Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Peer Kröger
Janina Bleicher, Florian Richter

München, 03.11.2016

Einführung in die Programmierung

WS 2016/17

Übungsblatt 3: Vollständige Induktion, p-adische Zahlendarstellung

Besprechung: 14./16./18.11.2016

Ende der Abgabefrist: Freitag, 11.11.2016 14:00 Uhr.

Aufgabe 3-1 *Vollständige Induktion mit Ungleichungen* **2+2+2 Punkte**

(a) Zeigen Sie: $n^2 > 2n + 1$ für alle $3 \leq n \in \mathbb{N}$.

(b) Zeigen Sie: $2^n > n^2$ für alle $5 \leq n \in \mathbb{N}$ (Sie dürfen 3-1-a hierfür als bewiesen annehmen).

(c) Zeigen Sie, dass für alle $x \in \mathbb{R}, x \geq -1$ und für alle $n \in \mathbb{N}$ gilt: $(1 + x)^n \geq 1 + nx$

Hinweis: Da sie hier keine Gleichungen zeigen müssen, ist es manchmal notwendig, die Terme geschickt zu erweitern oder zu verringern.

Aufgabe 3-2 *Fehlerhafte Induktion* **2 Punkte**

Einer Ihrer Mitstudenten hat per Email Folgendes behauptet: *Wenn Einer die EIP-Klausur besteht, dann auch alle anderen!*
Auf Nachfragen folgte die Erklärung des Studenten als Induktionsbeweis:
(Induktionsanfang) *Angenommen, nur ein Student schreibt die Klausur mit und besteht. Dann haben offensichtlich alle Mitschreibenden bestanden.*
(Induktionsvoraussetzung) *Nun nehmen wir mal an, dass für $n \in \mathbb{N}$ Studenten S_n, die Behauptung gilt: Wenn ein Student $\tilde{s} \in S_n$ besteht, dann bestehen auch alle anderen $s \in S_n \setminus \{\tilde{s}\}$.*
(Induktionsschritt) *Schließlich meldet sich noch ein weiterer Student an. Die Menge der Studenten S_{n+1} hat nun eine Größe von $n + 1$. Wir nehmen an, dass ein Student $\tilde{s} \in S_{n+1}$ die Klausur besteht. Wir können S_{n+1} in zwei sich überschneidende Mengen von Studenten aufteilen:*

$$S_{n+1} = \{s_1, \ldots, s_{n+1}\} = \{s_1, s_2, \ldots, s_n\} \cup \{s_2, s_3, \ldots, s_{n+1}\} =: S' \cup S''$$

Der Student $\tilde{s}$, der nach Annahme besteht, ist in mindestens einer der Mengen S' oder S''. Die erste Menge $S' = s_1, \ldots, s_n$ sind die n zuerst angemeldeten Studenten. Vermutlich ist $\tilde{s} \in S'$, weil er immer brav in die Vorlesung ging und alle Informationen bezüglich Klausuranmeldungen mitbekommen hat. Nach Anwendung der Induktionsvoraussetzung bestehen dann alle $s \in S'$. Die zweite Menge sind die zuletzt angemeldeten Studenten $S'' = \{s_2, \ldots, s_{n+1}\}$. Das sind ebenfalls n viele, von denen die $n - 1$ Studenten $\{s_2, \ldots, s_n\} = S' \cap S''$ ja schon bestanden haben. Also kann auch hier die Induktionsvoraussetzung angewendet werden (falls $\tilde{s}$ sich zuletzt angemeldet hat, dann hat auf jeden Fall S'' bestanden. Dann aber auch S'). Folglich bestehen alle $n + 1$ Studenten.
Damit folgt, dass wir alle bestehen, weil einer schafft es doch immer.

Begründen Sie, warum dieser Student die vollständige Induktion noch einmal nachschlagen sollte.

Aufgabe 3-3 *p-adische Zahlendarstellungen* **4 Punkte**

Die Datei `p-adisch.txt` enthält folgende Tabelle:

$p = 2$	$p = 8$	$p = 10$	$p = 16$
1101	15	13	D
	76		
		26	
1111001			
			7C

Ergänzen Sie die Tabelle so, dass in jeder Zeile die verschiedenen p-adischen Zahlendarstellungen für die selbe Zahl stehen.

