



Imperatives Problemlösen in Java (Turtlerechner)

Name:

Vorname:

Klasse:

Vorwort

Die Übungen dienen der Wiederholung und des Ausbaus von Programmierkonzepten. Die Aufgaben sind in *Theorie* (T) und *Praxis* (P) geteilt.

Berechnung des größten gemeinsamen Teiles zweier Zahlen

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 Java-Quelltext und als Struktogramm in den Abbildungen 1 und 2 vor.

```
28 public void berechneGGT(int a, int b){
29     while(a != b){
30         if (a > b){
31             a = a - b;
32         }
33         else{
34             b = b - a;
35         }
36     }
37     joe.write("Der ggT ist "+a+".");
38 }
39 }
```

Abb. 1: Java-Quelltext ggT

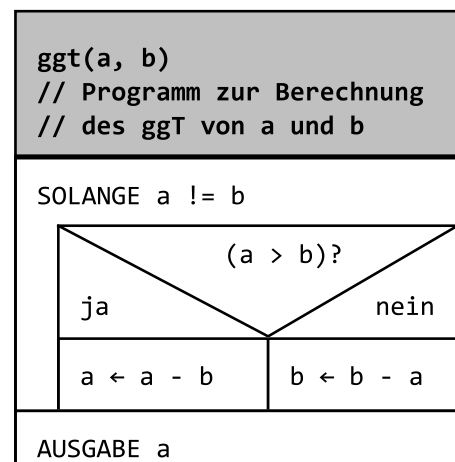


Abb. 2: Struktogramm ggT

- 1) Geben Sie die verwendeten Parameter sowie deren Datentyp an.
- 2) Nennen Sie allgemein die Aufgabe von Variable, Parameter und Datentyp.
- 3) Ordnen Sie den Elementen des Struktogramms in Abb. 2 die algorithmischen Strukturen Anweisung, Schleife und Verzweigung zu.
- 4) Ermitteln Sie im Quelltext die Realisierung von Ein- und Ausgaben.
- 5) Bestimmen Sie händisch mithilfe des Algorithmus den ggT von 2 und 6.
Notieren Sie die Variablenbelegung während des Ablaufs.
- 6) Implementieren Sie den Quelltext in Abb. 1 in der Turtle-Klasse und testen Sie die Methode.
Dokumentieren Sie die Tests.
- 7) Erweitern Sie das Programm so, dass zusätzlich das kleinste gemeinsame Vielfache (kgV) der beiden Zahlen ausgegeben wird. Es gilt $a \cdot b = \text{kgV}(a, b) \cdot \text{ggT}(a, b)$.
- 8) Prüfen Sie die Arbeitsweise des Programms für negative Zahlen und für die Zahl Null.
Ändern Sie den Algorithmus zweckmäßig ab.



Imperatives Problemlösen in Java (Turtlerechner)

Name:

Vorname:

Klasse:

Berechnung des Osterfests

CARL FRIEDRICH GAUß (1777 – 1855) entwickelte einen Algorithmus zur Berechnung des Ostersonntags in Abhängigkeit vom Jahr. Der Feiertag wird am ersten Sonntag nach dem ersten Frühlingsvollmond begangen. Damit ist der 22. März der früheste, der 25. April der spätmögliche Termin, auf den Ostern fallen kann. Vom Osterfest hängen auch die Feste Christi Himmelfahrt (40 Tage nach Ostern) und Pfingsten (50 Tage nach Ostern) ab.

- 1) Bestimmen Sie einen geeigneten

Datentyp für *jahr*.

- 2) Erläutern Sie die Operatoren MOD

und DIV an einem selbstgewählten

Beispiel. Geben Sie die

Operationszeichen in Java an.

- 3) Ermitteln Sie die Umsetzung des

Datentyps ZEICHENKETTE in

Java. Geben Sie ein Beispiel an.

- 4) Implementieren Sie das

Struktogramm als Methode in der

Turtle-Klasse und testen Sie die

korrekte Arbeitsweise des

Algorithmus. Realisieren Sie die

Eingabe über Parameter.

Dokumentieren Sie die Tests.

- 5) Der Algorithmus gilt in dieser Form

nur für den gregorianischen

Kalender, den Papst GREGOR XIII

im Jahr 1582 in Kraft setzte.

Ändern Sie den Algorithmus so ab,

dass für Jahreseingaben vor 1582

keine Berechnung erfolgt und

stattdessen ein Hinweis erscheint.

