



Imperatives Problemlösen in Java

Name:

Vorname:

Klasse:

Hinweise

Der jahrgangsübergreifende Hauptfachkurs Informatik 11/12 setzt Elemente des Problemlösens in der Programmiersprache Java aus der Klasse 10 voraus. Zum Angleichen der Voraussetzungen mit denen Ihren Mitschülern aus der Jahrgangsstufe 12 sind einige Grundlagen aufzuarbeiten. Dazu dienen die folgenden Übungen.

Als Ergebnis gestalten Sie ein Portfolio mit folgenden Elementen:

- vom Lehrer erhaltene Tafelwerkergänzungen
- vom Lehrer erhaltene Aufgaben und von Ihnen entwickelte Lösungen (Beschreibungen, ggf. Struktogramme, Quelltextausdrucke, ...)
- handschriftliches Glossar über die zu recherchierenden Fachbegriffe

Das Portfolio wird bewertet.



Imperatives Problemlösen in Java

Name:

Vorname:

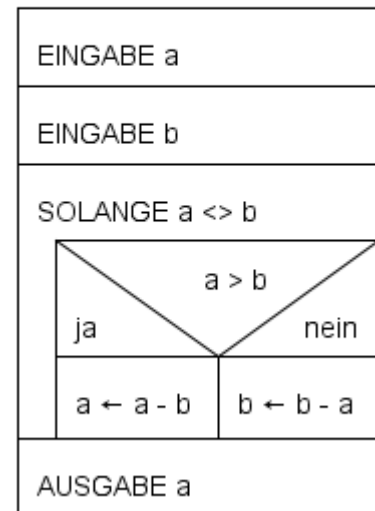
Klasse:

Berechnung des größten gemeinsamen Teiles zweier Zahlen

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

```
1 /**
2  * Programm zur Berechnung des ggT
3  * @version 1.0 from 25.01.2011
4  * @author T. Hempel
5  */
6
7 public class ggT {
8     public static void main(String[] args) {
9         int a = IO.getInt("Geben Sie die Zahl a ein.");
10        int b = IO.getInt("Geben Sie die Zahl b ein.");
11        while (a != b) {
12            if (a > b) {
13                a = a - b;
14            }
15            else {
16                b = b - a;
17            }
18        }
19        IO.show("Der ggT ist "+a+".");
20    }
21 }
```

ggT
// Programm zur
// Berechnung ggT



- 1) Beschreiben Sie den Begriff **Algorithmus**.
- 2) Begründen Sie die Umsetzung des **EVA-Prinzips** im gegebenen Projekt.
- 3) Geben Sie die verwendeten **Variablen** sowie deren **Datentyp** an.
- 4) Informieren Sie sich auf der Tafelwerksergänzung „Algorithmische Grundstrukturen“ über die grundlegenden algorithmischen Strukturen.
- 5) Ordnen Sie den **algorithmischen Grundstrukturen** aus dem Struktogramm die Quelltextelemente durch Einzeichnen von Pfeilen zu.
- 6) Implementieren Sie den Quelltext im JavaEditor und testen Sie das Programm.
- 7) Erweitern Sie das Programm so, dass das kgV der beiden Zahlen ebenfalls ausgegeben wird. Zur Erinnerung: Das Produkt von kgV und ggT der Zahlen a und b entspricht dem Produkt von a und b.
- 8) Prüfen Sie die Arbeitsweise des Programms für negative Zahlen. Passen Sie das Programm ggf. an.



