

## Umsetzung der Spielstrategie im Programm

Die entworfene Spielstrategie soll nun schrittweise in das Java-Programm eingebunden werden. Um die Handlungen und Entscheidungen der Computerspieler nachzuvollziehen, sollen diese vom Programm jeweils ausgegeben werden.

Im Gruppenverzeichnis finden Sie eine leicht erweiterte Version des Meiern-Programms. Die Klasse Spieler wurde um ein Attribut `name` erweitert, welches im Konstruktor gesetzt wird und dessen Wert über die Methode `gibName()` abgefragt werden kann. Es existieren mit `gibWertigkeitZu(pZahl)` und `gibErgebnisZu(pWertigkeit)` zudem zwei Methoden, welche es dem Computerspieler erlauben, Würfelergebnisse korrekt zu bewerten (siehe Überlegungen auf Arbeitsblatt „Wertung von Würfelergebnissen“).

|     | Ausgabe-Anweisungen                      |
|-----|--|
| 1.  | Zeige neue Spielrunde an                 |
| 2.  | Zeige Beginner an                        |
| 3.  | Zeige gewürfeltes Ergebnis an            |
| 4.  | Zeige an, dass Spieler nun am Zug ist    |
| 5.  | Zeige die vom Vorgänger genannte Zahl an |
| 6.  | Zeige Neu-Würfeln-Entscheidung an        |
| 7.  | Zeige Weiterreichen-Entscheidung an      |
| 8.  | Zeige Aufdecken-Entscheidung an          |
| 9.  | Zeige Niederlage an                      |
| 10. | Zeige Sieg an                            |

  

|     | Java-Programmcode der entsprechenden println-Anweisung: <code>System.out.println(...)</code> |
|-----|--|
| 1.  | <code>("——_Neue_Spielrunde_——");</code>  |
| 2.  | <code>("Es_beginnt_" + name);</code>   |
| 3.  | <code>(name + "_wuerfelt_" + kenntBecher.zeigeWuerfelergebnis());</code>                     |
| 4.  | <code>(name + "_ist_am_Zug");</code>   |
| 5.  | <code>(name + "_erhaelt_vom_Vorgaenger_" + vorgaengerzahl + "_genannt");</code>              |
| 6.  | <code>(name + "_entscheidet_sich_neu_zu_wuerfeln");</code>                                   |
| 7.  | <code>(name + "_entscheidet_sich_den_Becherweiter_zu_geben");</code>                         |
| 8.  | <code>(name + "_entscheidet_sich_den_Becheraufzudecken");</code>                             |
| 9.  | <code>(name + "_verliert");</code>   |
| 10. | <code>(name + "_gewinnt");</code>  |

### Aufgaben

1. **Struktogramm erweitern** Erstellen Sie ein erweitertes Struktogramm zur Methode `fuehreZugAus`, indem Sie die Anweisungen aus der ersten Tabelle an geeigneter Stelle als neuen Strukturblock einfügen, damit die jeweiligen Handlungen und Entscheidungen der Spieler im Spielverlauf nachvollzogen werden können.



2. **Implementierung** Übersetzen Sie das erweiterte Struktogramm in Java-Programmtext, indem Sie die Methode *fuehreZugAus()* so abändern, dass Sie dem Struktogramm entspricht.

Erinnern Sie sich dabei an die Java-Syntax für Verzweigungen. Verwenden Sie zur Ausgabe der jeweiligen Handlungen und Entscheidungen der Spieler geeignete `println`-Anweisungen aus der zweiten Tabelle.

3. **Erweiterung der Methode `starteSpielrunde()`** Fügen Sie in der Methode *starteSpielrunde()* geeignete `println`-Anweisungen aus der zweiten Tabelle ein, damit man anschließend die Startphase des Spiels auf dem Bildschirm nachvollziehen kann.

