

## Vorlesung "Spezielle Themen der Softwaretechnik"

Freie Universität Berlin, Institut für Informatik, Arbeitsgruppe Software Engineering

Prof. Dr. Lutz Prechelt, Stephan Salinger

Übungsblatt 8

WS 2007/2008

zum 2008-01-28

Auf diesem **über zwei Wochen laufenden Übungsblatt** wollen wir uns mit der Cleanroom-Methode beschäftigen und ein kleines Spiel unter Verwendung dieses Verfahrens implementieren.

### Aufgabe 8-1: (Mastermind mit Cleanroom implementieren)

In dieser Aufgabe sollen Sie das Logikspiel Mastermind unter Verwendung der Cleanroom-Methode (ohne „Statistical testing“) entwickeln. Beachten Sie dabei folgende Punkte:

- Die Regeln von Mastermind finden Sie z.B. unter <http://de.wikipedia.org/wiki/Mastermind>. Implementieren Sie die Variante mit 4 „Steckplätzen“, 6 Farben und keiner Doppelverwendung von Farben.
- Ihr Programm sollte in zwei verschiedenen Modi gestartet werden können:
  - 1) Spielmodus 1: Das Programm bestimmt eine Farbkombination, die der Aufrufer des Programms „erraten“ muss. Nach jedem Rateversuch bewertet das Programm den Zug.
  - 2) Spielmodus 2: Der Rechner rät eine Farbkombination, die sich der Aufrufer des Programms ausgedacht hat. Nach jedem Rateversuch des Programms gibt der Aufrufer die Bewertung des Zuges ein.
- **Schreiben Sie Ihr Programm in Java.**
- **Schreiben Sie keine graphische Benutzungsoberfläche für das Programm.** Dies würde den Umfang dessen, was wir in dieser Aufgabe über Cleanroom erfahren wollen sprengen. Arbeiten Sie also mit Ausgaben auf der Konsole und benutzen Sie dabei z.B. folgende Darstellungen:
  - Anstatt Farben verwenden Sie die Zahlen 1,2,3,4,5,6.
  - Verwenden Sie die Buchstaben s und w für die Bewertungsfarben Schwarz und Weis.
  - Bei Spielmodus 1 gibt das Programm bei jeder Bewertung immer das gesamte Spielbrett aus. Dies könnte nach dem dritten Zug des Spielers z.B. wie folgt aussehen:

3.	ws..	5316
2.	wws.	3526
1.	ww..	1234
- Bei Spielmodus 2 sollten Sie sich vor allem darauf konzentrieren, dass das Programm logisch richtige Spielzüge macht. Über eine gute Strategie können, müssen Sie sich nicht Gedanken machen. Allerdings wollen wir die „Spielstärke“ der Programme in der Übung vergleichen. Dies sollte Sie motivieren, ein möglichst starkes Programm zu entwickeln.
- Natürlich können Sie im Internet diverse Hinweise (oder vielleicht sogar Implementierungen) finden. Verderben Sie sich aber nicht den Spaß und den Lerneffekt und sehen Sie sich diese Lösungen auf keinen Fall an.
- Außerdem kann nicht kontrolliert werden, ob Sie Ihr Programm nicht doch testen. Auch hier gilt: **Verderben Sie sich nicht den Lerneffekt und testen Sie Ihr Programm nicht.** Allerdings ist es Ihnen erlaubt Ihr Programm in einer IDE zu entwickeln und somit einen Syntaxcheck zu verwenden.
- **Implementieren Sie das Programm unter Verwendung der Cleanroom-Methode.** Sehen Sie sich deshalb die entsprechenden Folien aus der Vorlesung vorher noch einmal genau an und benutzen Sie die dort dargestellt halbformale Verifikation. Falls Ihnen die Informationen in den Folien nicht reichen, so recherchieren Sie im Internet nach weiteren Quellen.
- Beachten Sie noch folgendes:

- **Arbeiten Sie auf jeden Fall mindestens zu zweit.** Ihre halbformale Darstellung muss ausreichen, um sich gegenseitig und später in der Übung auch Ihre Kommilitonen von der Richtigkeit jedes Schrittes zu überzeugen.
- **Bringen Sie als Ergebnis folgendes in die Übung am 28.01. mit:**
  - Das fertige nicht getestete Programm ausführungsbereit auf Ihrem Rechner (Laptop).
  - Die gesamte Schrittweise Zerlegung („Boxes“) mit den zugehörigen Verifikationen als Textdokument auf Ihrem Rechner. Bereiten Sie sich darauf vor, dies in der Übung vorzustellen.