

Kopfrechen-Training

Die Idee ist es, ein Programm zu schreiben, das in einem Zahlenraum (z. B. von 1 bis 30)

Kopfrechenaufgaben stellt und die Benutzereingabe auf ihre Richtigkeit hin überprüft.

Am Ende soll dem Benutzer eine Art Statistik ausgegeben werden, wie oft er richtig und wie oft er falsch gerechnet hat. Das Grundprogramm lässt sich beliebig erweitern. In der Konsole:

```
1  Kopfrechenaufgaben
2
3  Eingabe e zum Beenden!
4
5  Was ergibt: 20 * 2?
6  Ihre Eingabe: 40
7  Das ist richtig!
8
9  Was ergibt: 13 * 4?
10 Ihre Eingabe: 53
11 Falsch (Ergebnis: 52)!
12
13 Was ergibt: 17 * 15?
14 Ihre Eingabe: e
15 Richtige Antwort(en): 1
16 Falsche Antwort(en): 1
```

In den Zeilen 1 bis 3 wird der Anwender über das Programm (Titel und Abbruch des Programms) informiert. Die Zeilen 5 bis 7, 9 bis 11 und 13 bis 14 zeigen dann Beispiel-Aufgaben. Bei der letzten Aufgabe hat sich der Benutzer entschlossen, das Programm durch Eingabe eines „e“ zu Beenden. In den Zeilen 15 bis 16 erhält er dann eine Übersicht, wie oft er korrekt und wie oft er falsch geantwortet hat.

Im Programm werden wir die Funktion `randint()` aus der Bibliothek `random` (Zufall) nutzen, über die wir Zufallszahlen generieren können.

```
1  import random
2
3  zahl = random.randint(1, 20)
```

In Zeile 1 wird die Bibliothek importiert und ihre Funktionen können nun in unserem Programm genutzt werden. In Zeile 3 rufen wir die Funktion `randint()` mit den Werten 1 und 20 aus dem Modul auf. Die Funktion erzeugt jetzt eine ganzzahlige Zufallszahl

zwischen 1 und 20, also 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 oder 20 und speichert die Zufallszahl in der Variablen zahl.

Das Grundgerüst unseres Programms

Das Grundgerüst unseres Programms kann wie folgt aussehen:

```
1  import random
2
3  # Einstellungen
4  untere_grenze = 1
5  obere_grenze = 20
6
7  # Variablen
8  richtige_antworten = 0
9  falsche_antworten = 0
10
11 # Wir informieren den Benutzer über das Programm
12 print("Kopfrechenaufgaben")
13 print() # Leerzeile
14 print("Eingabe e zum Beenden!")
15 print() # Leerzeile
16
17 # Hauptschleife
18 while True:
19
20     zahl_1 = random.randint(untere_grenze, obere_grenze)
21     zahl_2 = 6 # Auch hier eine Zufallszahl generieren!
22
23     print("Was ergibt: " + str(zahl_1) + " * " + str(zahl_2) + "?")
24
25     eingabe = input("Ihre Eingabe: ")
26
27     if eingabe == "e":
28         break
29     else:
30         benutzer_ergebnis = int(eingabe)
31
32         if benutzer_ergebnis == zahl_1 * zahl_2:
33             print("Das ist richtig!")
34             # Die Zählvariable für korrekte Antworten muss erhöht werden
35         else:
36             print("Falsch! Richtig: " + str(zahl_1 * zahl_2) + "!")
37             # Die Zählvariable für falsche Antworten muss erhöht werden
38         print()
39
40 # Ausgabe der Statistik
41 print("Richtige Antwort(en): " + str(richtige_antworten))
42 print("Falsche Antwort(en): " + str(falsche_antworten))
```

In Zeile 1 importieren wir das benötigte Modul `random`, das uns Funktionalität bereitstellt, um Zufallszahlen zu erhalten. In den Zeilen 4 bis 5 können Einstellungen vorgenommen werden. Hier legen wir fest, aus welchem Zahlenbereich die Werte stammen sollen.

In den Zeilen 12 bis 15 geben wir grundlegende Informationen zum Spiel aus. Eine `print()`-Anweisung ohne Inhalt bewirkt eine leere Zeile.

Unser Hauptprogramm steckt in den Zeilen 18 bis 38 und startet mit einer Endlosschleife, die wir in Zeile 18 beginnen. Mit den Zeilen 20 und 21 legen wir für jeden Schleifendurchgang zwei neue Zahlen fest und geben die Frage nach dem Produkt dieser beiden Zahlen in Zeile 23 aus.

Der Anwender wird in Zeile 25 zur Eingabe aufgefordert. Das Ergebnis speichern wir zunächst in der Variable `eingabe` (Zeile 25), weil wir in den Zeilen 27 bis 30 prüfen müssen, ob der Benutzer durch die Eingabe eines „e“ das Programm beenden möchte oder mit einer Zahl ein Ergebnis angeben möchte.

Mit den Programmzeilen 32 bis 37 prüfen wir die Eingabe des Benutzers und geben ihm ein Feedback, ob sein Rechenergebnis richtig oder falsch ist. Hier müssten die Zählvariablen `richtige_antworten` und `falsche_antworten` hochgezählt werden.

Hat der Benutzer ein „e“ eingegeben, wird die Schleife in Zeile 28 mit der `break`-Anweisung verlassen und die letzten beiden Code-Zeilen kommen zur Ausführung. Dies bewirkt, dass der Benutzer über seine erfolgreichen und erfolglosen Versuche informiert wird.

Deine Aufgaben

Folgende Aufgaben sind noch zu bearbeiten, um ein einsatzbereites Programm zu entwickeln:

1. Passe die Zeile 21 dahingehend an, dass auch hier eine Zufallsvariable verwendet wird. Du kannst Dich an der vorherigen Codezeile orientieren.
2. Bei der Auswertung der Ergebnisse müssen die Zählvariablen ergänzt werden, so dass am Ende eine sinnvolle Statistik ausgegeben werden kann.
3. Gibt der Benutzer ein „e“ ein, sollte in der Konsole wenigstens eine Leerzeile erscheinen, so dass die Statistik nicht direkt unter der Eingabe erscheint.
4. Nach Möglichkeit sollte der Benutzer gefragt werden, in welchem Zahlenraum er Kopfrechen-Aufgaben bearbeiten möchte (Zeilen 4 und 5).

Als Zusatzaufgabe ist folgende Aufgabe gedacht:

5. Zusatzaufgabe: Wie kann der Quelltext ergänzt werden, so dass neben der Multiplikation auch die Division mit ganzzahligem Ergebnis geübt werden kann?

Als neuen Befehl wirst Du hier `continue` lernen, das Du wie ein `break` verwendest.

Während `break` die Schleife komplett verlässt, bewirkt `continue`, dass ein Schleifendurchgang verlassen und ein neuer Schleifendurchgang gestartet wird.

Tipp: Du musst für ein Zahlenpaar `zahl_1` und `zahl_2` prüfen, ob sich die erste Zahl ohne Rest durch die zweite Zahl teilen lässt. Falls ja, kannst Du die beiden Zahlen dem Anwender nennen, falls nein, solltest Du diesen Schleifendurchgang verlassen und ein neues Paar Zahlen versuchen.