

Bereich: Kontrollstrukturen

Schaltjahr

Musterlösung

Package: de.dhbwka.java.exercise.control

Klasse: LeapYear

```
package de.dhbwka.java.exercise.control;

import java.util.Scanner;

/**
 * @author DHBW lecturer
 * @version 1.01
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Suess
 */
public class LeapYear {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Welches Jahr auf Schaltjahr pruefen? ");
        int year = scan.nextInt();
        scan.close();
        boolean isLeapYear = false;
        if (year % 400 == 0) {
            isLeapYear = true;
        } else if (year % 100 == 0) {
        } else if (year % 4 == 0) {
            isLeapYear = true;
        }
        System.out.println(year + " ist " +
            (isLeapYear ? "ein" : "kein") +
            " Schaltjahr");
    }
}
```

Beispielausgaben:

Welches Jahr auf Schaltjahr prüfen? 2016
2016 ist ein Schaltjahr

Welches Jahr auf Schaltjahr prüfen? 2017
2017 ist kein Schaltjahr

Welches Jahr auf Schaltjahr prüfen? 2000
2000 ist ein Schaltjahr

Welches Jahr auf Schaltjahr prüfen? 1900
1900 ist kein Schaltjahr

Bereich: Kontrollstrukturen**Temperaturtabelle****Musterlösung****Package:** de.dhbwka.java.exercise.control**Klasse:** TemperatureTable

```
package de.dhbwka.java.exercise.control;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class TemperatureTable {

    public static void main(String[] args) {
        System.out.println("Fahrenheit | Celsius");
        System.out.println("-----+-----");
        for (int fahrenheit=0; fahrenheit<=300; fahrenheit+=20) {
            double celsius = (5./9) * (fahrenheit - 32);
            System.out.printf("%3d          | %5.1f",
                              fahrenheit,
                              celsius);
            System.out.println();
        }
    }
}
```

Beispielausgabe:

```
Fahrenheit | Celsius
-----+-----
0          | -17.8
...
300        | 148.9
```

Bereich: Kontrollstrukturen**Wildbestand****Musterlösung****Package:** de.dhbwka.java.exercise.control**Klasse:** Deers

```
package de.dhbwka.java.exercise.control;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class Deers {

    public static void main(String[] args) {
        int deers = 200;
        int iteration = 0;
        do {
            deers = (int) ((deers * 1.1) - 15);
            System.out.println(++iteration + ": " + deers + " Hirsche");
        } while (deers < 300);
    }
}
```

Beispielausgabe:

```
1: 205 Hirsche
2: 210 Hirsche
3: 216 Hirsche
4: 222 Hirsche
5: 229 Hirsche
6: 236 Hirsche
7: 244 Hirsche
8: 253 Hirsche
9: 263 Hirsche
10: 274 Hirsche
11: 286 Hirsche
12: 299 Hirsche
13: 313 Hirsche
```

Bereich: Kontrollstrukturen

Einmaleins

Musterlösung

Package: de.dhbwka.java.exercise.control

Klasse: MultiplicationTable

```
package de.dhbwka.java.exercise.control;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class MultiplicationTable {

    public static void main(String[] args) {
        // multiplication table 10x10
        for (int i = 1; i <= 10; i++) {
            for (int j = 1; j <= 10; j++) {
                System.out.printf("%4d", i*j);
            }
            System.out.println(); // next line
        }
    }
}
```

Beispielausgabe:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Bereich: Kontrollstrukturen

Aufsummieren

Musterlösung

Package: de.dhbwka.java.exercise.control

Klasse: AddUp

```
package de.dhbwka.java.exercise.control;

import java.util.Scanner;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class AddUp {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int input = 0;
        int sum = 0;
        while (input >= 0) {
            sum += input;
            System.out.print("Zahl eingeben (<0 für Abbruch): ");
            input = scan.nextInt();
        }
        System.out.println("Summe: "+sum);
        scan.close();
    }

    /**
     * alternative implementation with do-while
     */
    public static void mainAlternative(String[] args) {
        Scanner scan = new Scanner(System.in);
        int input = 0;
        int sum = 0;
        do {
            sum += input;
            System.out.print("Zahl eingeben (<0 für Abbruch): ");
            input = scan.nextInt();
        } while (input >= 0);
        System.out.println("Summe: "+sum);
        scan.close();
    }
}
```

