



Aufgaben zum algorithmischen Problemlösen

Name:

Vorname:

Klasse:

Sequenzen wiederholen

1) Gegeben ist der folgende fehlerfreie Quelltext in Java.

```
9 public class Zahlenraten {
10     public static void main (String [] args) {
11
12         int eingabe = 0;
13         int geheimzahl = 0;
14         boolean weiter = true;
15
16         IO.show("Errate eine Zahl zwischen 1 und 100.");
17         geheimzahl = (int) (100 * Math.random() + 1);
18
19         while (eingabe != geheimzahl) {
20             eingabe = IO.getInt("Ihr Tipp: ");
21
22             if (eingabe < geheimzahl) {
23                 IO.show("Zahl zu klein");
24             }
25             else {
26                 if (eingabe > geheimzahl) {
27                     IO.show("Zahl zu groß");
28                 }
29                 else {
30                     IO.show("korrekt!");
31                 }
32             }
33         }
34     }
35 }
36 }
```

- Übertragen Sie den Quelltext in den JavaEditor. Speichern Sie ihn mit dem Namen Zahlenraten.java auf Ihrem Laufwerk H:\ ab. Testen Sie das Programm mehrfach. Beschreiben Sie die Arbeitsweise des Programms.
 - Beschreiben Sie den Ausdruck in Zeile 14 und 17. Nutzen Sie die Tafelwerksergänzungen.
 - Beschreiben Sie die Struktur, die in Zeile 19 beginnt. Nutzen Sie die Tafelwerksergänzungen. Geben Sie die Zeile an, auf der die Struktur endet.
 - Stellen Sie den Algorithmus als Struktogramm dar. Markieren Sie den Bereich der geschachtelten Auswahl.
 - Erweitern Sie den Algorithmus so, dass die Anzahl der Versuche gezählt/ausgegeben wird.
 - Erweitern Sie den Algorithmus so, dass eine Wiederholung des Spiels möglich wird.
- 2) Entwickeln Sie einen Algorithmus mit einer Schleife, der alle Quadratzahlen von 0 bis zum Quadrat einer einzugebenden Zahl ermittelt.



Aufgaben zum algorithmischen Problemlösen

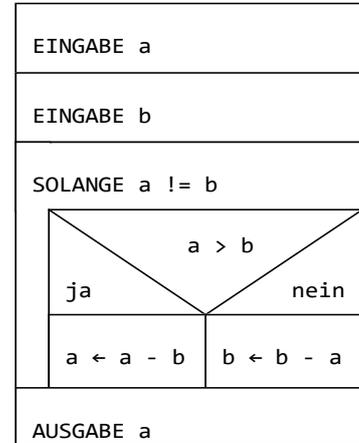
Name:

Vorname:

Klasse:

3) Für das Kürzen von Brüchen ist die Bestimmung des kleinsten gemeinsamen Vielfachen (kgV) oder des größten gemeinsamen Teilers (ggT) notwendig. Ein Algorithmus zum Berechnen des ggT zweier natürlicher Zahlen a und b liegt als Struktogramm vor.

- Nennen Sie die Aufgabe einer Variablen und deren Datentyp.
- Geben Sie die im Struktogramm verwendete Variablen sowie dafür geeignete Datentyp an.
- Ordnen Sie den Elementen des dargestellten Struktogramms die verwendeten algorithmischen Grundstrukturen zu.
- Implementieren Sie Algorithmus im JavaEditor unter dem Namen KGV.java und testen Sie das Programm.
- Prüfen Sie die Arbeitsweise des Programms für negative Zahlen. Ändern Sie das Programm zweckmäßig ab.
- Erweitern Sie das Programm so, dass zusätzlich das kgV der beiden Zahlen ausgegeben wird.
Es gilt $a \cdot b = \text{kgV}(a, b) \cdot \text{ggT}(a, b)$.



4) Das Struktogramm zeigt einen Algorithmus, der prüft, ob eine Zahl n eine Primzahl ist oder nicht.

- Begründen Sie die Wahl des Datentyps für die Variable n .
- Kennzeichnen Sie im Struktogramm Wertzuweisungen sowie Wahrheitswerte.
- Implementieren Sie das Struktogramm im Java-Editor unter dem Namen IsPrime.java und testen Sie die korrekte Arbeitsweise des Algorithmus an den Zahlen 5, 20 und 131.
- Beim Implementieren können sich Fehler einschleichen. Erstellen Sie eine Übersicht über die Fehlerarten Syntaxfehler, logischer Fehler und Laufzeitfehler sowie deren Auffinden und Beheben.