Imperatives Problemlösen in Java

Name:

Vorname:

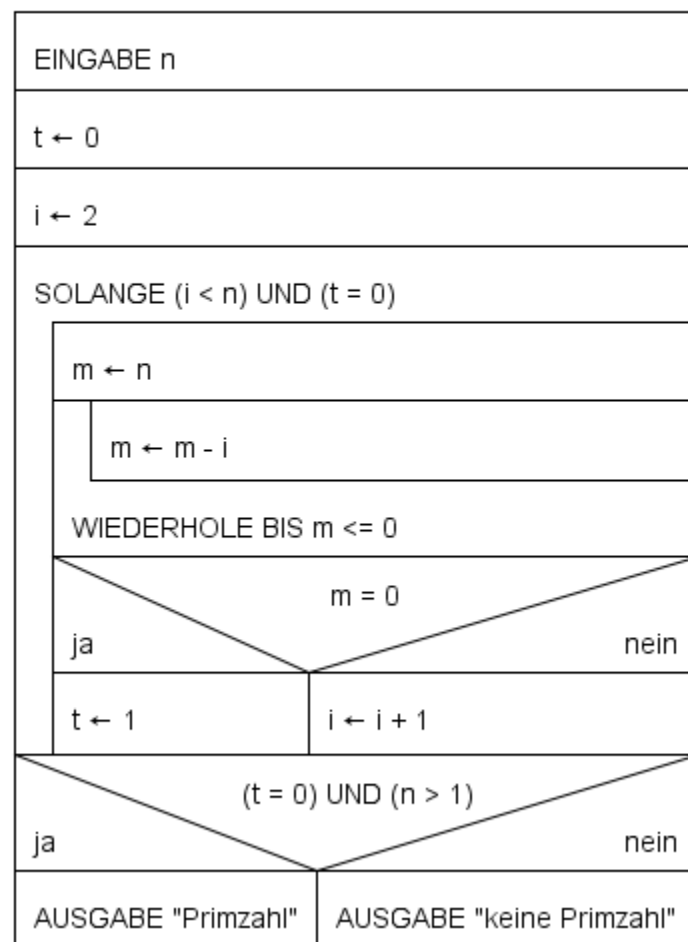
Klasse:

Primzahltester

In modernen Verschlüsselungsverfahren spielen Primzahlen eine wichtige Rolle. Das Struktogramm veranschaulicht einen Algorithmus der prüft, ob eine Zahl n eine Primzahl ist oder nicht.

isprime

// i, n, m, t : Ganzzahl



- 1) Informieren Sie sich auf der Tafelwerksergänzung „Datentypen“ über Ganzzahl-Typen.
- 2) Wählen Sie für die Variablen einen geeigneten Typ aus. Begründen Sie.
- 3) Informieren Sie sich auf der Tafelwerksergänzung „Operatoren“ über **boolesche Ausdrücke** und **Wertzuweisungen**.
 - a) Vergleichen Sie Wertzuweisung in Java und im Struktogramm.
 - b) Vergleichen Sie Gleichheitsvergleich in Java und im Struktogramm.
 - c) Markieren Sie im Struktogramm Wertzuweisungen und boolesche Operationen unterschiedlich.
- 4) Erläutern Sie den Unterschied zwischen einer kopf- und einer fußgesteuerten **Schleife** am Beispiel.
- 5) Beschreiben Sie die Besonderheit bei der Umsetzung einer fußgesteuerten Schleife in **Java**.
- 6) Implementieren Sie das Struktogramm und testen Sie die korrekte Arbeitsweise des Algorithmus an den Zahlen 5, 20 und 131.



Imperatives Problemlösen in Java

Name:

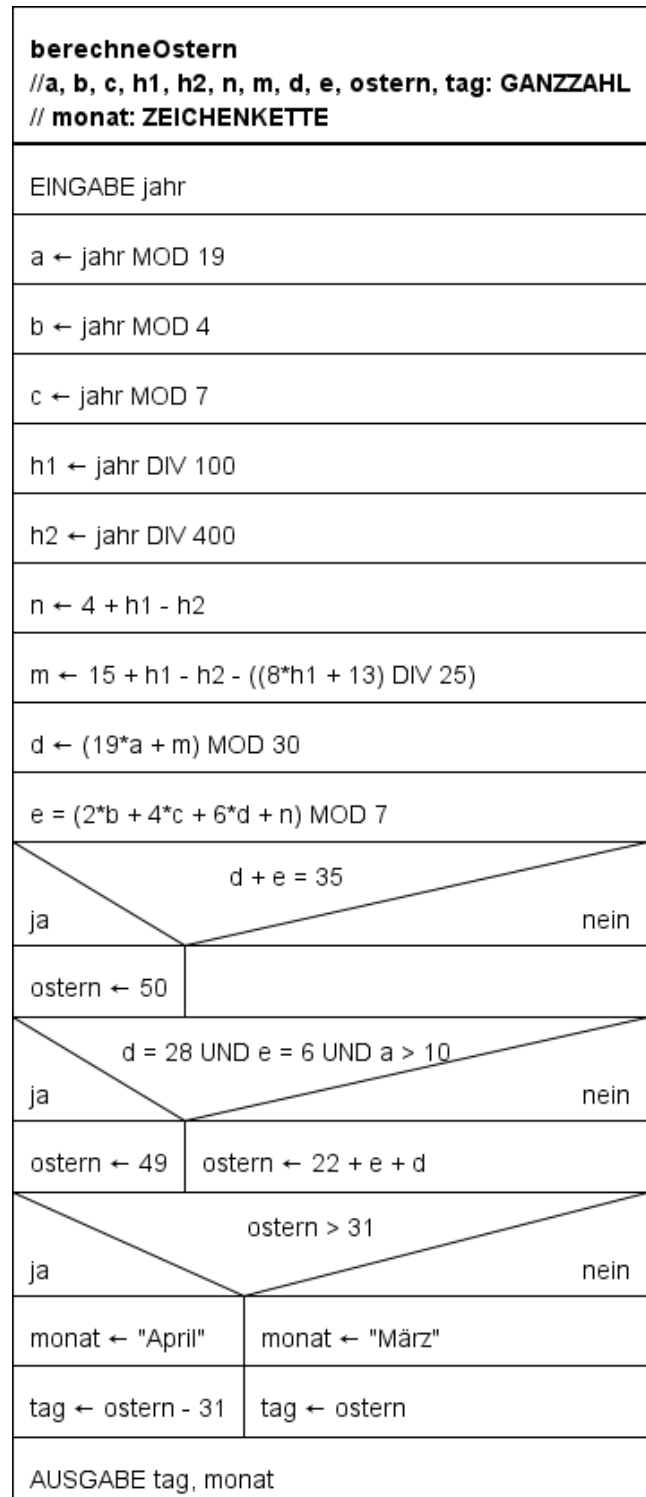
Vorname:

Klasse:

Berechnung des Osterfests

CARL FRIEDRICH GAUß (1777 – 1855) entwickelte einen Algorithmus zur Berechnung des Datums des Osterfestes. Dieses wird am ersten Sonntag nach dem ersten Frühlingsvollmond gefeiert. Damit ist der 22. März der früheste, der 25. April der letztmögliche Termin, auf den Ostern fallen kann. Von diesem Termin hängen auch die Feste Christi Himmelfahrt (40 Tage nach Ostern) und Pfingsten (50 Tage nach Ostern) ab.

- 1) Bestimmen Sie einen geeigneten Datentyp für die Variable `jahr`. Begründen Sie.
- 2) Informieren Sie sich auf der Tafelwerksergänzung „Operatoren“ über **arithmetische Operatoren**.
- 3) Erläutern Sie die Operationen MOD und DIV an einem selbstgewählten Beispiel.
- 4) Erläutern Sie den Unterschied zwischen einer ein- und zweiseitigen **Auswahl** am Beispiel.
- 5) Markieren Sie im Struktogramm die beiden Verzeigungsarten unterschiedlich.
- 6) Recherchieren Sie den **Datentyp Zeichenkette** und seine Umsetzung in Java.
- 7) Implementieren Sie das Struktogramm und testen Sie die korrekte Arbeitsweise des Algorithmus.
- 8) 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 ein Hinweis erscheint.
- 9) Erweitern Sie den Algorithmus um eine Aussage darüber, ob das eingegebene Jahr einen Schalttag hat(te) oder nicht. Recherchieren Sie dazu die korrekte Schaltjahrregel. Stellen Sie die Schaltjahrregel als Struktogramm dar.





