



# ADS Aufgabenblatt 42

## Allgemeine Hinweise

Dieses Blatt enthält zwei Aufgaben mit Musterlösung. Es wurde erstellt für das ADSL-Tutorium im WS 2016 von Dominik Meixner (1324227).

Ausgabe der Übung:	Mo., 09.01.2017
Abgabe:	Direkt vor dem Testat
Testat am:	dann und wann

- Die Abnahme der Übungen gilt als **Prüfungsleistung**. Bei einer Verhinderung durch Krankheit ist eine ärztliche Bescheinigung der Arbeitsunfähigkeit vorzulegen.
- Geben Sie Ihre Lösung vorab schriftlich beim Dozenten ab. Benutzen Sie das im Moodle-Kurs vorgegebene Deckblatt für die Übungen. Drucken bzw. Schreiben Sie auf jede Seite die Seitenzahl und heften Sie Ihre Ausarbeitung. **Ausarbeitungen in einem anderen Format werden nicht berücksichtigt**. Vergessen Sie nicht zu unterschreiben.
- Während der Abnahme sind die Ergebnisse am Rechner live zu demonstrieren.
- Bei der Abnahme der Übung ist der Studentenausweis vorzulegen.

## Lernziele

- 

## Hinweise zur Bearbeitung

Vermeiden Sie unterschiedliche Lösungen, wenn ein Algorithmus in verschiedener Darstellungsform (Struktogramm, Pseudo-Code, usw.) verlangt ist.

In diesem Arbeitsblatt sollen Sie Klassen entwerfen, die auf Klassen beruhen, die nicht vollständig ausprogrammiert sind. Sie brauchen nur solche Methoden zu implementieren, die zur Lösung der Aufgabe wirklich benötigt werden. Es kann sinnvoll sein, Sourcecode durch kopieren in einer neuen Klasse wiederzuverwenden, Vererbung ist aber auch möglich und erlaubt. Den Source-Code finden Sie über die Links im Moodle-Kurs.



# 1 Berechnung des Osterdatums

Ostern ist im Christentum die jährliche Gedächtnisfeier der Auferstehung Jesu Christi, der nach dem Neuen Testament (NT) als Sohn Gottes den Tod überwunden hat. Da Jesu Tod und Auferstehung laut NT in eine Pessach-Woche fielen, bestimmt der Termin dieses beweglichen jüdischen Hauptfestes auch das Osterdatum. Es wird über einen Lunisolarkalender bestimmt und fällt in der Westkirche immer auf den Sonntag nach dem ersten Frühlingsvollmond, im gregorianischen Kalender also frühestens auf den 22. März und spätestens auf den 25. April.<sup>1</sup>

Für Berechnung des Datums des Ostersonntags in den Jahren 1900 bis 2099 hat T. O'Beirne folgenden Algorithmus entwickelt:

---

**Algorithm 1** Pseudo-Code Berechnung des Osterdatums

---

```
Gib aus: "Bitte Jahreszahl eingeben.";
Lies natürliche Zahl  $y$  ein;
 $a := (y - 1900) \bmod 19$ ;
 $b :=$  ganzzahliger Teil von  $(7a + 1)/19$ ;
 $m := (11a + 4 - b) \bmod 29$ ;
 $q :=$  ganzzahliger Teil von  $(y - 1900)/4$ ;
 $w := (y - 1900 + q + 31 - m) \bmod 7$ ;
 $d := 25 - m - w$ ;
if  $d > 0$  then
    Gib aus: "Der Ostersonntag im Jahr  $y$  ist am  $d$ . April";
else
     $d2 = 31 + d$ ;
    Gib aus: "Der Ostersonntag im Jahr  $y$  ist am  $d2$ . März";
fi
```

---

---

<sup>1</sup><https://de.wikipedia.org/wiki/Ostern>



## 1.1 Aufgabe: Ablaufprotokoll

Erstellen Sie ein Ablaufprotokoll für die Jahre 1817, 1917, 2016, 2017. Was fällt Ihnen auf?

Lösung/Hinweise:

### Ablaufprotokolle

Ausgabe: "Bitte Jahreszahl eingeben."

y:	1817	1917	2016	2017
a:	-7	17	2	3
b:	-2	6	0	1
m:	-13	11	26	7
q:	-20	4	29	29
w:	-3	6	3	2
d:	41	8	-4	16
d2:	–	–	27	–

Ausgabe: „Der Ostersonntag im Jahr 1817 ist am 41. April.“

Ausgabe: „Der Ostersonntag im Jahr 1917 ist am 8. April.“

Ausgabe: „Der Ostersonntag im Jahr 2016 ist am 27. März.“

Ausgabe: „Der Ostersonntag im Jahr 2017 ist am 16. April.“

### Was fällt auf?

Das Jahr 1817 liegt außerhalb des Definitionsbereiches, daher ist die Lösung 41. April falsch.

## 1.2 Aufgabe: Java-Programm

Schreiben Sie das Java-Programm hierfür. Fehlerhafte Eingaben sollen abgefangen werden.

