

Javacrashkurs Semesterferien FSS 2012

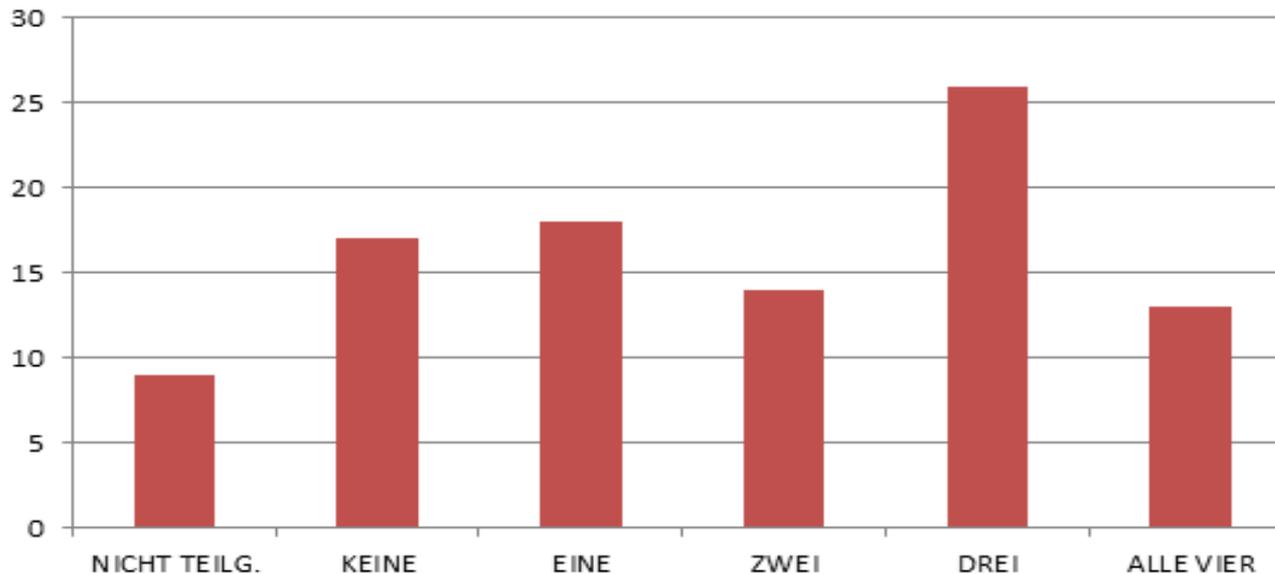
10000 ZEILEN JAVA IN 4 TAGEN

Lehrstuhl Stuckenschmidt

Einführung und Überblick

Hintergrund & Motivation

- PI-2 Programmiertests FSS 2012
 - Keine guten Ergebnisse, ... einige Kollegen drücken es etwas negativer aus
 - Vorbereitung auf 2ten Versuch



Hintergrund & Motivation

- Andere Fächer in denen Programmierkenntnisse notwendig sind
 - Atkinson/Janjic: Softwaretechnik + Praktikum
 - Stuckenschmidt/Meilicke: Künstliche Intelligenz I
 - Auslandsstudium und diverse Wahlfächer
- Bachelorarbeit (+ evtl. Seminararbeit) ohne Programmieren?
 - Betreuer: "Na dann muss es aber theoretisch schon anspruchsvoll sein, wenn man das überhaupt bei uns machen kann"
- Praktikum

Hintergrund & Motivation

- Und was kommt nach dem Studium?
- Jobs bei denen man nicht programmieren muss?
 - Was möchte ich später mal machen?
 - Welchen Status möchte ich dort haben wo ich arbeite?
- Kleine Programmieraufgaben sind nicht selten Teil eines Bewerbungsverfahrens
 - Blatt 1: Fizz-Buzz Aufgabe (5-15 Minuten)
 - Blatt 1: Rabe Aufgabe (20-60 Minuten)

Moralpredigt ...

- ... ist hier zuende
- Wir wollen euch helfen **besser/scheller/sicherer** in Java **programmieren** zu können!
- Wir wollen euch zeigen, wie und dass ihr **selbstständig üben** könnt/müsst!