Imperatives Problemlösen in Java

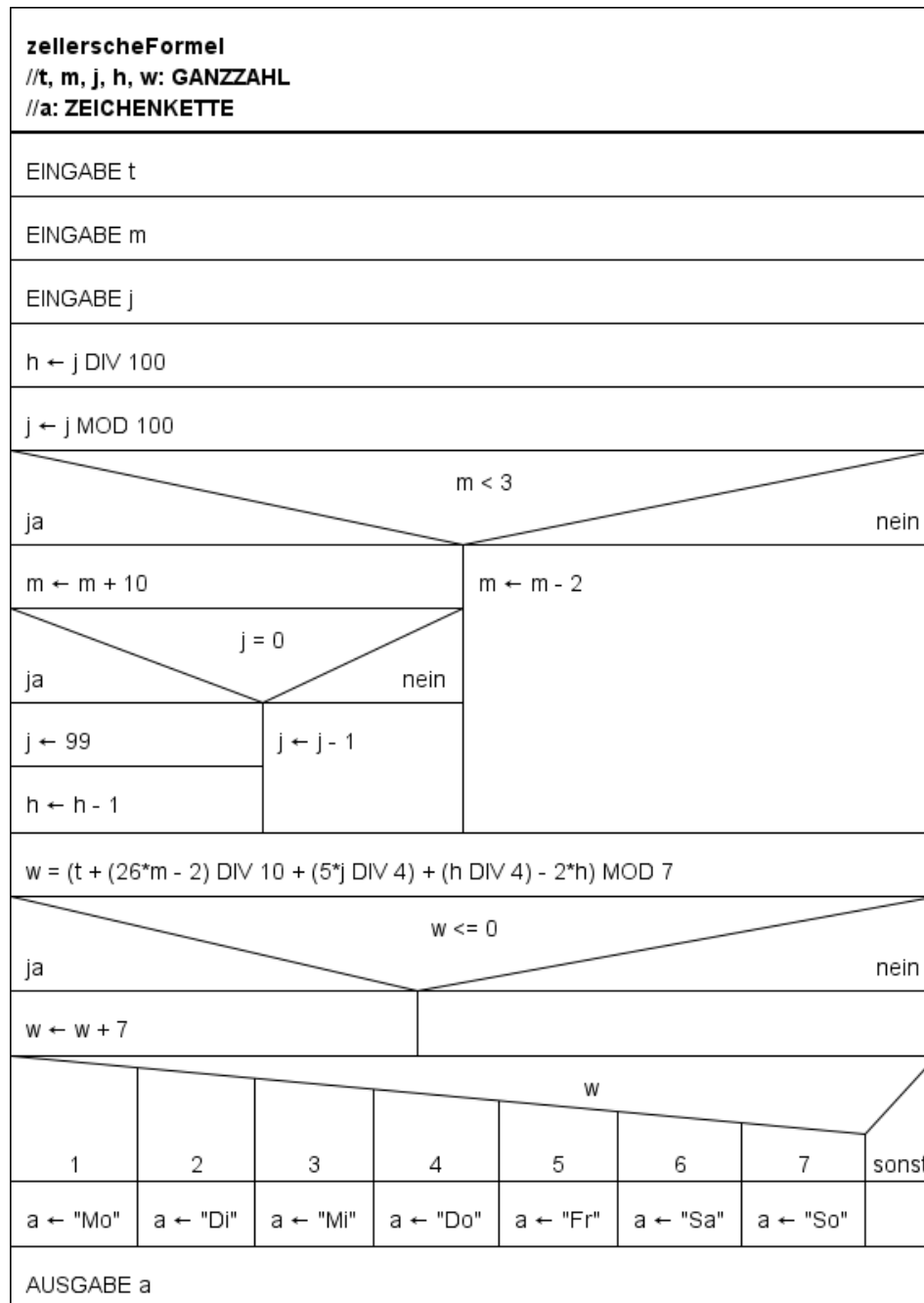
Name:

Vorname:

Klasse:

ZELLERSche Formel

Die Gesetzmäßigkeiten des immerwährenden Kalenders wurden von dem Geistlichen CHRISTOPH ZELLER 1885 zu einer mathematischen Formel zusammengefasst. Der darauf fußende Algorithmus erlaubt es, für jedes Datum den Wochentag zu bestimmen.



