

# 1 Grundlagen

## 1.1 Aufgabe

Implementiere eine Funktion `showTriesLeft`. Dieser wird die Anzahl der übrigen Versuche als Integer übergeben. Sie gibt einen String zurück, in dem steht wie viele Versuche der Spieler noch hat.

```
1 showTriesLeft(3)
```

3 Versuche übrig

```
1 showTriesLeft(1)
```

1 Versuch übrig

Beachte das e in Versuche.

## 1.2 Aufgabe

Implementiere eine Funktion `contains`. Diese prüft, ob ein Buchstabe in einem Wort enthalten ist..

```
1 contains("GHSE", 'G')
```

true

```
1 contains("Informatik", 'y')
```

false

## 1.3 Aufgabe

Implementiere eine Funktion `contains`. Diese prüft, ob ein Buchstabe in einer Liste von Buchstaben enthalten ist.

```
1 contains(listOf('G', 'H', 'e', 'S', 'E'), 'G')
```

true

```
1 contains(listOf('I', 'n', 'f', 'o', 'r', 'm', 'a', 't', 'i', 'k'), 'y')
```

false

## 1.4 Aufgabe

Implementiere eine Funktion `compTriesLeft`. Dieser werden die Anzahl der übrigen Versuche vor dieser Runde, das Wort, das erraten werden soll und der zuletzt geratene Buchstabe übergeben. Sie gibt die Anzahl der übrigen Versuche nach dieser Runde zurück.

```
1 compTriesLeft("GHSE", 'G', 3)
```

3

```
1 compTriesLeft("Informatik", 'y', 10)
```

9

Nutze contains

## 1.5 Aufgabe

Implementiere eine Funktion `showGuessedCharsAndBars`. Dieser werden das Wort, das erraten werden soll und eine Liste mit Buchstaben übergeben. Sie gibt einen String zurück. In diesem wurden alle Buchstaben aus dem übergebenen Wort, die nicht in der Liste sind durch Bindestriche ersetzt. Alle anderen Buchstaben wurden unverändert übernommen.

```
1 showGuessedCharsAndBars("GHSE", listOf('G', 'H', 'e', 'S', 'E'))
```

GHSE

```
1 showGuessedCharsAndBars("Informatik", listOf('I', 'a', 'y'))
```

I-----a---

Nutze contains

## 1.6 Aufgabe

Implementiere eine Funktion `checkGuessedWholeWord`. Dieser werden das Wort, das erraten werden soll und eine Liste mit Buchstaben übergeben. Sie gibt zurück, ob alle Buchstaben aus dem Wort in der Liste enthalten sind.

```
1 checkGuessedWholeWord("GHSE", listOf('G', 'H', 'e', 'S', 'E'))
```

true

```
1 checkGuessedWholeWord("Informatik", listOf('I', 'a', 'y'))
```

false

Nutze showGuessedCharsAndBars

## 1.7 Aufgabe

Implementiere eine Funktion `showHangmanAsciiArt`. Dieser wird die Anzahl der übrigen Versuche übergeben. Sie gibt die entsprechende Zeichnung des Spielers und des Galgens zurück. Bei 6 übrigen Versuchen ist der Spieler noch nicht zu sehen. Bei 0 Versuchen ist der Spieler komplett zu sehen.

```
1 showHangmanAsciiArt(6)
```

```
+---+
|   |
|   |
|   |
|   |
=====
```

```
1 showHangmanAsciiArt(5)
```

```
+---+
|   |
0   |
|   |
|   |
=====
```

```
1 showHangmanAsciiArt(1)
```

```
+---+
|   |
0   |
/|\  |
/    |
=====
```

```
1 showHangmanAsciiArt(0)
```

```
+---+
|   |
0   |
/|\  |
/\   |
    |
=====
```

## 1.8 Aufgabe

Implementiere eine Funktion `showHangmanState`. Dieser werden das Wort, das erraten werden soll, die Liste der erratenen Buchstaben und die Anzahl der übrigen Versuche übergeben. Sie gibt einen String zurück, in dem der gesamte Zustand des Spiels inklusive Ascii-Art dargestellt ist.

```
1 showHangmanState("GHSE", listOf('G', 'H', 'e', 'S', 'E'), 3)
```

```
GHSE
3 Versuche übrig
+---+
|   |
0   |
/|  |
    |
=====
```

```
1 showHangmanState("blabla", listOf('b', 'l', 'S'), 1)
```

```
bl-bl-
1 Versuch übrig
+---+
|   |
0   |
/|\  |
/    |
    |
=====
```

Nutze `showHangmanAsciiArt`, `showGuessedCharsAndBars` und `showTriesLeft`! Erzeuge mit `"\n"` einen Zeilenumbruch.

## 1.9 Aufgabe

Implementiere eine Funktion `showGameResult`. Dieser werden das Wort, das erraten werden soll und die Liste der erratenen Buchstaben übergeben. Sie gibt einen String zurück. Wenn das gesamte Wort erraten wurde, steht in dem String, dass das Spiel gewonnen wurde. Ansonsten steht in dem String, dass das Spiel verloren wurde.

```
1 showGameResult("GHSE", listOf('G', 'H', 'x', 'S', 'E'))
```

Du hast gewonnen!

```
1 showGameResult("blabla", listOf('b', 'a', 'S', 'E'))
```

Du hast verloren!

