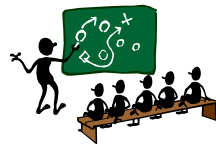


Einführung in die Methoden der Künstlichen Intelligenz [KI] – SoSe 2007

Übungen – Organisatorisches



Prof. Dr. Ingo J. Timm, Andreas D. Lattner
Wirtschaftsinformatik und Simulation

Allgemeines

- Kontakt: adl@tzi.de
- <http://www.informatik.uni-frankfurt.de/~bachki>

- Diplom vs. Bachelor
 - Diplom
 - Übungszettel, mind. 50%
 - Regelmäßige und aktive Teilnahme an den Übungen
 - Bachelor
 - Mündliche Prüfung
 - „Bonus“ durch Übungszettel
 - Gruppenarbeit mit 2-3 Personen empfohlen

- Eintragen in Teilnehmerlisten am Ende der Übung



Übungszettel

- Übungszettel werden in der Vorlesung verteilt; i.d.R. 2 Wochen vor Abgabetermin
- Abgabe **vor Beginn** der Vorlesung (auch elektronisch)
- spätere Abgaben werden nicht gewertet
- Schriftliche Abgabe
 - Keine handschriftlichen Abgaben der Lösungen
 - Auch (wichtige) Sourcen in das Dokument integrieren
 - Alles (nicht nur Sourcen) dokumentieren
 - Auf Nachvollziehbarkeit der Lösung achten!
 - (Alle Gruppenmitglieder nennen)
- Elektronische Abgabe
 - per E-Mail (zunächst) an **adl@tzi.de**
 - Source-Code (Java)
 - Programme als **ausführbare JARs**
 - Auch Testfälle dokumentieren
 - elektronische Version der Lösung als PDF abgeben
 - Bitte nur zip/tar.gz-Archive
- Programmiersprache: nur JAVA (muss mit JDK6u1 laufen)
- Sourcecode gut dokumentieren

Form & Lesbarkeit der Abgaben

- Schriftgröße ≥ 10 pt, bei 2-auf-1-Seitendruck ≥ 12 pt
- Kein Buchstabensalat (z.B. „verrutschte“ Positionen von Buchstaben)
- Kein „plain text“ (ASCII)
- Ein einheitliches Dokument pro Gruppe (kein „Flickenteppich“)
- Keine Zeilenumbrüche mitten im Text
- Quelltexte in nicht-proportionaler Schrift (z.B. Courier); gut lesbar formatiert
- Keine losen Blätter
- Bewertung kann durch unzureichende Form beeinflusst werden



Programmieraufgaben

- In die Bewertung von Programmieraufgaben fließt ein
 - Funktionalität
 - Dokumentation
 - Test
 - Flexibilität der Lösung/Softwareentwurf
- Wichtige Punkte
 - Lösungen möglichst flexibel gestalten, damit auch Varianten des Problems gelöst werden können
 - Komplexere Programme ordentlich strukturieren, **nicht** alles in eine Klasse/Methode

