

Martin Slowikowski
Matrikelnummer: 199 91 66

Jan-Tristan Rudat
Matrikelnummer: 200 78 52

Teamname: **Bernie und Ert**

Inhaltsverzeichnis:

- I. Funktion
- II. Architektur, Feststellungen
- III. Mängel, Fragen
- IV. Hilfsmittel & Quellen

Ausführliche Dokumentation zu Aufgabe 7

I. Funktion

In den letzten zwei Wochen haben wir an einer vollständigen Implementierung des Spieles MasterMind gearbeitet. MasterMind kann in dieser Version entweder gegen den Computer gespielt werden, oder der Computer kann eine von Ihnen eingegebene Zahl mit Hilfe des "Knuth-Algorithmus" erraten.

Im deutschen Wikipedia Eintrag wird MasterMind wie folgt beschrieben:

Mastermind, auch **SuperHirn**, in der DDR auch als **Super Code** oder **Variablo** bekannt, ist ein Logikspiel für zwei Personen, das mit farbigen Stiften gespielt wird, die in ein spezielles Gestell gesteckt werden. Das Spiel wurde 1970 von **Meirowitz** erfunden und entwickelte sich zu einem der erfolgreichsten Spiele der 1970er. Bis 2000 wurden über 55 Millionen Spiele in 80 Ländern verkauft.

In unserer Version von MasterMind muss der Spieler eine Zahl erraten, welche zwischen 0123 und 9876 liegt. Die Zahl ist so gestrickt, dass keine Ziffer doppelt vorkommt, die Null kann an jeder Position vorkommen, natürlich auch nicht doppelt.

II. Architektur:

Die Klasse **MasterMindApp** beinhaltet die MainClass. Von hier aus geht der Aufruf des Programms aus. Verschiedene Parameter werden an Klassen übergeben und es werden diverse Objekte erzeugt.

Die Klasse **RandomNumber** dient der Erzeugung einer Zufallszahl nach den oben beschriebenen Kriterien. Die Klasse hat eine Methode, welche die Zufallszahl in einem Integer Array der Länge 4 zurückgibt; jeder Index in dem Array ist eine Ziffer der Zufallszahl.

Die Klasse **MasterMindIO** wickelt sämtliche Ein- und Ausgaben auf der Konsole ab.

Die Klasse **CheckNumber** prüft, ob eine Zahl den oben vorgegebenen Kriterien entspricht.

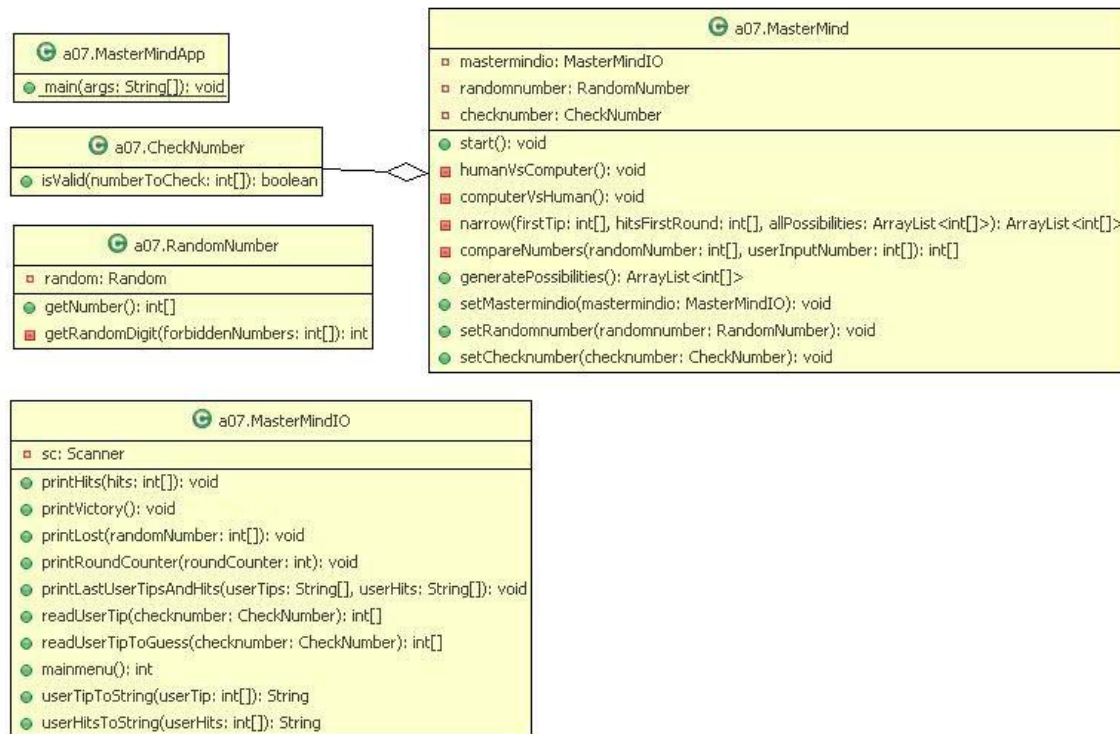


Abb. 1: UML Diagramm zu MasterMind

Die Klasse **MasterMind** beinhaltet die "restliche" Logik zum Ablauf des Programms.

Im wesentlichen sehen unsere Klassen wie in Abbildung 1 aus. Wir denken, dass die Methodenbezeichnungen die "Funktionen" gut widerspiegeln. Die Funktionen der einzelnen Methoden haben wir im javadoc Stil auskommentiert, damit wir gut partnerübergreifend arbeiten konnten.

In der Klasse MasterMind wird unter anderen der Algorithmus beschrieben, welcher den Computer eine Zahl erraten lässt. Durchschnittlich schafft er dies in vier bis fünf versuchen.

III. Mängel

Bei einigen Bezeichnungen von Attributen waren wir uns nicht einig. Ebenso lässt sich die ein oder andere Methode sicherlich schöner programmieren.

IV. Hilfsmittel und Quellen:

Quellen: <http://de.wikipedia.org/wiki/Mastermind> zuletzt gesehen am 19. Mai 2010

[http://en.wikipedia.org/wiki/Mastermind_\(board_game\)](http://en.wikipedia.org/wiki/Mastermind_(board_game))
zuletzt gesehen am 18. Mai 2010

<http://www.tutorials.de/forum/java-grundlagen/326837-mastermind.html>