzulassungsquote zu erreichen und wie viele (Teil-)Aufgaben Sie bearbeiten sollten, um dieses Ziel **sicher** zu erreichen!
Es ist vermutlich nicht notwendig, alle (Teil-)Aufgaben zu bearbeiten; für das Abschneiden in der Klausur ist nicht relevant, ob Sie zum Beispiel 75% oder 98% der Übungspunkte erreicht haben.