Nutze `guessedWholeWord`!

## 1.10 Aufgabe

Implementiere eine Funktion `showHangmanStateAndGameResult`. Dieser werden das Wort, das erraten werden soll, die Liste der erratenen Buchstaben und die Anzahl der übrigen Versuche übergeben. Sie gibt einen String zurück, in dem der Zustand des Spiels und das Ergebnis des Spiels stehen.

```
1 showHangmanStateAndGameResult("GHSE", listOf('G', 'H', 'e', 'S', 'E'), 1)
```

GHSE

1 Versuch übrig

```
+---+
|   |
0   |
/|\  |
/    |
     |
=====
```

Du hast gewonnen!

```
1 showHangmanStateAndGameResult("blabla", listOf('b', 'l', 'E'), 0)
```

bl-bl-

0 Versuche übrig

```
+---+
|   |
0   |
     |
```

```

/|\ |
/\  |
   |
=====

```

Du hast verloren!

Nutze showHangmanState und showGameResult!

### 1.11 Aufgabe

Implementiere eine Funktion gameFinished. Dieser werden das Wort, das erraten werden soll, die Liste der erratenen Buchstaben und die Anzahl der übrigen Versuche übergeben. Sie gibt zurück, ob das Spiel zu Ende ist.

```
1 gameFinished("GHSE", listOf('G', 'H'), 1)
```

```
false
```

```
1 gameFinished("GHSE", listOf('G', 'H', 'e', 'S', 'E'), 1)
```

```
true
```

```
1 gameFinished("blabla", listOf('b', 'l', 'E'), 0)
```

```
true
```

Nutze checkGuessedWholeWord! Das Spiel ist zu Ende, wenn das Wort erraten wurde oder der Spieler keine Versuche mehr übrig hat.

### 1.12 Aufgabe

Definiere zwei globale Variablen mit den folgenden Inhalten:

```
1 questionWordToGuess
```

```
Welches Wort soll erraten werden?
```

```
1 questionGuessChar
```

```
Rate einen Buchstaben!
```

### 1.13 Aufgabe

Implementiere eine Funktion askChar. Diese fragt die Benutzerin, welchen Buchstaben sie rät. Der eingegebene Buchstabe wird anschließend als **Char** zurückgegeben.

## 2 Fertigstellung des Spiels

Ein Durchlauf des Spiels kann folgendermaßen aussehen.

```
1  Welches Wort soll erraten werden?
2  GHSE
3  ----
4  6 Versuche übrig
5      +----+
6      |    |
7          |
8          |
9          |
10         |
11 =====
12 Rate einen Buchstaben!
13 G
14 G---
15 6 Versuche übrig
16     +----+
17     |    |
18         |
19         |
20         |
21         |
22 =====
23 Rate einen Buchstaben!
24 y
25 G---
26 5 Versuche übrig
27     +----+
28     |    |
29     0    |
30         |
31         |
32         |
33 =====
34 Rate einen Buchstaben!
35 H
36 GH--
37 5 Versuche übrig
38     +----+
39     |    |
40     0    |
```

```

41         |
42         |
43         |
44     =====
45     Rate einen Buchstaben!
46     S
47     GHS-
48     5 Versuche übrig
49     +---+
50     |   |
51     0   |
52         |
53         |
54         |
55     =====
56     Rate einen Buchstaben!
57     E
58     GHSE
59     5 Versuche übrig
60     +---+
61     |   |
62     0   |
63         |
64         |
65         |
66     =====
67     GHSE
68     5 Versuche übrig
69     +---+
70     |   |
71     0   |
72         |
73         |
74         |
75     =====
76     Du hast gewonnen!

```

## 2.1 Aufgabe

Implementiere eine Funktion `playHangmanHelper`. Dieser wird ein Wort als String übergeben. Dieses muss anschließend erraten werden. Ein Aufruf der Funktion ist oben ab Zeile 3 zu sehen.

Nutze `showHangmanState`, `gameFinished`, `askChar`, `compTriesLeft` und `showHangmanStateAndGameResult`.



## 2.2 Aufgabe

Implementiere eine Funktion `askWordToGuess`. Diese fragt die Benutzerin, welches Wort erraten werden soll. Das eingegebene Wort wird anschließend zurückgegeben.

Nutze die Variable `questionWordToGuess`!

## 2.3 Aufgabe

Implementiere eine Funktion `playHangman`. Diese hat keine Parameter. Sie fragt zunächst, welches Wort erraten werden soll. Anschließend wird mit diesem Wort eine Runde Galgenmännchen gespielt.

## 2.4 Aufgabe

Überspringe mit `for`-Schleifen, dem Kontrollzeichen `"\n"` und `println` Zeilen um das Spiel spielbar zu machen!

### 3 Anhang

```
listOf("""
+---+
|   |
0   |
/|\  |
/\   |
|   |

===== "", ""

+---+
|   |
0   |
/|\  |
/   |
|   |

===== "", ""

+---+
|   |
0   |
/|\  |
|   |

===== "", ""

+---+
|   |
0   |
/|   |
|   |

===== "", ""

+---+
|   |
0   |
|   |

===== "", ""

+---+
|   |
0   |
|   |
```