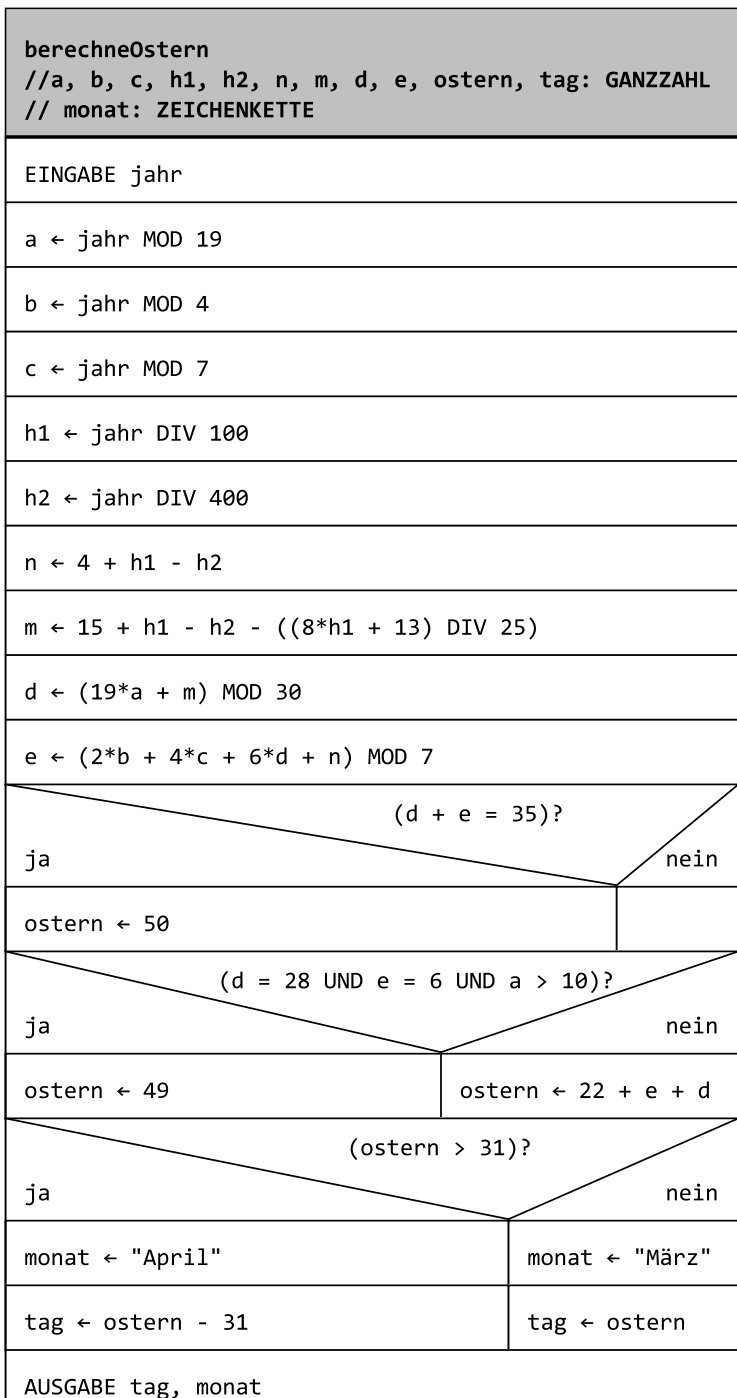
- 6) Erweitern Sie den Algorithmus um

eine Aussage darüber, ob das

eingeebene Jahr einen Schalttag

hat(te) oder nicht. Recherchieren

Sie dazu die **korrekte** Schaltjahrregel.





Imperatives Problemlösen in Java (Turtlerechner)

