

Übung 02: Verzweigungen

Abgabetermin: 28.10.2008

Name: _____ Matrikelnummer: _____

Gruppe: G1 (Prähofer) G2 (Wolfinger) G3 (Wolfinger)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 2.1	12	<input type="checkbox"/>	Prosabeschreibung Ablaufdiagramm Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	
Aufgabe 2.2	12	<input type="checkbox"/>	Prosabeschreibung Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	

Aufgabe 2.1: Datumsprüfung

In dieser Aufgabe wird ein Programm geschrieben, das ein Datum in der Form von drei ganzen Zahlen für Tag, Monat und Jahr einliest und überprüft, ob es sich um ein gültiges Datum handelt.

Lesen Sie ein Datum in der Form von drei Zahlen für Tag, Monat und Jahr ein und prüfen Sie, ob

- Das Datum innerhalb der gregorianischen Zeitrechnung, d.h. seit 15.10.1582 und dem heutigen Tag, liegt.
- Es sich um ein gültiges Monat handelt (1 bis 12).
- Der Tag gültig ist, d.h., manche Tage haben 31, manche Tage nur 30, der Februar 28 oder 29. Berücksichtigen Sie auch Schaltjahre. Ein Jahr ist ein Schaltjahr, wenn es durch 4 teilbar ist. Jahre die durch 100, aber nicht durch 400 teilbar sind, sind keine Schaltjahre.

Das Programm soll ausgeben, ob es sich um ein gültiges Datum handelt. Ist das Datum nicht gültig, soll der Grund ausgegeben werden.

Abzugeben sind:

- Die Beschreibung des Programms in Prosa
- Das Ablaufdiagramm
- Das Java-Programm
- Testfälle und die Ergebnisse

Aufgabe 2.2: Einlesen und formatierte Ausgabe von Telefonnummern

Telefonnummern können in unterschiedlichem Format nach folgender Grammatik eingegeben werden

```
PhoneNumber = [ "+" [ "+" ] countrycode ]
              "( " "0" ( " )" areacode "/" | areacode " )" )
              number [ ( "-" | "*" ) extension ] "."
```

Das heißt, eine Telefonnummer kann mit einem "+" und optional einem weiteren "+" und der Länderkennzahl (*countrycode*) beginnen. Dann folgt entweder "(0)" und die Vorwahl (*areacode*) gefolgt von einem "/" oder die Vorwahl zur Gänze in Klammer "(0areacode)". Dann folgt die Telefonnummer. Optional kann noch eine Klappe (*extension*) folgen. Dann steht davor ein "-" oder ein "*". Abgeschlossen wird die Telefonnummer immer mit einem Punkt "."

Länderkennzahl (*countrycode*), Vorwahl (*areacode*), Nummer (*number*) und Klappe (*extension*) sind positive Integerzahlen. Als Einschränkung gilt zusätzlich, dass der Ländercode nicht größer als 999 sein darf.

Gültige Eingaben sind z.B.:

```
+43(0)732/2468-7132.
++43(0732)2468-7132.
++43(0732)2468*7132.
(0732)2468-7132.
(0)732/2468*7132.
++43(0732)2468.
(0)732/2468.
```

Schreiben Sie ein Programm, das eine Eingabe von der Konsole liest und die Korrektheit der Eingabe nach der oben angeführten Grammatik prüft. Geben Sie sprechende Fehlermeldungen aus, wenn die Eingabe als nicht korrekt erkannt wurde und brechen Sie das Programm ab (`return;`).

Schließlich soll die Telefonnummer im folgenden Standardformat wieder ausgegeben werden:

```
++countrycode(0)areacode/number[-extension]
```

wobei die *extension* nur ausgegeben wird, wenn sie nicht 0 ist. Weiters wird die Länderkennzahl 43 angenommen, wenn diese bei der Eingabe nicht angegeben wurde.

Zum Beispiel wird die Eingabe

```
(0732)2468*7132.
```

folgend ausgegeben

```
++43(0)732/2468-7132
```

Verwenden Sie dazu folgende Methoden von `In`:

- `readInt()` liest die nächste ganze Zahl
- `peek()` liefert das nächste Zeichen, ungleich dem Leerzeichen, ohne es zu lesen. (Achtung: Leerzeichen werden überlesen!)
- `done()` überprüft, ob die letzte read-Anweisung erfolgreich war
- `read()` liest das nächste Zeichen (Achtung: liefert auch Leerzeichen!).
- `readChar()` liest das nächste Zeichen, ungleich dem Leerzeichen (Achtung: Leerzeichen werden überlesen!)

Abzugeben sind:

- Die Beschreibung des Programms in Prosa
- Das Java-Programm
- Testfälle und die Ergebnisse

Achtung: Kein Ablaufdiagramm verlangt.