

Aufgaben (Loops/Schleifen)

Allgemeine Hinweise

- Jede Aufgabe soll in einer eigenen Java-Datei bearbeitet werden.
- Jede Datei enthält eine `public static void main(String[] args)` -Methode.

Aufgabe 1: FizzBuzz

Schreibe ein Programm das alle Zahlen von 1 bis 100 ausgibt. Wenn die Zahl allerdings ein Vielfaches von 3 ist, soll statt der Zahl das Wort "Fizz" ausgegeben werden. Wenn die Zahl ein Vielfaches von 5 ist, soll statt der Zahl das Wort "Buzz" ausgegeben werden. Ist die Zahl sowohl ein Vielfaches von 3 als auch von 5, soll statt der Zahl das Wort "FizzBuzz" ausgegeben werden.

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
```

Aufgabe 2: Passwortabfrage

Übe die Verwendung von `while` - oder `do-while` -Schleifen und den Vergleich von Strings mit der Methode `equals()` .

Anforderungen:

1. Lege ein Passwort als `String` fest, z. B.:

```
String password = "java123";
```

2. Frage den Benutzer in einer Schleife nach dem Passwort.

3. Die Schleife soll so lange laufen, bis der Benutzer das richtige Passwort eingibt.
4. Nach jeder falschen Eingabe soll eine Fehlermeldung ausgegeben werden.
5. Bei richtiger Eingabe soll „Passwort korrekt!“ ausgegeben werden.

Achtung: Der Vergleich von Strings in Java funktioniert nicht mit dem `==` Operator

Erklärung zu `String.equals()` :

- In Java ist ein `String` ein **Objekt** und kein primitiver Datentyp wie `int` oder `double` .
- **Vergleich von Strings:**
 - `==` prüft bei primitiven Datentypen wie `int`, ob sie den gleichen Wert haben.
 - `==` prüft bei `String`-Variablen, ob sie auf dasselbe Objekt im Speicher zeigen.
 - `equals()` prüft bei Strings, ob die Inhalte (also die Zeichenfolge) der Strings identisch sind.

Beispielausgabe:

```
Bitte Passwort eingeben: test
Falsches Passwort, versuche es erneut.
Bitte Passwort eingeben: java123
Passwort korrekt!
```

Aufgabe 3: Zahlenraten (Guessing Game)

Der Benutzer soll eine zufällige Zahl zwischen 1 und 100 erraten.

Anforderungen:

1. Verwende die Klasse `Random` , um eine Zufallszahl zwischen **1 und 100** zu erzeugen.
2. In einer `while` -Schleife soll der Benutzer wiederholt eine Zahl eingeben.
3. Das Programm gibt je nach Eingabe aus:
 - „Zu groß“
 - „Zu klein“
 - „Richtig!“
4. Die Schleife endet erst, wenn die richtige Zahl erraten wurde.

Hinweis: Zufallszahl mit `Random` erzeugen

In Java kann eine Zufallszahl mit der Klasse `Random` erstellt werden:

```
import java.util.Random;

Random rand = new Random();
int zufallsZahl = rand.nextInt(101); // erzeugt 1 bis 100
```