

Programmierpraktikum

WS04/05

Institut für Informatik

Ludwig-Maximilians-Universität München

Organisatorisches

Ablauf

1) Plenum Mi 14-18 E04, Schellingstr. 3

- Vermittlung von Inhalten
- Meilensteine / Hausaufgaben
- Diskussion
- Ankündigungen
- Organisatorisches

2) Gruppentreffen wöchentlich 1-std. nach Vereinbarung

- individuelle und Gruppenarbeit darlegen

Anwesenheitspflicht, Fehlen bei (2) nur entschuldigt!

Team

Praktikumsleiter:

- Martin Lange
- Ralph Matthes

Gruppenbetreuer:

- Cornelius Dirmeier
- Markus Jehle
- Mathis Koblin
- Nicolas Rachinsky
- Robert Reitmeier
- Jörg Schmidl
- Olha Shkaravska
- Ilja Vishnevski
- Andriy Zherdin

Kommunikation

1) Plenum

2) Webseite

`http://www.tcs.ifi.lmu.de/lehre/WS04-05/ProgPrakt`

3) Diskussionsforum

`http://www.die-informatiker.net`

4) Mailinglisten

`http://lists.tcs.ifi.lmu.de/mailman/listinfo/
ppgruppenname`

5) Automatische Emails

besteht aus Teilen

- neuen **Stoff** vermitteln
- **Ankündigungen**
- **Vorgaben**, z.B. Def. von Schnittstellen
- **Diskussion**

Gruppen

Zulassung

Zulassung erteilt aufgrund **nachgewiesener Vorkenntnisse**

bei einigen muss Nachweis noch erbracht werden

→ **Aufforderung** (per Mail) zur Scheinvorlage

Stichtag Mittwoch, 27.10.

Bestätigung des Platzes durch

- **Unterschrift**
- **Terminvereinbarung** mit Gruppenbetreuer bis Montag, 25.10.

Gruppendynamik

zu Beginn maximal: 5 Teilnehmer pro Gruppe

Gruppen sind

- schrumpfbar (z.B. durch Disqualifikationen)
- zusammenlegbar (z.B. bei zu grosser Schrumpfung)
- auflösbar (z.B. bei Nichtfunktionieren)

Praktikum ist kein “Abgeben in Gruppen zu fünft”

Leistung individuell in einer Gruppe erbracht

Gruppenübergreifende Arbeit

erwünscht:

- Diskussion über Probleme / Lösungsansätze im Rahmen von
 - Plenum
 - Forum
- Kompatibilitätstests

verboten:

- weitergeben von Quellcode

Bewertung

Bewertung

erfolgreiche Teilnahme:

- wöchentlicher Leistungsnachweis in Gruppentreffen
- Zwischenabnahme durch GB/PL
- Zwischenabnahme für / durch andere Gruppe, Bericht
- Endabnahme
- keine Disqualifikation

~> unbenoteter Schein

besondere Leistungen werden bescheinigt

Bewertung

bei wöchentlichen Treffen darlegen

- eigene Arbeit
- Gesamtstruktur

individuelle Prüfungen möglich
(z.B. bei Verdacht auf Passivität)

Disqualifikationen

gelbe Karte (Verwarnung) u.a. für



- Nichtbeachten von Vorgaben
- unentschuldigtes Fehlen
- kein wöchentlicher Leistungsnachweis
- individuelle Prüfung nicht bestanden
- Weitergeben / Annehmen fremden Quellcodes

rote Karte (Ausschluss) u.a. für



- Angabe falscher Tatsachen bei Anmeldung
-  +  = 

Inhalt

Inhalt

realitätsnahes Software-Projekt, aber nicht unbedingt realistisch

Aspekte:

- Objekt-Orientierung (Java 1.5)
- graphisches I/O
- verteilte Systeme, Netzwerk
- Security
- Algorithmik
- AI
- ...

ohne entsprechende (Java-)Vorkenntnisse wohl kaum zu schaffen

Vorgaben

Vorgaben, z.B. durch Interfaces

- müssen unbedingt eingehalten werden
- werden minimal gehalten

“Austoben” erlaubt und erwünscht

- Hinblick auf Wettbewerb
- Praktikum ist kein Nachprogrammieren
- künstlerische Freiheit