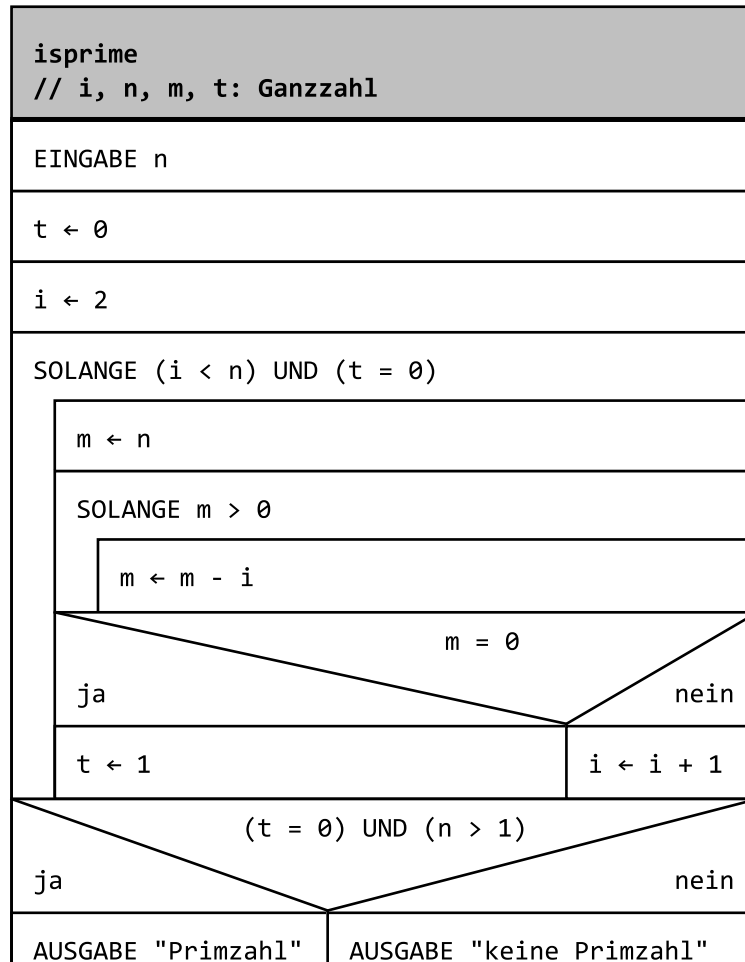
Name:

Vorname:

Klasse:

Primzahltester

Modernen Verschlüsselungsverfahren nutzen Primzahlen. Das Struktogramm veranschaulicht einen Algorithmus, der prüft, ob eine Zahl n eine Primzahl ist oder nicht.



- 1) Markieren und beschriften Sie im Struktogramm unterschiedlich:
 - a) Wertzuweisungen
 - b) boolesche Ausdrücke
 - c) Verzweigungen
 - d) Schleifen
- 2) Implementieren Sie das Struktogramm als Methode in der Turtle-Klasse. Realisieren Sie die Eingabe über einen Parameter. Testen Sie die korrekte Arbeitsweise des Algorithmus an den Zahlen 5, 20 und 131. Dokumentieren Sie die Tests.
- 3) Ändern Sie die Methode so ab, dass statt der Ausgabe durch die Turtle ein Wahrheitswert zurückgegeben wird (Auftrag → Anfrage). Recherchieren Sie dafür ggf. die Umsetzung von Methoden mit einer Rückgabe in Java.



Imperatives Problemlösen in Java (Turtlerechner)

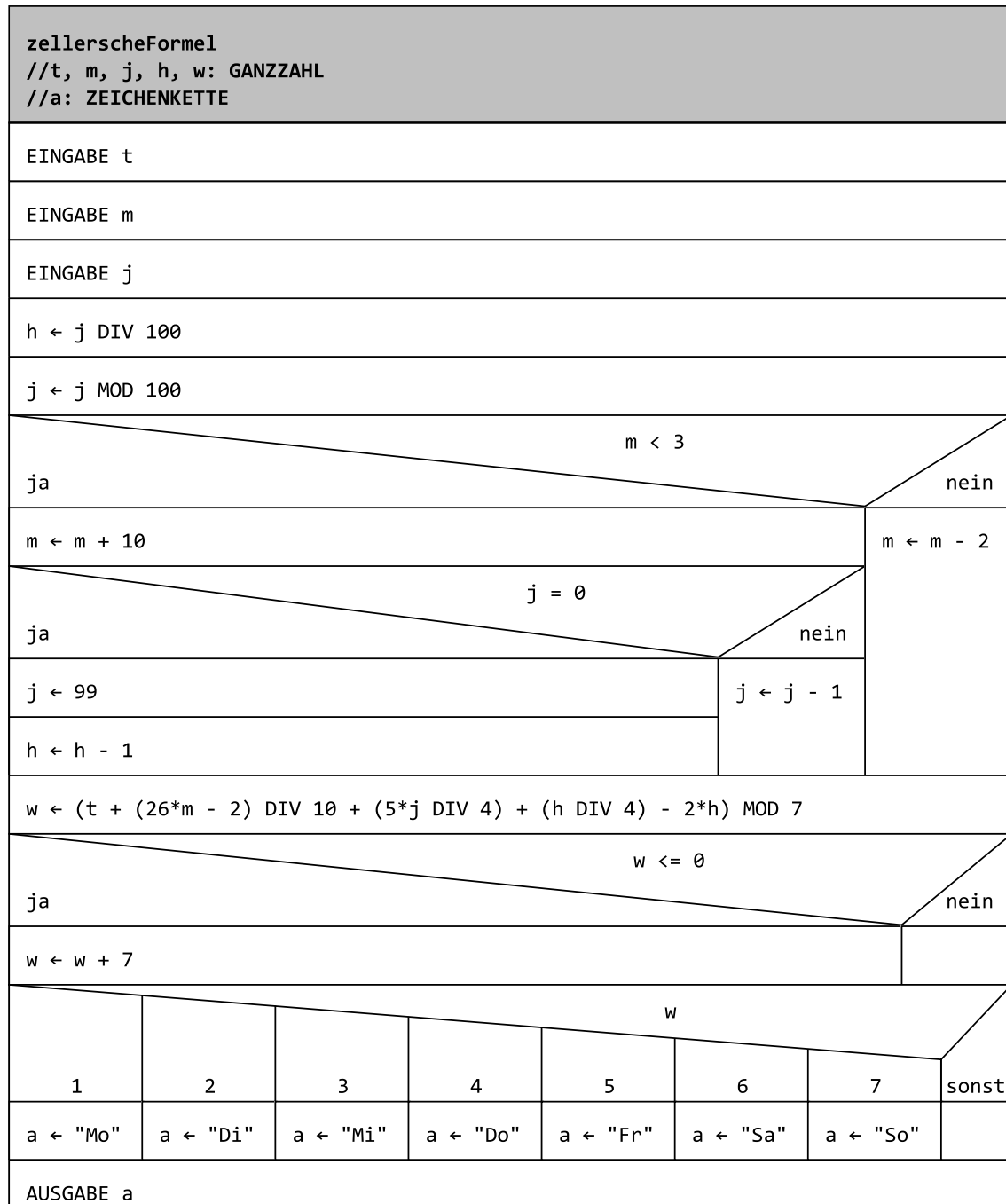
Name:

Vorname:

Klasse:

ZELLERSche Formel

Die Gesetzmäßigkeiten des immerwährenden Kalenders wurden von CHRISTOPH ZELLER 1885 zu einer mathematischen Formel zusammengefasst. Der darauf basierende Algorithmus bestimmen für jedes Datum bestehend aus Tag, Monat und Jahr den Wochentag.



- 1) Kennzeichnen Sie im Struktogramm die drei Arten für Auswahlstrukturen unterschiedlich.
- 2) Implementieren Sie das Struktogramm als Anfragemethode in der Turtle-Klasse und testen Sie die korrekte Arbeitsweise. Dokumentieren Sie die Tests.
- 3) Beim Implementieren können sich Fehler einschleichen.
Erstellen Sie eine Übersicht über die Arten Syntaxfehler, logischer Fehler und Laufzeitfehler mit Beispiel sowie deren Auffinden und Beheben.



Imperatives Problemlösen in Java (Turtlerechner)

Name:

Vorname:

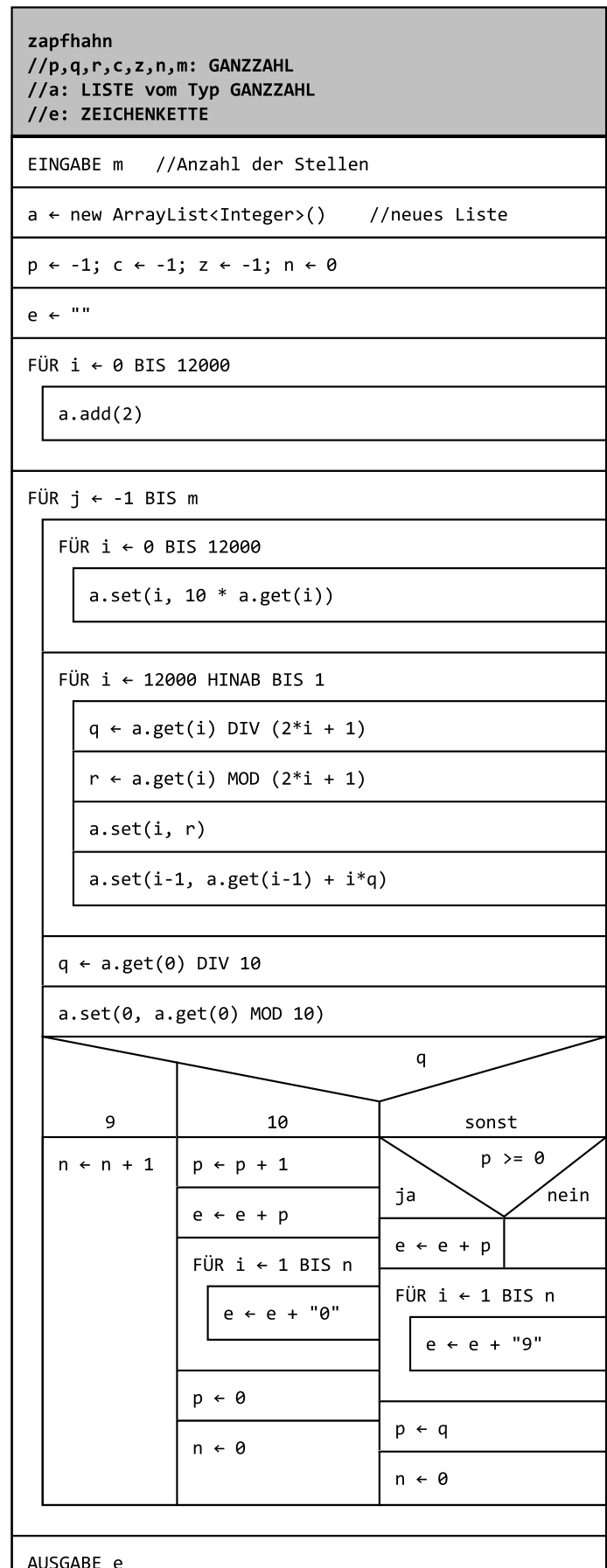
Klasse:

Zahl π

Die Zahl π ist eine der faszinierendsten Zahlen der Mathematik. Die Ziffernfolge lässt keine Logik erkennen, daher wird sie in der Informatik zur Bestimmung von Pseudozufallszahlen eingesetzt.

Eine Möglichkeit der Berechnung von π bietet der sog. Tröpfel-Algorithmus von RABINOWITZ und WAGON (1995), den das Struktogramm zeigt.

- 1) Beschreiben Sie das Prinzip und die Java-Umsetzung einer Zählschleife.
- 2) Ordnen Sie den zählenden Schleifen die Begriffe auf- bzw. absteigende Schleife zu.
- 3) Informieren Sie sich über das Prinzip „Liste“ als Datenstruktur. Beschreiben Sie die Umsetzung in Java.
- 4) Implementieren Sie das Struktogramm und testen Sie die korrekte Arbeitsweise des Algorithmus für verschiedene Eingaben.





Imperatives Problemlösen in Java (Turtlerechner)

Name:

Vorname:

Klasse:

Primfaktorzerlegung

Eine Einwegfunktion ist eine Funktion, die in die eine Richtung leicht und in die andere Richtung sehr schwer zu berechnen ist. Beispielsweise lassen sich Primzahlen leicht zu einem Produkt multiplizieren, jedoch lässt sich eine große Zahl nur sehr aufwändig in Primfaktoren zerlegen. Das Struktogramm zeigt einen Algorithmus, der eine Zahl n in seine Primfaktoren zerlegt.

1) Markieren Sie im Struktogramm

Verneinungen und zusammengesetzte

Bedingungen unterschiedlich.

Beschreiben Sie deren Umsetzung in Java.

2) Ermitteln Sie den Typ der Variablen gefunden und die Verwendung in Java.

3) Bestimmen Sie den Befehl, der den Wert von p zur Liste der Faktoren anfügt. Nutzen Sie die Tafelwerksseiten.

4) Implementieren Sie das Struktogramm und testen Sie die korrekte Arbeitsweise des Algorithmus für verschiedene Eingaben.

5) Ändern Sie das Programm so, dass die Ausgabe der Primfaktoren als Gleichung erfolgt, also z. B. $12 = 2 * 2 * 3$ statt aktuell nur $[2, 2, 3]$. Nutzen Sie dazu eine Listenschleife.

```
ermittlePrimfaktoren
//i, n, p, z: GANZZAHL
//gefunden: ?
//faktoren: ArrayList<Integer>
```

EINGABE n

$p \leftarrow 0$

faktoren $\leftarrow$ new ArrayList<>()

$z \leftarrow n$

while ($z > 1$)

$i \leftarrow 2$

gefunden $\leftarrow$ false

while ($i*i \leq z$ AND NOT gefunden)

$z \text{ MOD } i = 0$

ja

nein

gefunden $\leftarrow$ true

i inkrementieren

$p \leftarrow i$

NOT gefunden

ja

nein

$p \leftarrow z$

p zu faktoren hinzufügen

$z \leftarrow z \text{ DIV } p$

joe.write(faktoren.toString())