

Praktische Informatik 3

Tutorien Dienstag, 8–10 Uhr und 10–12 Uhr, Dr. Diedrich Wolter

3.11.2011

Zur Person...

- Mitglied der Arbeitsgruppe „**Cognitive Systems**“
- Ich forsche an Themen der **Künstlichen Intelligenz (KI)**...
 - Wissensrepräsentation
 - Schlussfolgern, insbesondere mit räumlichen und zeitlichem Wissen
 - Navigation autonomer mobiler Roboter
- Funktionale Programmierung verwende ich...
 - bei der Entwicklung unseres **Reasoners SparQ**
 - bei algorithmisch schwierigen Problemen
 - oft und gerne wegen der interaktiven Arbeitsweise

Kontakt

- Am allerbesten und rund um die Uhr per Mail:

dwolter@informatik.uni-bremen.de

- Persönlich—bitte macht (per Mail) einen Termin ab!

Büro: Cartesium, 03.13

Telefon: (218)-64181

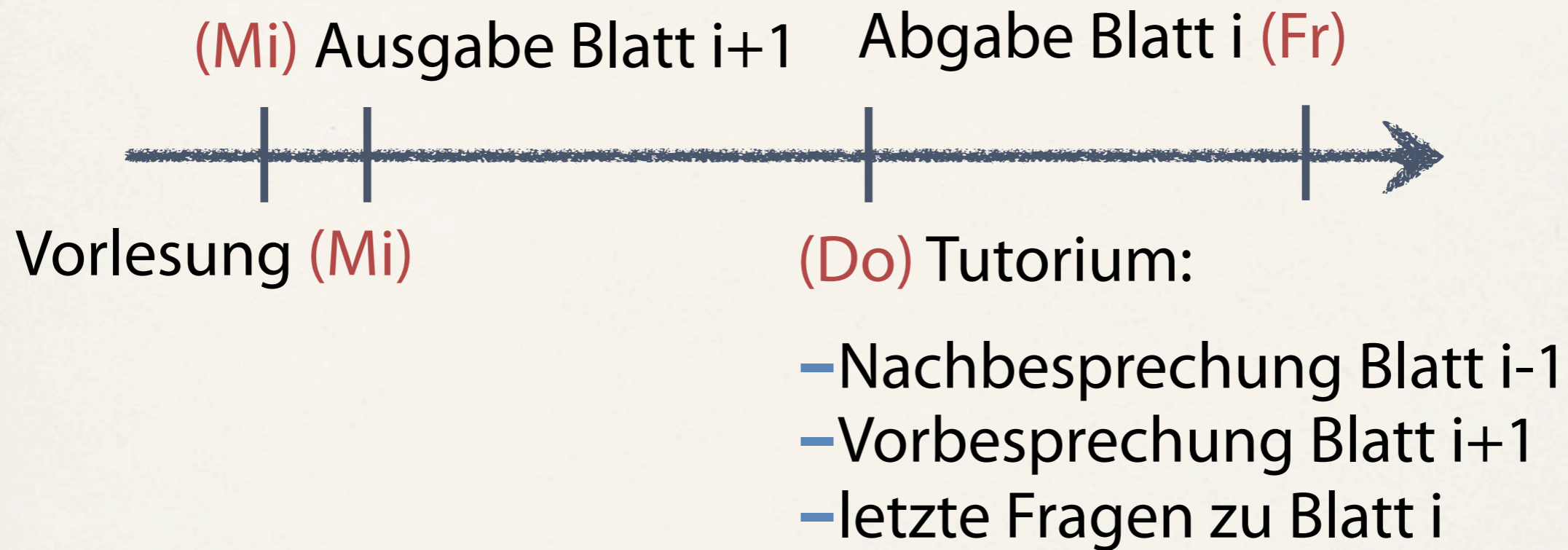
Organisatorisches

- Tutorien:

Zeit	Raum	Teilnehmer
8:30–10:00	MZH 1110	?
10:15–11:45	MZH 1110	?