Vorschläge durch Gruppenbetreuer

- aufgreifen und umsetzen

Das Projekt

Spiel “Würmer versenken”

- wabenförmiges **Spielbrett**
- freie **Felder**, Hindernis, 1 Loch
- 1-5 Spieler
- jeder Spieler hat n **Würmer** der Länge l
- Spieler ziehen abwechselnd mit **Zeitlimit** (pro Zug und gesamt)
- **Zug**: Spieler zieht einen Wurm um ein Feld weiter
- **Ziel**: alle Würmer ins Loch navigieren
- **Sieger**: erster Ziel erreicht
- **Unentschieden** möglich

Anforderungen

zu programmieren:

- **Server:** Spielbrett, Scheduler, Rekorder
- **Client:** graphische Darstellung, interaktiver Spieler
- **Tools:** Chatserver, Spielbretteditor
- **automatische Spieler** (Strategien): mind. 1 nicht-trivialen!

absolut wichtig: jeder Server mit jedem Client funktioniert

zu leisten:

- **Dokumentation**
- **Testbericht** über andere Gruppe

Wettbewerb

am Ende Kür von drei Siegern in Kategorien:

- **Programmstruktur**
programmtechnisch beste Lösung
- **Ergonomie**
beste Bedienfreundlichkeit Client+Server
- **Strategie**
Sieger des Wettbewerbs jeder gegen jeden

Gruppeneinteilung

Gruppen

komplett:

PRO_gramming5, epsilon, Team2000, dell4never, OlympicCoders, 23, Icecode, Augustiner, Sosuwayi, Augustinerburschen, Freimaurer, Gerch

noch 1 Teilnehmer:

Wurscht, MAKK, Codemasters, hanabit, tetasenopecho

noch 2 Teilnehmer:

MensaWG, B.I.E.R

keine Gruppenstärke erreicht:

23 Teilnehmer

erst auffüllen, dann noch drei Gruppen bilden

Quellcodeverwaltung

Subversion

Versionskontrolle <http://subversion.tigris.org/>

Dokumentation: <http://svnbook.red-bean.com/>

Nachfolger von CVS

Versionskontrolle von grösseren Projekten über HTTP

gemeinsamer Schreib-/Lesezugriff mit Konfliktvermeidung

SVN-Server verwaltet *Repository*:

- Inhalt des Projektes
- Info über Änderungen
- Historie
- ...

hier: `http://svn.tcs.ifi.lmu.de/repos/ppgruppenname`

Zugriff über SVN-Client (Kommandozeile, GUI)

aktuelle Version verwenden!

SVN-Client-Kommandos

- lokale Kopie anlegen
`svn checkout Repository-URL`
- lokale Kopie aktualisieren
`svn update`
- Datei zu Repository hinzufügen
`svn add Dateiname`
- Datei umbenennen / verschieben
`svn move Datei1 Datei2`
- Datei löschen
`svn delete Dateiname`
- lokale Änderungen ins Repository schreiben
`svn commit`

Zu beachten

keine Änderungen durch `add`, `move`, `delete`, erst durch `commit`

`commit` nur nach `update` möglich

`update` kann Konflikte erzeugen

- teils automatisch gelöst
- teils nur von Hand lösbar

Zugriffe passwortgeschützt

Entwicklungsumgebungen

nicht vorgeschrieben, aber empfohlen, z.B.

NetBeans 4.0Beta2

<http://www.netbeans.org/>

Eclipse 3.0 (mit Cheetah-Plugin)

<http://www.eclipse.org/>

Java 1.5

Neuerungen

siehe <http://java.sun.com/j2se/1.5.0/index.jsp>

Auszug:

Some highlights of J2SE 5.0:

- * New language updates: Metadata, Generics, Enumerated types, Autoboxing of primitive types
- * New JVM Monitoring and Management API
- * Improved out-of-box performance
- * New (but compatible) default Java look and feel

Hausaufgaben

Hausaufgaben

1. Benutzung (evtl. Installation) eines SVN-Clients
2. Kompilation des Java-1.5-Programms (auf Webseite)
3. Eintragung auf Mailingliste der Gruppe
4. Termin für wöchentliche Treffen mit Gruppe+GL ausmachen