Jeder Tag = eine neue Herausforderung

- 10:00 bis ca. 12:00: Grundlagen für (nahezu) alles was nachmittags benötigt wird
 - Folien, hier und da Code Beispiele
 - Nachmittags verfügbar zum Nachschauen
- Ab ca 13:00 Uhr Betreutes Üben
 - Ausreichend Aufgaben vorbereitet
 - Aufgabenzettel abholen und los geht's
 - Am Ende einem Betreuer mitteilen, wie weit gekommen
- Gesamter Lehrgang + Aufgabenzettel erst am Freitag verfügbar + Lösungen (vielleicht)

Bearbeitung der Aufgaben I

- Schwierigkeitsgrad von leicht bis schwer
- Jede Aufgabe gründlich bearbeiten
 - Auch wenn du denkts sie ist zu einfach
 - Löse die Aufgabe korrekt und schnell
- Kontrolliere ob Aufgabe korrekt gelöst ist
 - Ab Tag 2 sind zum Teil Testfälle mitgeliefert
 - Ansonsten main() laufen lassen, Ausgaben analysieren und selbst entscheiden ob korrekt gelöst
 - Im Zweifelsfall Betreuer oder Nachbarn fragen

Bearbeitung der Aufgaben II

- Beim Nachbarn schauen
 - Ist erlaubt aber erst nachdem man selbst *lange* herum probiert hat
 - **Falsch** Code der Kommilitonin anschauen, sich vielleicht erklären lassen, und verstehen => nächste Aufgabe
 - **Richtig** Wie oben, aber: In jedem Fall am eigenen Rechner eintippen und laufen lassen
- Hilfe deinen Kommilitonen
 - Wieso? => Einfach so!



Bearbeitung der Aufgaben III

- Was wenn ich es einfach nicht hinbekomme?
 - Versuch es nochmal!
 - Frag um Hilfe (Betreuer, Nachbar)
 - Freu dich über kleine Erfolge
- Was wenn ich am Ende des Tages nicht fertig werde mit allen Aufgaben?
 - Ist bei den meisten zu erwarten,
 - Verbleibende Aufgaben als weiteres Übungsmaterial
 - **Nächster Tag: Mit neuem Blatt starten**

Bearbeitung der Aufgaben IV

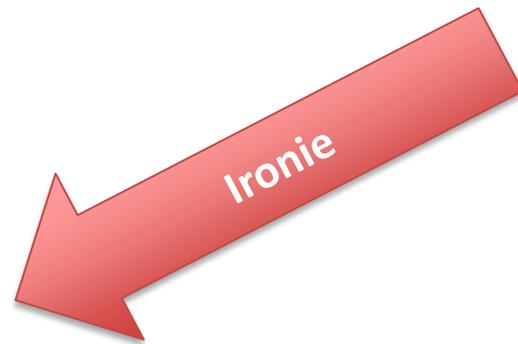
- Königsaufgabe oder eine andere schwere Aufgabe korrekt gelöst?
 - Wird mit Piraten Aufkleber belohnt
 - Gibt's beim Betreuer am Ende



Bearbeitung der Aufgaben V

- Ach, ich mach das alles lieber daheim ... da kann ich zwischendrin ein bisschen Staubsaugen oder Fernsehen

- Ja klar!



Tag 1 - Inhalt

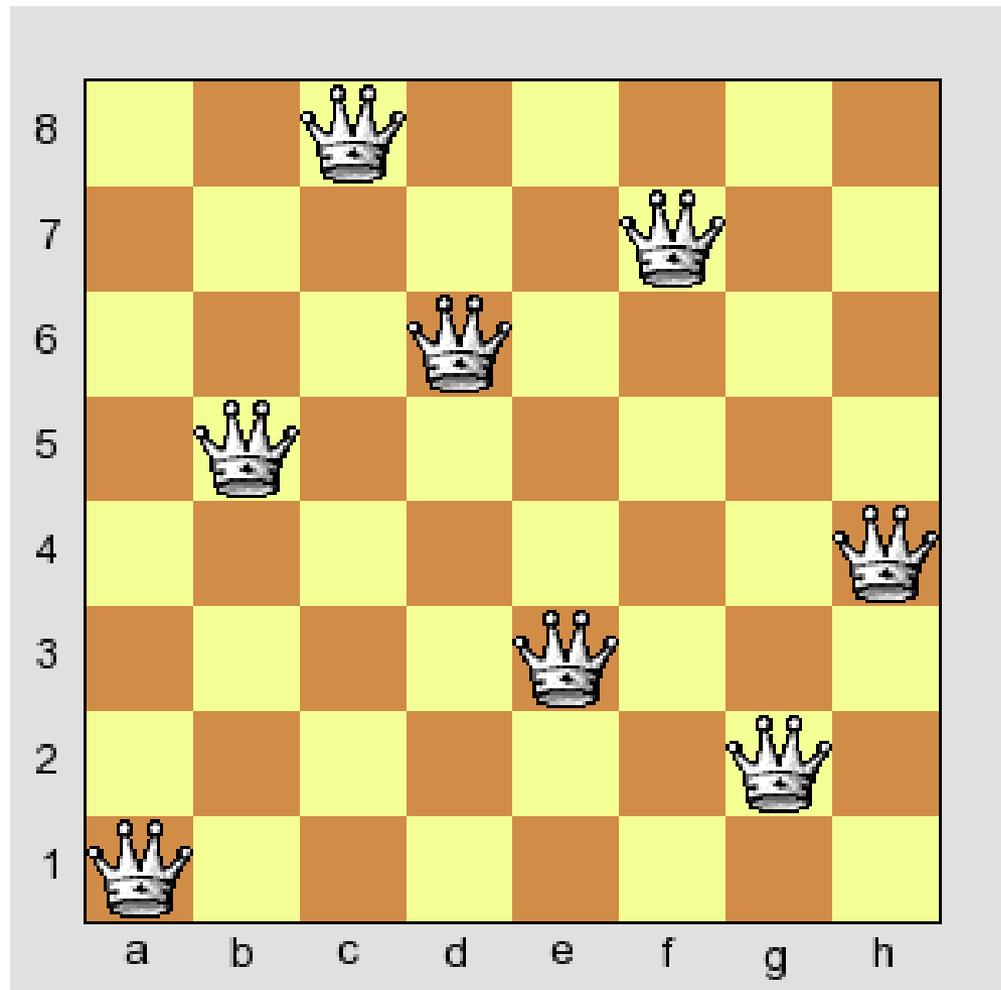
- Schleifen und Verzweigungen
- Primitive Datentypen
- Deklaration und Initialisierung von Variablen
- Rechenoperationen
- Keine Arrays, keine Methoden, keine Objekte, alles in der main-Methode