Beispielausgabe:

```
Zahl eingeben (<0 für Abbruch): 4  
Zahl eingeben (<0 für Abbruch): 3  
Zahl eingeben (<0 für Abbruch): 7  
Zahl eingeben (<0 für Abbruch): 2  
Zahl eingeben (<0 für Abbruch): -1  
Summe: 16
```

Bereich: Kontrollstrukturen**Schuhgrößen****Musterlösung****Package:** de.dhbwka.java.exercise.control**Klasse:** ShoeSize

```
package de.dhbwka.java.exercise.control;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class ShoeSize {

    public static void main(String[] args) {
        System.out.println("Zentimeter | Größe");
        System.out.println("-----+-----");
        for (int size=30; size<50; size++) {
            double length = size / 1.5;
            System.out.printf("%5.2f - %5.2f | %2d",
                              (length - 2./3),
                              length,
                              size);
            System.out.println();
        }
    }
}
```

Bereich: Kontrollstrukturen**Babylonisches Wurzelziehen (Heronverfahren)****Musterlösung****Package:** de.dhbwka.java.exercise.control**Klasse:** Babylon

```
package de.dhbwka.java.exercise.control;

import java.util.Scanner;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class Babylon {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Wurzel aus welcher Zahl ziehen? ");
        double a = scan.nextInt();
        scan.close();

        double xn = 1.0; // randomly selected as x0
        double oldxn;
        do {
            oldxn = xn;
            xn = (xn + (a / xn)) / 2;
        } while (Math.abs(oldxn-xn)>=10E-6);
        System.out.println("Die Wurzel aus " + a + " ist " + xn);
    }
}
```

Beispielausgabe:

```
Wurzel aus welcher Zahl ziehen? 25
xn: 1.0
xn: 13.0
xn: 7.461538461538462
xn: 5.406026962727994
xn: 5.015247601944898
xn: 5.000023178253949
xn: 5.000000000053722
Die Wurzel aus 25.0 ist 5.0
```


Bereich: Kontrollstrukturen**Zahlenraten****Musterlösung****Package:** de.dhbwka.java.exercise.control**Klasse:** NumberGuess

```
package de.dhbwka.java.exercise.control;

import java.util.Scanner;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 */
public class NumberGuess {

    public static void main(String[] args) {
        int limit = 100;
        int end;
        Scanner scan = new Scanner(System.in);
        String bestPlayer = "";
        int leastTrials = -1;        // negativer Wert: bisher wurde noch kein
                                   // Spiel durchgeführt

        do {
            int numberToGuess = (int) (Math.random() * limit + 1);
            // System.out.println("#### Zu erratende Zahl: "
            //                     + zuErratendeZahl + " ####");
            int trials = 0;
            boolean guessed = false;
            System.out.print("Geben Sie bitte Ihren Namen ein: ");
            String name = scan.next();

            do {
                System.out.print("Geben Sie bitte eine Zahl aus dem Bereich "
                                   + "1 und 100 ein: ");
                int tip = scan.nextInt();

                trials++;
                if (tip > numberToGuess) {
                    System.out.println("Versuch Nr. " + trials + ": " + tip
                                         + " zu hoch!");
                } else if (tip < numberToGuess) {
                    System.out.println("Versuch Nr. " + trials + ": " + tip
                                         + " zu niedrig!");
                } else {
                    System.out.println("Versuch Nr. " + trials
                                         + ": Sie haben die Zahl erraten!!!");
                    guessed = true;
                    if ((leastTrials < 0) | (trials < leastTrials)) {
                        leastTrials = trials;
                        bestPlayer = name;
                    }
                }
            } while (!guessed);
        } while (true);
    }
}
```

```
        }  
    }  
    } while (!guessed);  
  
    do {  
        System.out.print("Wollen Sie das Spiel beenden (0), fortsetzen "  
            + "(1) oder den bisher besten Spieler(in) ausgeben "  
            + "lassen (2)? ");  
        end = scan.nextInt();  
        if ((end<0) || (end>2))  
            System.out.println("Eingabefehler!");  
        if (end == 2) {  
            System.out.println("Bisher beste/r Spieler/in: "  
                + bestPlayer  
                + ", geringste Anzahl Versuche: "  
                + leastTrials);  
        }  
    } while ((end<0) || (end>1));  
} while (end == 1);  
}
```