

Java Programmierkurs

Mastermind (Games)

Institut: Beuth Hochschule für Technik Berlin
Dozent: Prof. Dr. Christian Forler
Url: <https://lms.beuth-hochschule.de/>
Email: cforler@beuth-hochschule.de

Mastermind

Heute wollen das Spiel [Mastermind](https://tinyurl.com/qg59ecj) (SuperHirn) implementieren. Diese Spiel wurde 1970 von Mordechai Meirovitz designed. Weitere Information Mastermind finden Sie unter <https://tinyurl.com/qg59ecj>. Um zu gewinnen muss der Spieler einen verstelligen Farbcode, der aus sechs Farben ausgewählt wird, rekonstruieren. In unserer Java Variante muss der Spieler eine vier stellige PIN Nummer, die aus den Ziffern 1-6 besteht rekonstruieren. Der Spieler darf insgesamt 12 mal raten. Nach jedem Rateversuch bekommt er in Form von schwarzen und weissen Stiften folgende Information mitgeteilt:

Schwarzer Stift: Ziffer befindet sich an der richtigen Position

Weisser Stift: Ziffer befindet sich an falscher Position

Wenn der richtige Farbcode gefunden wurde, lautet die Antwort viermal schwarz.

Aufgabe 1 (4 Punkte) Mastermind.java

Erstellen Sie eine Klasse `Mastermind` welche die folgenden Variablen/Konstanten enthält.

```
public static final int ROW_SIZE = 4;
public static final int COLORS = 6;
public static final int MAX_GUESSES = 12;
public static final int EMPTY_CELL = '-';

private final char[] emptyRow;
private int[] code;
private int[][] board;
private int turn;
```

Implementieren Sie (unter anderem) die folgenden Methoden.

- `public Mastermind()`
Konstruktur der für die Initialisierung der Objektvariablen verantwortlich ist.
- `public void reset()`
Das Spiel wird resettet.

- `public void print()`
Gibt das Board als ASCII-Art aus. Die Methode soll unter anderem die beiden folgenden Codezeilen enthalten.
 - `System.out.print(java.util.Arrays.toString(board[i]));`
 - `System.out.println(java.util.Arrays.toString(emptyRow));`
- `private void printPegs(int turn)`
Gibt die Pegs für den Zug `turn` auf der Konsole aus. Beispielausgabe:
`Pegs: Black White White\n`. Es empfiehlt sich zwei private Helfermethoden `printBlackPegs` und `printWhitePegs` zu schreiben.
- `public boolean checkWin()`
Die Methode liefert `true` zurück falls der Spieler den Code erraten hat, ansonsten `false`.
- `public boolean timeOver()`
Gibt nach `MAX_GUESSES` Rateversuchen `true` zurück, ansonsten `false`.
- `public boolean isOver()`
Diese Methode testet ob das Spiel bereits zu Ende ist. Falls dem so sei liefert Sie `true` zurück, ansonsten `false`.
- `public boolean guess(int[] guess)`
Diese Methode soll bei einem validen Rateversuch die Membervariablen `board` und `turn` updaten sowie `true` zurückliefern. Bei einem invaliden Rateversuch soll `false` zurückgegeben werden.

Aufgabe 2 (4 Punkte) Das Mastermind Spiel

Schreiben Sie eine `Main`-Klasse die es einem Spieler ermöglicht Mastermind zu spielen. Verwenden Sie dazu Ihre Klasse `Mastermind` aus Aufgabe 1.

Beispielausgabe

```
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
1 2 3 4
```

```
[1, 2, 3, 4] Pegs: Black Black White
[-, -, -, -]
[-, -, -, -]
[-, -, -, -]
```

```
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]
```

.

.

.

1 2 4 2

```
[1, 2, 3, 4] Pegs: Black Black White  
[1, 2, 6, 6] Pegs: Black Black  
[1, 2, 4, 3] Pegs: Black Black Black  
[1, 2, 4, 6] Pegs: Black Black Black  
[1, 2, 4, 5] Pegs: Black Black Black  
[1, 2, 4, 1] Pegs: Black Black Black  
[1, 2, 4, 2] Pegs: Black Black Black Black  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]  
[-, -, -, -]
```

Congratulations! You won. :)

The secret code was: [1, 2, 4, 2]