# Projekt 1: Mastermind

## Ausgangslage

Der Modulplan sieht vor, dass Sie während den ersten fünf Lektionen die Programmiersprache Java und ihre Grundlagen kennenlernen. Eine Programmiersprache erlernt man am besten, indem man selber möglichst viel programmiert. Als Übungsfeld dient uns für diese Zeit das Spiel Mastermind. Sie werden im einfachsten Fall ein konsolenbasiertes Spiel realisieren. Das Feld ist aber offen für individuelle Erweiterungen, je nach Vorkenntnissen, bis hin zu einer grafischen Benutzeroberfläche.

Damit Sie sich auf die Sprache selber konzentrieren können, werden wir in diesem Projekt nochmals prozedural arbeiten und an bereits bekannte Techniken aus dem Modul 403 anknüpfen. Da Java eine objektorientierte Sprache ist, werden wir gleichwohl eine Klasse brauchen. Allerdings reicht eine davon.

## Ziele

* Sie können ein Java-Programm (eine Klasse) mit einer main-Methode erstellen.
* Sie kennen die einfachen Datentypen von Java und können damit lokale Variablen und Klassenvariablen deklarieren
* Sie kennen die Operatoren von Java und können damit Ausdrücke formen
* Sie kennen die Grundanweisungen von Java (Zuweisung, Entscheidung und Schleife) in ihren verschiedenen Varianten und können diese anwenden
* Sie können in Java Arrays deklarieren, erzeugen und anwenden
* Sie können in Java Methoden mit Parametern und Rückgabewerten deklarieren und aufrufen
* Sie können in Java Eingaben von der Konsole lesen und Ausgaben auf die Konsole machen.

## Vorgehen

* Arbeiten Sie zu zweit, wobei Sie aber Ihren eigenen Code in Ihrer eigenen Lernumgebung erstellen.
* Wenn Sie alleine nicht weiter kommen
  + suchen Sie Hilfe im Team,
  + konsultieren Sie die Ressourcen
  + lesen Sie im Buch
  + recherchieren Sie im Internet
* Sie werden in den weiteren Arbeitsblättern der Lektionen 1 bis 5 immer wieder auf Arbeitsaufträge stossen, welche Teilprobleme des Spiels Mastermind zum Inhalt haben.

## Anforderungen

In diesem Abschnitt werden die Minimalanforderungen an das zu erstellende Spiel präsentiert.

Selbstverständlich dürfen Sie diese Anforderungen ausbauen oder allenfalls anpassen, solange die Idee des Spiels nicht verloren geht. Wenn Sie dies tun, halten Sie die ergänzenden oder angepassten Anforderungen schriftlich fest und zeigen Sie diese der Lehrperson.

#### Minimalanforderungen

Mastermind ist ein Spiel für zwei Personen, einen Codierer und einen Rater. Sie finden eine Beschreibung des Originalspiels unter <https://de.wikipedia.org/wiki/Mastermind_(Spiel)>. Sie können im Internet auch lauffähige Masterminds finden, teilweise sogar in Java.



In unserem Fall wird der Codierer durch den Computer realisiert und der Rater durch den Spieler.

Folgender Spielablauf soll realisiert werden:

1. Am Anfang bestimmt der Computer einen zufälligen, vierstelligen Geheimcode aus sechs zur Verfügung stehenden Farben.
2. Nun wird der Spieler aufgefordert den Geheimcode zu erraten und einen eigenen Code einzugeben.
3. Der Computer vergleicht den eingegebenen Code mit dem Geheimcode und zeigt dem Spieler an
   1. wie viele Stifte des eingegebenen Codes korrekte Farbe und Platz haben und
   2. wie viele Stifte zwar die korrekte Farbe, aber nicht den korrekten Platz haben.
4. Wenn der Spieler den Geheimcode erfolgreich aufgedeckt hat ist das Spiel zu Ende. Andernfalls wird der Spieler erneut aufgefordert einen Code einzugeben.

Da auf der Konsole keinerlei grafische Ausgabe möglich ist, sollen die Farben durch einzelne Buchstaben dargestellt werden:

r Rot

b Blau

g Grün

y Gelb

w Weiss

s Schwarz

Die Eingabe des Spielers soll jeweils auf einer neuen Zeile eingelesen werden. Sie soll aus einer Folge von vier Buchstaben ohne Zwischenraum bestehen und wird mit der Return-Taste abgeschlossen.

Das Programm soll die Benutzereingabe auf Korrektheit überprüfen (validieren) und im Falle einer fehlerhaften Eingabe eine entsprechende Meldung ausgeben und zur Wiederholung der Eingabe auffordern.

Bei einer gültigen Eingabe soll auf einer neuen Zeile die Bewertung der Eingabe ausgegeben werden (Schritt 3 in obigem Ablauf).

Die Ausgaben des Programms sollen so gestaltet werden, dass dem Benutzer jederzeit klar ist, was von ihm erwartet wird (Benutzerfreundlichkeit).

## Weitere Schritte

Die weiteren Schritte werden Sie im Rahmen der Lektionen 1 bis 5 durchführen. Grundsätzlich geht es dabei um folgendes:

1. Analysieren der Anforderungen  
   Hier geht es darum festzustellen, ob die Anforderungen verständlich, vollständig und einander nicht widersprechend sind. Andernfalls müssen weitere Abklärungen getroffen werden.
2. Entwurf einer Lösung  
   Jetzt müssen Sie verschiedene Dinge festlegen:
   1. Datenstrukturen: Wie wollen Sie im Programm die Werte von Farben, oder dem Geheimcode speichern? Es geht also um Datentypen und Variablen.
   2. Zerlegung des Gesamtproblems in Teilprobleme. Einzelne Teilprobleme müssen eventuell ihrerseits in noch kleinere Teilprobleme zerlegt werden.  
        
      Beispiel:  
      - Bestimmen eines zufälligen Geheimcodes  
      - Ausgeben der Eingabeaufforderung  
      - Einlesen der Benutzereingabe  
      - …
   3. Ablaufstrukturen zur Lösung der einzelnen Teilprobleme (Struktogramme, Ablaufdiagramme, etc.)
   4. Texte für die Benutzerschnittstelle
3. Umsetzung der Lösung  
   Jetzt giessen Sie die Lösung für jedes Teilproblem in eine Java-Methode. Übergeordnete Methoden rufen dabei untergeordnete Methoden auf. Zuoberst steht die main-Methode.
4. Test der Lösung  
   Jede Methode muss getestet werden. Dies kann auf verschiedene Weisen geschehen:
   1. Von Hand durch möglichst repräsentative, auch ungültige, Benutzereingaben
   2. Durch automatisierte Testmethoden

Wir werden dieses Thema noch eingehender anschauen.