Vorlesung "Spezielle Themen der Softwaretechnik"
Freie Universität Berlin, Institut für Informatik, Arbeitsgruppe Software Engineering
Prof. Dr. Lutz Prechelt, Martin Gruhn
Übungsblatt 8 WiSe 2008/2009 zum 2009-01-12 und 2009-01-19

Lernziele:
- Cleanroom-Methode wiederholen und verstehen.
- Praktische Erfahrung mit Cleanroom sammeln (Aha-Effekt).

Aufgabe 8-1: Mastermind mit Cleanroom implementieren
In dieser Aufgabe sollen Sie das Logikspiel Mastermind unter Verwendung der Cleanroom-Methode (ohne „Statistical testing“) entwickeln.

Anforderungen an das Programm:
- *Spielregeln*: Richten Sie sich nach den Regeln von Mastermind unter http://de.wikipedia.org/wiki/Mastermind. Implementieren Sie die Variante mit 4 „Steckplätzen“, 5 Farben und keiner Doppelverwendung von Farben.
- *Spielmodi*: Ihr Programm sollte in zwei verschiedenen Modi gestartet werden können:
  1) Spielmodus 1: Das Programm bestimmt eine Farbkombination, die der Nutzer „erraten“ muss. Nach jedem Rateversuch bewertet das Programm den Zug.
  2) Spielmodus 2: Das Programm rät eine Farbkombination, die sich der Nutzer ausgedacht hat. Nach jedem Rateversuch bewertet der Nutzer den Zug.
  Wir werden die Spielstärken der Programme in der Übung vergleichen. Bitte überlegen Sie sich selbst eine sinnvolle Strategie[1].
- *Programmiersprache*: Schreiben Sie Ihr Programm in Java.
- *Benutzeroberfläche*: Schreiben Sie eine zeichenbasierte (keine graphische!) Nutzerschnittstelle für das Programm. Benutzen Sie dabei folgende Darstellung auf der Konsole:
  - Anstatt Farben verwenden Sie die Zahlen 1,2,3,4,5.
  - Verwenden Sie die Buchstaben s und w für die Bewertungsfarben Schwarz und Weiß.
  - Beispiel für die Ausgabe des Spielbrettes nach dem dritten Spielzug:
    ```
    1.    1234  ww..
    2.    3526  wws.
    3.    5316  ws..
    ```
  - Es ist Ihnen überlassen, ob Sie das Spielbrett nach jedem Spielzug immer vollständig ausgeben oder die Spielzüge nacheinander erscheinen.

Anforderungen an Ihr Vorgehen:
- *Cleanroom-Methode*: 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.
- *Keine Tests*: Sie dürfen das Programm während der Entwicklung **nicht ausführen!** Allerdings sollten Sie Ihr Programm in einer IDE entwickeln und somit einen Syntaxchecker verwenden. Der erste Test darf **erst am 19.01. in der Übung** erfolgen!
- *Gruppenarbeit*: Arbeiten Sie **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.

---

[1] 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 **nicht** an.

Aufgabe 8-2: Fragen zur Cleanroom-Methode
Zum 12.01.: Notieren Sie Ihre Antworten zu folgenden Fragen kurz und präzise (in wenigen Sätzen).

1. Wie erfolgt die Verifikation bei Cleanroom?
2. Wann eignet sich Cleanroom gut und wann nicht? Warum?
3. Inwiefern ist die Cleanroom-Methode für diese Aufgabe geeignet?
4. Nennen und charakterisieren Sie die Kernidee der Cleanroom-Methode?

Optional: Schicken Sie Ihre Antworten auf die Fragen bis Montag, den 12.01., 9 Uhr an mgruhn at inf... Verwenden Sie folgenden Betreff: *[SWT2 UE8-2] Vorname Nachname*

Aufgabe 8-3: Abschlusspräsentation: Mastermind im Cleanroom
Zum 19.01.: Bringen Sie folgende Dinge mit in die Übung:

- 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.