- 1) Beschreiben Sie das Prinzip einer mehrseitigen **➤ Auswahl** am Beispiel.
- 2) Implementieren Sie das Struktogramm und testen Sie die korrekte Arbeitsweise des Algorithmus für verschiedene Eingaben.
- 3) Beim Implementieren können sich Fehler einschleichen. Informieren Sie sich über die Fehlerarten **➤ Syntaxfehler**, **➤ logischer Fehler** und **➤ Laufzeitfehler**, deren Auffinden und Beheben.



Imperatives Problemlösen in Java

Name:

Vorname:

Klasse:

Zahl π

Die Zahl π ist eine der faszinierendsten Zahlen der Mathematik. Da ihre Ziffernfolge keine Logik erkennen lässt, wird sie in der Informatik zur Bestimmung von Pseudozufallszahlen eingesetzt. Eine Möglichkeit der Berechnung von π auf n Stellen hinter dem Komma bietet der sog. Zapfhahn- oder Tröpfel-Algorithmus von RABINOWITZ/WAGON (1995), den das Struktogramm zeigt.

- 1) Beschreiben Sie das Prinzip und die Java-Umsetzung einer **Zahlschleife**.
- 2) Ordnen Sie den vorhandenen Zählschleifen die Begriffe auf- bzw. absteigende Schleife zu.
- 3) Informieren Sie sich über das Prinzip eines **Feldes** als Datenstruktur. Beschreiben Sie mit Hilfe des Arbeitsblattes „Datenfelder (Array) in Java“ die Umsetzung in Java.
- 4) Implementieren Sie das Struktogramm und testen Sie die korrekte Arbeitsweise des Algorithmus für verschiedene Eingaben.

zapfhahn

//p,q,r,c,z,n,m: GANZZAHL

//a: FELD von GANZZAHL

//e: ZEICHENKETTE

EINGABE m //Anzahl der Stellen

a[] \leftarrow new int[12001] //neues Feld mit 12001+1 Elementen

p \leftarrow -1; c \leftarrow -1; z \leftarrow -1; n \leftarrow 0

e \leftarrow ""

FÜR i \leftarrow 0 BIS 12000

a[i] \leftarrow 2

FÜR j \leftarrow -1 BIS m

FÜR i \leftarrow 0 BIS 12000

a[i] \leftarrow 10 * a[i]

FÜR i \leftarrow 12000 HINAB BIS 1

q \leftarrow a[i] DIV (2*i + 1)

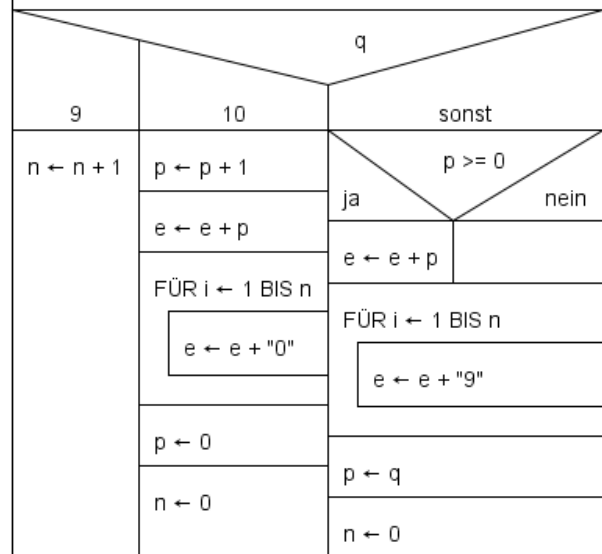
r \leftarrow a[i] MOD (2*i + 1)

a[i] \leftarrow r

a[i-1] \leftarrow a[i-1] + i*q

q \leftarrow a[0] DIV 10

a[0] \leftarrow a[0] MOD 10



AUSGABE e