- Ziemlich wenig, oder? Du wirst sehen, dass die Aufgaben gegen Ende des Blatts nicht einfach zu lösen sind.

Tag 2 - Inhalt

- Arrays
- Methoden
 - Kein Spaghetti Code mehr wie bei Tag 1
 - Oft Testmethoden gegeben, ähnlich wie in der Programmierklausur
 - Unklusive Rekursion
- Nicht viel mehr? Aber kannst du damit das 8-Damen Problem lösen?

Tag 2 - Beispielaufgabe



Tag 3 - Inhalt

- Objektorientierung
 - Erzeugen und Verwenden von Objekten
 - Objektmethoden/variablen vs. Klassenmethoden/variablen
 - Vererbung
 - ...

Tag 4 - Inhalt

- Ausgewählte Datenstrukturen
 - ArrayList
 - HashMap
 - HashSet
 - PriorityQueue
- Beispielaufgaben, an denen man sehen kann wann/wie man diese Datenstrukturen einsetzt

Lernziel insgesamt

- Programmieren einfacher Algorithmen
- Erkennen typischer Muster
- Wichtige Datenstrukturen kennen/verwenden
- Sicherheit im Umgang mit Fehlern/Eclipse
- Weitere Anmerkungen
 - Objektorientiertes Programmieren?
 - Keine GUIs?
 - Vorbereitung auf PI2 Test?
- **Selbstwirksamkeit = Vertrauen es lernen zu können und es letztlich zu beherrschen**

Weg zum Erfolg

(nochmal Moralpredigt)

- Lernziele können nur erreicht werden, wenn man aktiv versucht jede Aufgabe zu lösen
 - Scheitern vor allem verursacht durch mangelnde Anstrengung
 - Erfolg = Produkt der eigenen Anstrengung und der eigenen Fähigkeiten
 - Erfolg immer relativ zu der eigenen Ausgangslage sehen
- Was nicht weiter hilft
 - "Ich versteh den Mist nicht ...?"
 - "Die Aufgabe ist blöd, was soll man genau machen?"
 - "Das schaff ich nie, ich bin eben nicht so gut!"
 - "Ich studiere **Wirtschaftsinformatik** und nicht Programmieren!

Jeder Tag = eine neue Herausforderung

- 10:00 bis ca. 12:00: Grundlagen für (nahezu) alles was nachmittags benötigt wird
 - Folien, hier und da Code Beispiele
 - Nachmittags verfügbar zum Nachschauen
- Ab ca. 13:00 Uhr Betreutes Üben
 - Ausreichend Aufgaben vorbereitet
 - Aufgabenzettel abholen und los geht's
 - Am Ende einem Betreuer mitteilen, wie weit gekommen
- Gesamter Lehrgang + Aufgabenzettel erst am Freitag verfügbar + Lösungen (vielleicht)

**Viel Spass und viel Erfolg
in den nächsten vier Tagen!**

(das mit den 10000 Zeilen ist vermutlich nicht ganz richtig,
es werden wohl zwischen 500 und 3000 werden)