Aufgabe 3-4 *Karten umdrehen* **2+1+1 Punkte**

Gegeben ist das folgende Ein-Spieler-Spiel. Vor Ihnen liegen $n \in \mathbb{N}$ Spielkarten verdeckt in einer Reihe nebeneinander auf dem Tisch. Ein Zug besteht immer aus zwei Schritten:

1. Eine beliebige verdeckte Karte wird ausgewählt und herumgedreht, sodass ihre Bildseite zu sehen ist.

2. Die Karte, die direkt rechts neben der gewählten Karte liegt, wird ebenfalls herumgedreht - ungeachtet dessen, ob sie noch verdeckt ist oder nicht. Gibt es dort keine Karte, dann überspringe diesen Schritt.

(a) Terminiert das Spiel nach endlich vielen Zügen? Begründen Sie schlüssig Ihre Antwort.

(b) Angenommen, in jedem Zug muss die Karte links von der gewählten Karte statt der rechten Nachbarkarte umgedreht werden. Ändert dies etwas? Begründen Sie.

(c) Nun erlauben wir, dass sich der Spieler in jedem Zug aussuchen darf, ob er im 2. Schritt die linke oder die rechte Karte umdreht. Was können Sie nun über das Terminierungsverhalten sagen?

Hinweis: Sie können die binäre Zahlendarstellung als Repräsentation der Kartenreihe nutzen .

Aufgabe 3-5 *Datum-zu-Wochentag-Konverter* **2+1+2+0 Punkte**

In dieser Aufgabe sollen Sie ein Programm erstellen, dass ein Datum vom Benutzer einliest und den entsprechenden Wochentag (Montag bis Sonntag) ausgibt.

(a) Erstellen Sie wie zuvor schon eine .java-Datei `Wochentag.java`, die eine public Klasse `Wochentag` enthält. Diese soll wie gewohnt eine Methode `public static void main()` enthalten. Bitten Sie in dieser Methode den Nutzer um die Eingabe des Datums. Zur einfacheren Handhabung können Sie dies in drei Einzelabfragen machen: Tag, Monat und Jahr. Die drei Nutzereingaben sollen Sie in Variablen des Typs `int` speichern. Eine mögliche Konsolenausgabe könnte wie folgt aussehen:

```
Fuer die Berechnung des Wochentags brauche ich das Datum des Tages.
Gebe das Jahr an: 2045
Gebe den Monat an: 9
Gebe den Tag des Monats an: 14
```

(b) Wir nehmen nun an, dass diese Variablen `y,m,d`(year, month, day) heißen. Gemäß dem gregorianischen Kalender gibt es natürlich Schaltjahre, die wir nun berücksichtigen müssen. Nutzen Sie die folgenden Berechnungsformeln, um den Wochentag d_0 in einer ganzzahligen Repräsentation (int) zu erhalten:

```
y0 = y-(14-m)/12
x = y0-y0/4-y0/100+y0/400
m0 = m+12*((14-m)/12)-2
d0 = (d+x+(31*m0)/12)%7
```

(c) Die Wochentage werden nun durch Zahlen repräsentiert, d.h. 0=Sonntag, 1=Montag, ... Nutzen Sie if-Abfragen, um in der Konsole einen passenden für Menschen interpretierbaren Text auszugeben. Das vollständige Programm könnte dann wie folgt aussehen:

```
Fuer die Berechnung des Wochentags brauche ich das Datum des Tages.
Gebe das Jahr an: 2045
Gebe den Monat an: 9
Gebe den Tag des Monats an: 14
Der 14.9.2045 ist ein Donnerstag!
```

(d) * Als kleiner Zusatz: Bisher sind auch falsche Eingaben erlaubt, z.B. `42.17.0`. Fangen Sie diese Eingaben ab und beenden Sie das Programm mit einer entsprechenden Meldung.