- Sucht im eigenen Interesse kleine Tutorien auf

Zeitplan



- arbeitet die Vorlesung nach
 - etwas nicht verstanden? Problem selbst zu lösen?
- bereitet die Tutorien vor
 - mit Übungsaufgaben beschäftigen

Scheinkriterien

- Es werden 11 Übungsblätter ausgegeben
- jedes Übungsblatt muss **erfolgreich bearbeitet** werden ($\geq 20\%$ der Punkte)
- Insgesamt müssen $\geq 50\%$ der Punkte erreicht werden
- Note gemäß erzielter Punkte (siehe Netz)
- Bearbeitung in Kleingruppen mit **max. 3 Teilnehmern**
- **Fachgespräch** am Ende der Vorlesungszeit

Übungen

- Übungsblätter werden Mittwoch Mittag veröffentlicht:

http://www.informatik.uni-bremen.de/agbkb/Lehre/pi3/index_d.htm

(auch über StudIP)

- Abgabe per Mail in der übernächsten Woche bis **Freitag, 23:59 Uhr**
- zu spät abgegebene Blätter werden als **0 Punkte** gewertet

Abgabe

- Mail an: `dwo1ter@informatik.uni-bremen.de`
Betreff: `PI3 dddd Xyz`
bitte einsetzen:
`dddd` Tutoriumszeit: 8–10 oder 10–12
`Xyz` Gruppenkürzel
Gruppe **Donald**, **Dagobert**, **Mickey** -> **DDM**
- Abgabe als **ein komprimiertes Verzeichnis** (.tar.gz oder .zip),
benannt nach dem Schema **Blatt01.zip**
 - Ausarbeitung als **PDF-Dokument** (Latex-Klasse im Netz)
 - Quellcode aller Dateien (**.hs** oder **.lhs**)

Bewertung

- Punktverteilung nach folgendem Schlüssel:
 - 40% Quellcode
 - 40% Dokumentation
 - 20% Tests

- Neben dem Inhalt zählt auch die Form
 - Achtet auf Ausdruck, Rechtschreibung und Typographie!

Quellcode

- pro Modul eine Datei (Dateiname = Modulname)
(bitte fügt wiederverwendete Module bei!)
 - Kommentare zur Strukturierung und zur Erklärung
- Implementation, nach Aufgaben getrennt
- Bewertung:
 - Korrektheit
 - **Kommentare**
 - Organisation, Form, Lesbarkeit

Quellcode

- erklärende Kommentare

```
isqrt :: Integer -> Integer
-- isqrt x: „integer square root“ von x           Welche Argumente?
-- berechnet die ganzzahlige wurzel                Was?
-- mit Hilfe des Newtonverfahrens                  Wie? (ggf.)
-- nicht definiert für negative werte!             Wichtige Bedingungen?
```

- überlegt Euch sprechende Bezeichner

- help, help2, tmp,... sind meist unakzeptabel!
- „zaehler“ ist auch nicht viel besser
- x,y sind für Zahlenwerte (z.B. Koordinaten) wunderbar

Ausarbeitung und Dokumentation

- Wir unterscheiden zwischen **Dokumentation** und **Kommentaren**
 - **Kommentare** beschreiben elementaren Bausteine wie eigene Typen, Funktionen, etc.
 - **Dokumentation** erklärt Euren Lösungsansatz
 - Wie gehen wir das Problem an?
 - Welche Bausteine (Funktionen, Algorithmen) hat unsere Lösung?
 - Quellcode oft fehl am Platz
 - auch: im Falle des Scheiterns Probleme besprechen

Tests

- Das Testen dient dem Überprüfen der Funktionsfähigkeit
 - Wenn der Test vorhandene Fehler nicht aufspürt, dann ist der Test fehlerhaft
- Auswahl geeigneter Testfälle (Warum geeignet?)
- Ein Test beinhaltet
 - erwartetes Ergebnis (Wieso erwartet?)
 - erhaltenes Ergebnis
 - Bewertung (OK/Fehler)
- Spezielle Testfunktionen können helfen

Tests

- Ein Beispiel zur Funktion `isqrt`

Nr.	bestanden?	Argument	Erwartung	Ergebnis
1.	ja	2	$\sqrt{2} = 1.41\dots \rightsquigarrow 1$	1
2.	ja	0	$\sqrt{0} = 0$	0
3.	ja	100	$\sqrt{100} = 10$	10
4.	NEIN	-2	$\sqrt{-2}$ nicht definiert! \rightsquigarrow Fehler	-1

- Testen in Haskell ist leicht—nutzt die Interaktivität
- Tests helfen Euch, Fehler zu finden und zu beheben
 - Schreibt zu Beginn eigene Testfunktionen!

Arbeiten mit Haskell

Arbeitsumgebung, Übungsblatt 0

2.11.2010

Arbeitsumgebung

- Informationen zu Haskell und Werkzeugen erhältlich unter www.haskell.org
- In PI3 arbeiten wir mit dem **GHC** (Glasgow Haskell Compiler), bzw. GHCi (... interactive)
- Es gibt einen hilfreichen **Haskell Mode for Emacs**