Lösung/Hinweise:

```
1 import java.util.Scanner;
2 public class easterdate {
3     public static void main(String[] Args) {
4         System.out.println("Bitte geben Sie die Jahreszahl "+
5             "zur Berechnung des Osterdatums ein:");
6         Scanner sc = new Scanner(System.in);
7         int y = sc.nextInt();
8         sc.close();
9         if(y < 2100 && y >= 1900) {
```



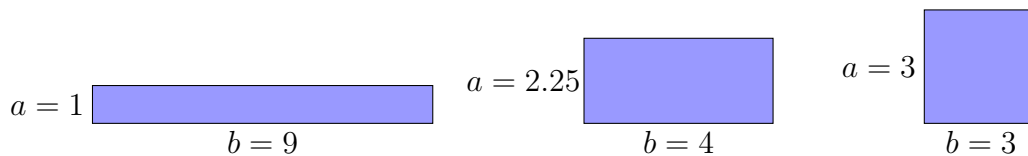
```
10         int a = (y-1900)%19;
11         int b = (7*a+1)/19;
12         int m = (11*a+4-b)%29;
13         int q = (y-1900)/4;
14         int w = (y-1900+q+31-m)%7;
15         int d = 25-m-w;
16         if(d > 0) {
17             System.out.println("Der Ostersonntag im Jahr "
18                               + y + " ist am " + d + ". April.");
19         } else {
20             int d2 = 31+d;
21             System.out.println("Der Ostersonntag im Jahr "
22                               + y + " ist am " + d2 + ". März.");
23         }
24     } else {
25         System.out.println("Fehlerhafte Eingabe.");
26     }
27 }
28 }
```



## 2 Rekursives Wurzelziehen

Ein Quadrat mit dem Flächeninhalt  $A$  hat die Seitenlänge  $a = \sqrt{A}$ . Aus dieser Tatsache leitet sich folgendes Annäherungsverfahren zum Wurzelziehen her. Man beginnt mit einem beliebigen Rechteck des Flächeninhalts  $A$  und nähert Schritt für Schritt die Seitenlängen  $a$  und  $b$  aneinander an. Das Verfahren bricht ab, wenn der Abstand von  $a$  und  $b$  ausreichend klein, das heißt  $a^2 \approx A$  ist.

Beispielhaft ist diese Annäherung in folgender Abbildung dargestellt. Zunächst hat das Rechteck mit dem Flächeninhalt  $A = 9$  die Seitenlängen  $a = 1$  und  $b = 9$ . Im nächsten Schritt folgt die bessere Schätzung  $a = 2,25$  und  $b = 4$ . Zum Schluss ist aus dem Rechteck ein Quadrat mit der Seitenlänge  $a = b = 3$  geworden. Das Verfahren hat also die Lösung  $\sqrt{9} = 3$ .



Die Rekursionsvorschrift des Verfahrens lautet:

$$x_{n+1} = \frac{1}{2} * \left( x_n + \frac{A}{x_n} \right)$$

### 2.1 Aufgabe: Pseudocode

Erstellen Sie den Pseudocode zum oben erläuterten Verfahren. Die Genauigkeit soll bei fünf Nachkommastellen liegen.

Lösung/Hinweise:

---

**Algorithm 2** Pseudo-Code Wurzelziehen

---

```
function main()
    Ausgabe: "Bitte geben Sie eine positive Dezimalzahl ein.";
    Lies positive Dezimalzahl  $A$  ein.;
     $a = \text{sqrtRek}(1, A)$ ;
    Ausgabe: "Die Wurzel aus  $A$  ist  $a$ .";
tcnuf
function sqrtRek( $x_n, A$ )
     $x_{n+1} = (x_n + A/x_n)/2$ ;
    if  $|x_{n+1} * x_{n+1} - A| < 0.00001$  then
         $x_{n+1} = \text{sqrtRek}(x_{n+1}, A)$ ;
    fi
    return  $x_{n+1}$ ;
tcnuf
```

---



## 2.2 Aufgabe: Java-Programm

Überführen Sie den Pseudocode in ein lauffähiges Java-Programm. Hierbei soll bei jedem Schritt das Zwischenergebnis ausgegeben werden.

Fehlerhafte Eingaben sollen abgefangen werden.

Lösung/Hinweise:

```
1 import java.util.Scanner;
2
3 public class wurzelziehen {
4     public static void main(String[] args) {
5         System.out.println("Bitte geben Sie eine "+
6             "positive Dezimalzahl ein.");
7         Scanner sc = new Scanner(System.in);
8         double A = sc.nextDouble();
9         sc.close();
10        if(A > 0) {
11            double a = wurzelRek(1,A);
12            System.out.println("Die Wurzel aus " + A +
13                " ist " + a + ".");
14        } else {
15            System.out.println("Fehlerhafte Eingabe.");
16        }
17    }
18
19    public static double wurzelRek(double xN, double A) {
20        double xNplus1 = (xN + A/xN)/2;
21        System.out.println(xNplus1);
22        if(Math.abs(xNplus1 * xNplus1 - A)> 0.00001) {
23            xNplus1 = wurzelRek(xNplus1, A);
24        }
25        return xNplus1;
26    }
27 }
```